Modular-Hardware Toolbox for Data Acquisition Systems in High Energy Physics Experiments

Zur Erlangung des akademischen Grades eines

DOKTORS DER INGENIEURWISSENSCHAFTEN (Dr.-Ing.)

von der KIT-Fakultät für Elektrotechnik und Informationstechnik des
Karlsruher Instituts für Technologie (KIT)
angenommene

DISSERTATION

von

M.Sc. Torben Mehner
geb. in Speyer

Tag der mündlichen Prüfung:	10.03.2026
Hauptreferent:	Prof. Dr. rer. nat. Marc Weber
Korreferent:	Prof. Dr.-Ing. Dr. h. c. Jürgen Becker
Korreferent:	PD Dr.-Ing. Oliver Sander

Kurzfassung

Das Upgrade des High-Luminosity Large Hadron Collider (HL-LHC) für das Compact Muon Solenoid (CMS)-Experiment erfordert angesichts stark steigender Datenraten eine Neugestaltung der Back-End-Elektronik. Herkömmliche Entwurfsprozesse für Datenerfassungssysteme (DAQ-Systeme) stoßen dabei an Grenzen hinsichtlich Entwicklungsdauer, Komplexität und Wiederverwendbarkeit. Während bestehende Modularitätskonzepte vorwiegend auf Systemebene ansetzen, fehlt bislang ein methodischer Rahmen zur Anwendung modularer Prinzipien innerhalb einzelner Leiterplatten. Diese Thesis schließt diese Lücke durch einen werkzeugunabhängigen Entwurfsansatz zur Modularisierung komplexer Hardwarearchitekturen.

Hierzu wurde eine modulare Hardware-Toolbox entwickelt, die Prozesse, Strukturen und Schnittstellendefinitionen zur Entwicklung wiederverwendbarer Hardwarekomponenten beschreibt. Sie dient als methodisches Rahmenwerk für zentrale Entwurfsprozesse und unterstützt eine systematische Modularisierung des Designs, wodurch Entwicklungsaufwand und Fehleranfälligkeit reduziert sowie Wiederverwendbarkeit und Anpassungsfähigkeit erhöht werden. Die Anwendung der Toolbox wird exemplarisch mit dem Werkzeug *Altium Designer* demonstriert.

Die Validierung des Ansatzes erfolgte anhand der *Serenity-S1*-Karte, einer Back-End-Verarbeitungseinheit für mehrere CMS-Subdetektoren, die auf dem Advanced Telecommunications Computing Architecture (ATCA)-Standard basiert. Sie integriert einen modularen Servicebereich für Infrastrukturfunktionen und einen flexiblen Nutzbereich mit einem AMD Virtex UltraScale+ VU13P Field-Programmable Gate Array (FPGA) und optischen Links. Die Karte erreicht einen aggregierten Datendurchsatz von über $3\,\mathrm{Tbit\,s^{-1}}$ bei Latenzen im Mikrosekundenbereich, die maßgeblich durch die Datenverarbeitung im FPGA bestimmt sind. Ergänzend zur technischen Verifikation wurde der Entwicklungsprozess anhand einer Capability Maturity Model Integration (CMMI) bewertet, wobei der Vergleich mit Projekten ohne Toolbox eine deutliche Steigerung der Prozessreife und Standardisierung zeigte.

Die Arbeit zeigt, dass der modulare Ansatz die verteilte Entwicklung in internationalen Kollaborationen erleichtert und eine schnelle Anpassung an technologische Änderungen ermöglicht. Die Wiederverwendung mehrerer entwickelter Module in unabhängigen Projekten belegt die Übertragbarkeit des Ansatzes und unterstreicht den methodischen Beitrag der Modular-Hardware-Toolbox zur Systematisierung des Leiterplattenentwurfs.

Abstract

The upgrade of the High-Luminosity Large Hadron Collider (HL-LHC) for the Compact Muon Solenoid (CMS) experiment imposes new requirements on the architecture and development of back-end electronics due to the rapidly increasing data rates. Conventional design processes for Data Acquisition (DAQ) systems are reaching their limits in terms of development time, complexity, and reusability. While existing modularity concepts primarily focus on the system level – combining multiple printed circuit boards (PCBs) into complete systems – a methodological framework for applying modular design principles within individual boards has been lacking. This thesis addresses this gap by introducing a tool-agnostic design approach for structuring complex hardware architectures.

For this purpose, a modular hardware toolbox has been developed that defines processes, structures, and interface specifications for creating reusable hardware components. It serves as a methodological framework for the standardisation of core design processes and supports systematic modularisation of the design, thereby reducing development effort and error susceptibility while increasing reusability and adaptability. The approach is exemplified using an implementation with *Altium Designer*.

The concept was validated through the development of the *Serenity-S1* Advanced Telecommunications Computing Architecture (ATCA) board, a back-end processing card for multiple CMS subdetectors. The design integrates a modular service region for infrastructure functions and a flexible payload area hosting an AMD Virtex UltraScale+ VU13P FPGA and optical links. The board achieves an aggregate data throughput exceeding $3\,\mathrm{Tbit\,s^{-1}}$ with latencies in the microsecond range, which are largely determined by the data processing implemented in the FPGA. In addition to technical verification, the development process was evaluated using a Capability Maturity Model Integration (CMMI), where comparison with projects developed without the toolbox demonstrated a significant increase in process maturity and standardisation.

The results show that the modular approach facilitates distributed development in international collaborations and enables rapid adaptation to technological changes. The reuse of several developed modules in independent projects demonstrates the transferability and sustainability of the approach and underlines the methodological contribution of the modular hardware toolbox to the systematisation of printed circuit board design.

Acknowledgments

I would like to express my sincere gratitude to all those who contributed to the completion of this dissertation. My deepest appreciation goes to my supervisors, Prof. Marc Weber, Prof. Jürgern Becker and PD Dr. Oliver Sander, for their invaluable guidance, continuous support, and for giving me the opportunity to work on this challenging yet highly rewarding project.

I am especially grateful to Matthias Balzer for his continuous assistance – not only during the writing of this thesis, but also throughout the development of the Serenity-S1. My sincere thanks also go to all colleagues – Luis, Marvin, Hendrik, Michael, Denis, Alexander, Jennifer, Gregory, Duncan, Alex, Thomas, Kate, and Andrew – for their dedication and for spending countless hours on the Serenity-S1 card, often stepping in for me while I was working on this dissertation.

I would also like to thank my fellow PhD candidates – Marvin, Hendrik, Robert, Timo, Lukas, and Fabian – for their helpful discussions, encouragement, and friendship during this time.

Finally, I wish to thank my wife Jessica for her unwavering support and patience, especially during the more difficult phases of this journey, as well as both our families for their constant encouragement and understanding.

I acknowledge the support by the Doctoral School “Karlsruhe School of Elementary and Astroparticle Physics: Science and Technology”

Declaration

I declare that I have developed and wrote the enclosed thesis entirely by myself, and that I did not use any sources or means that were not explicitly stated in the text. The development of the Serenity-S1 described in Chapter 5 are collaborative efforts of the community which led to the publications [1, 2, 3].

Contents

1 Introduction

The High-Luminosity Large Hadron Collider (HL-LHC) [15, 16] constitutes the next major upgrade of the LHC, designed to increase its instantaneous luminosity by approximately an order of magnitude compared with its initial configuration. This enhancement will enable the collection of integrated datasets of up to $3\,\mathrm{ab}^{-1}$ over 10 years[1] of operation for the CMS and A Toroidal LHC ApparatuS (ATLAS) experiments each, supporting precision measurements of Higgs boson couplings, improved sensitivity to electroweak processes such as vector boson scattering, and extended searches for physics beyond the Standard Model, including rare and unconventional signatures such as long-lived particles [18, 19]. Operating under these conditions implies an average of up to 140–200 proton–proton interactions per bunch crossing (so-called pileup), significantly increasing detector occupancy, event complexity, and the volume of data associated with each recorded collision.

To cope with this environment, the CMS experiment is undergoing an extensive Phase-2 upgrade [18], introducing a new silicon tracker [20], high-granularity calorimeter endcaps [21], and upgraded muon [22] and timing systems [23], all equipped with front-end electronics optimised for high-rate and high-radiation operation. In the HL-LHC regime, the majority of the data read out per event originates from overlapping low-energy pileup interactions rather than from the high-energy collisions of interest. While detector upgrades are essential to mitigate the effects of pileup through improved granularity and timing, they also lead to a substantial increase in data rates, channel counts, and trigger complexity. Addressing these challenges requires back-end architectures that can evolve over decades of operation while supporting the technical and organisational complexity inherent in large, distributed detector collaborations.

DAQ architectures have become increasingly sophisticated, reflecting broader trends in experimental science towards configurable, reusable, and collaboratively developed instrumentation. Early systems were largely bespoke and monolithic [24, 25], optimised for single detectors and specific operating conditions, but offering limited flexibility and reuse beyond their original context. In contrast, contemporary DAQ designs [3, 26, 27, 28, 29, 30, 31] increasingly adopt modular architectures to support scalability, technology evolution, and parallel development across institutes. Modularity at system level alone, however, does not resolve all challenges; it introduces new

[1] The $3\,\mathrm{ab}^{-1}$ collected in the CMS experiment are equivalent to more than 6 EB of raw data storage [17]

demands on interface definition, signal integrity, verification strategy, and configuration management, particularly in large international collaborations. Addressing these demands requires not only architectural decomposition but also a structured methodological framework that formalises modular design principles across the full hardware development lifecycle.

To address these constraints, this thesis explores modular hardware design at finer levels of granularity, enabling development tasks to be distributed more effectively and design scalability to be improved. While conventional modularity in DAQ systems typically operates at the board level, this research extends the concept to intra-board modularity by defining modular boundaries within a single Printed Circuit Board (PCB). In this approach, sub-circuits are treated as reusable and verifiable functional units with well-defined interfaces, allowing them to be designed, validated, and evolved independently while preserving electrical and mechanical integration at system level. Inspired by principles long established in software engineering [32, 33, 34, 35], such intra-board modularity provides a technical foundation for greater reuse and consistency. Realising its full potential, requires a structured framework that integrates design practices, verification artefacts, and process governance. This thesis addresses that need by developing a formalised methodology for modular, collaborative hardware development.

Research Aim and Objectives

The aim of this research is to develop and validate a structured framework – the Modular-Hardware Toolbox – for the modular design of DAQ systems used in High-Energy Physics (HEP) experiments. The Toolbox is intended to facilitate reconfigurable, maintainable, and reproducible hardware development through formalised modular design principles.

In parallel, this research delivers a production-scale DAQ processing card for the CMS Phase-2 upgrade. The Serenity-S1 ATCA processing card serves both as a primary validation environment for the proposed methodology and as an operational back-end DAQ system within the CMS experiment.

The work presented in this thesis pursues the following objectives:

1. Analyse recurring technical and organisational challenges in the development of high-performance HEP DAQ electronics and derive requirements for scalable, reproducible hardware design methodologies.

2. Develop the *Modular-Hardware Toolbox* as a structured framework for modular intra-board hardware design, enabling reuse, verification, and collaborative development across the full electronic design lifecycle.

3. Validate the proposed methodology through the design, deployment, and process-oriented evaluation of the *Serenity-S1* back-end DAQ processing card developed for the CMS Phase-2 upgrade.

Scope and Methodology

To achieve these objectives, this thesis adopts a design-oriented engineering methodology that combines requirements analysis, framework development, prototyping, and structured empirical evaluation. Functional and non-functional requirements are first derived through a systematic analysis of technical and organisational challenges in existing DAQ developments. On this basis, the Modular-Hardware Toolbox is designed, implemented, and applied as a structured methodology supporting modular design, verification, and collaborative development.

The Serenity-S1 ATCA processing card, developed for the CMS Phase-2 upgrade, serves both as the primary validation environment for the proposed methodology and as a production-scale back-end DAQ system. Its development enables the evaluation of the Toolbox under realistic constraints, including high data rates, long design cycles, and distributed collaboration. The evaluation therefore combines a technical performance assessment of the Serenity-S1 system with a process-oriented analysis addressing reproducibility, configuration management, and collaborative efficiency. As fully controlled process comparisons are not feasible in complex PCB-based developments, the methodology integrates a structured, semi-quantitative assessment of process capability to evaluate development maturity and methodological impact.

The scope of this work is limited to HEP data acquisition systems; while the underlying concepts are transferable, applications in other scientific or industrial domains are considered beyond the immediate focus of this thesis. Nevertheless, the framework and insights developed here are expected to be applicable to other domains that face comparable challenges in terms of complexity, scalability, and distributed hardware development.

2 Fundamentals

This chapter introduces the technical and conceptual foundations required for the subsequent discussion of modular hardware design for data acquisition systems. Its purpose is to establish common terminology and to outline the architectural and technological constraints under which modern high-performance DAQ electronics are developed.

The chapter first summarises the role and structure of DAQ systems in high-energy physics, followed by an introduction to field-programmable gate arrays (FPGAs), which provide the deterministic, low-latency processing capabilities central to contemporary back-end architectures. Building on this background, fundamental concepts of modular hardware design are introduced, with a focus on reusability, scalability, and the practical limits imposed by performance-critical electronic systems. Finally, the CMS experiment and its DAQ requirements are briefly outlined to provide an operational context for the hardware platforms discussed in later chapters.

2.1 Data Acquisition Systems

A data acquisition (DAQ) system is designed to capture physical phenomena, convert them into digital representations, and make the resulting data available for processing, storage, and analysis in a reliable and reproducible manner [36]. DAQ systems are ubiquitous across experimental science and engineering, ranging from simple laboratory instruments and industrial monitoring equipment to large-scale scientific facilities such as particle detectors, astronomical observatories, and medical imaging systems. Their scale and complexity vary widely, depending on factors such as signal bandwidth, required timing precision, data throughput, environmental constraints, and the number of readout channels.

This diversity is illustrated by comparing a low-rate sensor, such as a digital thermometer sampling temperature every few seconds, with the CMS experiment at the Large Hadron Collider (LHC), where detector front-ends produce data synchronised to a $40\,\mathrm{MHz}$ bunch-crossing clock, corresponding to a sampling interval of $25\,\mathrm{ns}$. Despite these differences in scale, all DAQ systems share a common functional structure and are governed by the same fundamental trade-offs between bandwidth, latency, precision, and reliability.

2.1.1 System Architecture

From a hardware perspective, a generic DAQ system can be decomposed into a sequence of functional blocks, as illustrated in Figure 2.1 [37]:

- a **sensor** that transduces a physical quantity into an electrical signal,
- **signal-conditioning circuitry** that amplifies, filters, and shapes the analog signal,
- an **Analog-to-Digital Converter (ADC)** that digitises the conditioned signal,
- a **processing unit** that performs real-time operations on the digital data stream, and
- a **host computer or server** for aggregation, control, and long-term storage.

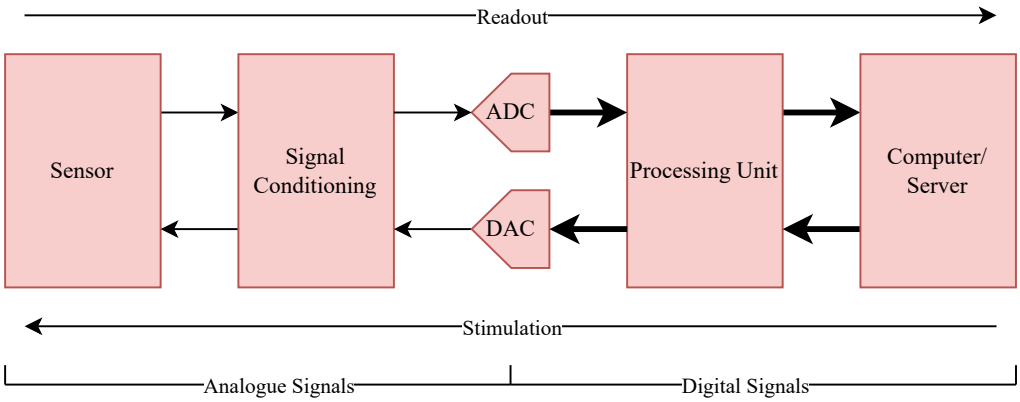


Figure 2.1: Conceptual architecture of a generic data acquisition (DAQ) system.

In some applications, the DAQ system also generates controlled stimulation or calibration signals. These are typically produced by a Digital-to-Analog Converter (DAC), shaped by analog circuitry, and applied to the measured system. In high-energy physics, such calibration paths are used to inject defined charge or voltage pulses into detector electronics to calibrate gain, timing, and linearity and to compensate for radiation-induced effects [38]. Similar concepts appear in other domains, for example in biomedical imaging, where excitation pulses are transmitted into tissue prior to signal reception by the sensors.

The components directly interfacing with the sensor – including analog conditioning, digitisation, and any local calibration circuitry – are commonly referred to as the *front-end* of a DAQ system. The analog front-end must be closely matched to the characteristics of the employed sensor technology, as detector types differ widely and impose distinct requirements on noise performance, dynamic range, and bandwidth. In high-energy physics, sensors are designed to measure the

position, momentum, or energy deposition of relativistic particles and are typically based on photon emission or ionisation mechanisms [39]. Photonic detectors such as scintillators or Cherenkov radiators are read out using photodetectors including Silicon Photomultipliers (SiPMs) or Avalanche Photodiodes (APDs), which convert incident light into current pulses. Ionisation-based detectors measure the charge generated by particle interactions in gases or semiconductors, where an applied electric field transports charge carriers to electrodes, producing a measurable signal proportional to the deposited energy.

The analog signal-conditioning chain – comprising amplification, shaping, and digitisation – must therefore be tuned to the intrinsic sensor response to achieve the desired resolution and timing precision [39, 40, 41]. Because analog signals are sensitive to electromagnetic interference, particularly over extended transmission paths, modern detector designs often integrate the analog front-end and digitisation stages close to the sensor. This minimises analog cabling and allows only digital data to be transmitted to downstream electronics. Practical constraints such as cooling capacity, radiation tolerance, and mechanical integration, however, often limit the extent to which such integration is feasible.

Once digitised, data are transmitted away from the detector toward downstream processing and storage systems. This transition marks the boundary between the front-end and the *back-end* of a DAQ system. The back-end comprises the processing, buffering, formatting, and high-speed data transport functions required to aggregate data from many front-end sources and to interface with trigger, control, and storage infrastructures. Depending on system requirements, various interconnect technologies are employed, including PCIe, Ethernet, and high-speed optical links. In large-scale physics experiments, optical fibre communication is typically preferred due to its high bandwidth, electromagnetic immunity, and ability to span the several-hundred-metre distances separating detectors from back-end systems.

This work focuses on the detector back-end, i.e. the processing layer responsible for data concentration, formatting, buffering, and high-speed transmission. While modularisation of front-end electronics is possible in certain contexts, it is often constrained by sensor-specific analog requirements and environmental conditions. By contrast, the back-end exhibits a higher degree of architectural commonality across otherwise heterogeneous DAQ systems, making it a natural target for modular hardware design. The implications of these constraints for modularisation are discussed in section 4.4.

2.1.2 Trigger and Synchronisation Systems

In high-rate DAQ systems, storing all acquired data is typically infeasible. Instead, triggering mechanisms are employed to select events of potential interest while discarding the majority

of background data [39]. In the CMS experiment, event selection is implemented in a two-stage trigger system. The first stage, the Level-1 Trigger (L1T), is realised entirely in hardware and performs coarse real-time selection with deterministic latency using custom FPGA-based processing platforms. The second stage, the High-Level Trigger (HLT), applies more refined software-based algorithms to a reduced event sample using Commercial Off-The-Shelf (COTS) computing infrastructure [42].

With collisions occurring at the frequency of $40.079\,\mathrm{MHz}$, the CMS L1T makes a trigger decision every $25\,\mathrm{ns}$, which reduces the readout rate to approximately $750\,\mathrm{kHz}$, while the HLT further decreases it to about $7.5\,\mathrm{kHz}$. During the trigger decision latency, event data must be buffered locally, requiring memory resources whose depth scales directly with both input bandwidth and trigger latency [43]. Trigger architecture and buffering strategy therefore have a direct impact on the design of back-end processing systems.

An alternative paradigm is triggerless DAQ, in which all data are transmitted to a central processing farm and event selection is performed entirely in software. This approach is adopted by experiments such as Compressed Baryonic Matter (CBM) [44] and Large Hadron Collider beauty (LHCb) [45], enabled by advances in high-performance computing and high-bandwidth data links. While triggerless architectures simplify front-end logic, they impose stringent requirements on back-end throughput and network infrastructure.

Accurate system-wide synchronisation is essential to maintain temporal coherence across distributed acquisition nodes. Dedicated timing networks ensure that data streams from different detector components can be correctly correlated. For the HL-LHC, this requirement translates into extremely tight phase-stability constraints, with allowable timing variations in the order of $14\,\mathrm{ps}$ across the system [46]. In such sub-nanosecond regimes, as encountered in trigger distribution or time-of-flight measurements, protocols such as White Rabbit [47] provide deterministic phase alignment across large-scale detector installations.

2.1.3 System Diversity and Design Space

DAQ systems span a wide range of application domains, including high-energy and astroparticle physics, medical imaging, industrial monitoring, geophysics, and cryogenic quantum experiments. Rather than enumerating individual use cases, it is more instructive to consider the fundamental design dimensions along which these systems differ, as these dimensions directly constrain hardware architecture and methodology.

Key dimensions include the aggregate data rate, the number of readout channels, the required timing precision, the permissible system latency, and the operational environment. High-energy physics experiments such as CMS operate at extreme data rates, with raw detector outputs on the

order of $2.7\,\mathrm{Pbit\,s^{-1}}$ [42], necessitating aggressive data reduction and highly parallel back-end processing. Astroparticle experiments such as the LIGO gravitational-wave observatory [48] or the JUNO neutrino detector [49], by contrast, prioritise ultra-low noise performance and precise timing to detect rare or weak signals. Distributed observatories like IceCube or the Pierre Auger Observatory rely on globally synchronised DAQ networks to merge data from spatially separated sensors [50, 51].

Cryogenic DAQ systems introduce additional constraints related to thermal load and accessibility. Experiments employing superconducting or quantum sensors often operate well below $1\,\mathrm{K}$ and require readout electronics that minimise heat dissipation and cabling into the cryostat [52, 53]. In such systems, multiplexing and remote processing are essential architectural features. Biomedical and environmental monitoring systems impose yet different constraints, including functional safety, regulatory compliance, and long-term autonomous operation [54, 55].

Despite this diversity, all DAQ systems must address the same fundamental challenges: capturing signals at the required bandwidth, preserving signal integrity through analog and digital processing, maintaining coherent timing across the acquisition chain, and ensuring that data can be filtered, transmitted, and stored efficiently. The relative importance of these challenges varies across applications, but the underlying system structure remains consistent.

Although sensor technologies, analog front-ends, and operating environments differ widely across application domains, the dominant architectural constraints in modern DAQ systems arise at the back-end. Requirements on bandwidth, latency, synchronisation, and scalability impose similar processing, buffering, and communication structures across otherwise heterogeneous systems. This structural convergence makes the back-end processing layer the natural focus for reusable hardware architectures and methodological abstraction. In high-rate DAQ systems, these functions are almost exclusively realised in programmable logic, which provides the deterministic timing, parallelism, and throughput needed to meet these system-level constraints.

2.2 Field-Programmable Gate Arrays

Field-Programmable Gate Arrays (FPGAs) form the dominant processing devices for modern high-performance data acquisition back-ends. Their combination of deterministic timing, massive parallelism, and tight coupling to high-speed I/O distinguishes them from general-purpose processors and makes them uniquely suited to the stringent latency and throughput requirements encountered in large-scale DAQ systems [56]. This section introduces FPGAs from a systems-engineering perspective, progressing from their role within the design space between software and custom hardware to the architectural properties that ultimately constrain modularity and reuse at board level.

2.2.1 Fundamentals of FPGA-Based Processing

An FPGA is a type of integrated circuit whose internal logic functions can be configured by the user after manufacturing [57]. Unlike Application-Specific Integrated Circuits (ASICs), which are custom-fabricated for a fixed task, FPGAs remain reconfigurable throughout their lifetime, enabling adaptation to evolving requirements. Compared to general-purpose processors, FPGAs can provide orders-of-magnitude higher spatial parallelism, while avoiding the non-deterministic latency effects introduced by cache hierarchies and operating systems [58].

The development effort associated with FPGAs occupies an intermediate position between software-based systems and fully custom hardware. While FPGA designs are reconfigurable in principle, modifying their functionality requires changes to hardware description code, re-synthesis, and re-validation of timing and functional correctness. As a result, iteration cycles are substantially longer than in software-only systems, yet far shorter and less costly than the design cycles associated with ASICs.

Compared to Central Processing Units (CPUs), FPGAs trade ease of programming for deterministic execution and the ability to realise deeply pipelined, high-throughput datapaths. In contrast to Graphics Processing Units (GPUs), modern FPGAs expose large numbers of high-speed serial transceivers directly at the device boundary; some contemporary devices provide more than 128 serial lanes, yielding aggregate I/O bandwidths exceeding $3\,\mathrm{Tbit\,s^{-1}}$ per device [59]. Recent interconnect technologies such as NVIDIA's NVLink significantly increase the bandwidth available to GPU-based systems into the $\mathrm{Tbit\,s^{-1}}$ regime [60], and GPU-centric back-end architectures have already been successfully deployed in experiments such as LHCb, even when limited to PCIe-based interfaces [61]. Nevertheless, the fundamental architectural distinction remains: GPUs achieve throughput through scheduled, batch-oriented execution on a fixed microarchitecture, whereas FPGAs implement application-specific datapaths with deterministic timing and bounded latency. As a result, FPGAs are naturally suited for latency-critical trigger processing, while GPUs are predominantly employed in triggerless DAQ systems or in the HLT.

These properties make FPGAs particularly well suited to DAQ back-ends, where large fan-in from detector front-ends, deterministic latency, and precise synchronisation must be maintained simultaneously. Their reconfigurability further supports long experimental lifetimes, during which trigger algorithms may evolve based on changing physics cases.

2.2.2 Architectural Building Blocks of Modern FPGAs

The suitability of FPGAs for high-performance DAQ back-ends is fundamentally determined by the organisation of their silicon architecture. The structure and placement of programmable logic,

dedicated hard blocks, and routing resources directly constrain timing closure, data movement, and architectural partitioning.

FPGAs consist of thousands to millions of Configurable Logic Blocks (CLBs), which form the fundamental units of programmable logic. Figure 2.2 shows a simplified view of a CLB in an AMD UltraScale device. Each CLB contains a set of Look-Up Tables (LUTs), flip-flops, and multiplexers, enabling the implementation of arbitrary combinational and sequential logic functions.

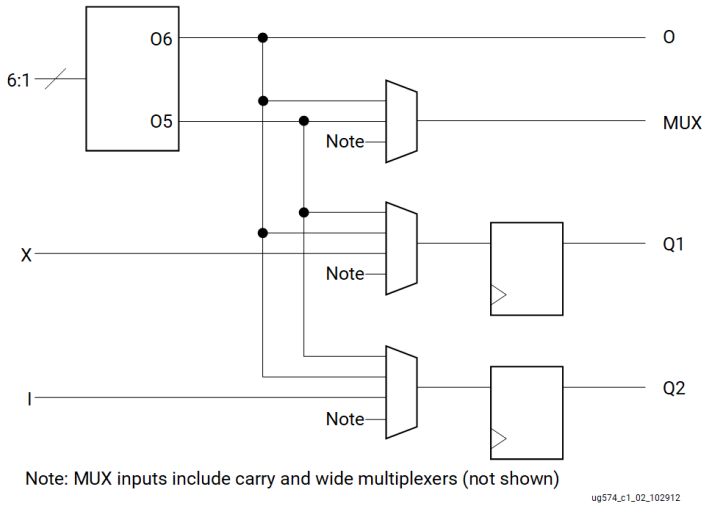


Figure 2.2: Simplified view of a Configurable Logic Block (CLB) from the AMD UltraScale architecture [62].

In an AMD UltraScale architecture, eight 6-input LUTs are grouped into a logic slice (SLICEL). Multiplexing structures allow these LUTs to be combined to realise logic functions with up to 7 to 9 input signals (see Figure 2.3). Each slice further incorporates an 8-bit carry chain that can be cascaded with neighbouring slices, enabling efficient implementation of arithmetic operations. In memory-capable slices (SLICEM), LUTs may alternatively be configured as distributed Random-Access Memory (RAM) or as shift registers [62].

Beyond general-purpose logic, modern FPGAs are structured around dedicated hard blocks that act as fixed architectural anchors within the fabric, as can be seen in Figure 2.4. These include embedded memories such as Block Random-Access Memory (BRAM) and High-Bandwidth Memory (HBM) for low-latency buffering, specialised arithmetic units in the form of Digital Signal Processings (DSPs) slices, Phase-Locked Loops (PLLs) for clock generation and conditioning, and Multi-Gigabit Transceiverss (MGTs) for high-speed serial communication. In many cases, these blocks are not merely efficiency optimisations but enable functionality that cannot be realised using CLB-based logic alone due to analog or mixed-signal requirements.

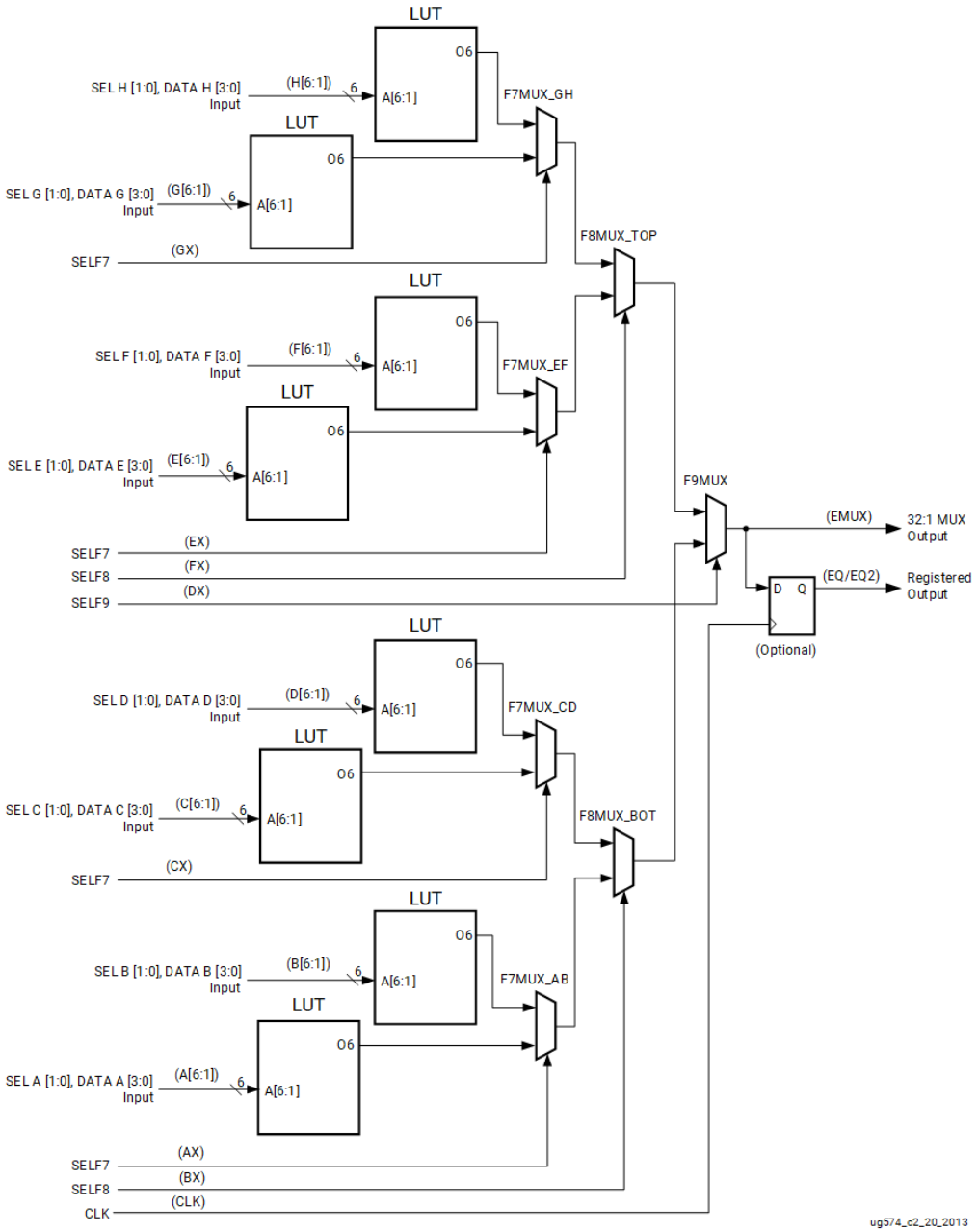


Figure 2.3: Full Configurable Logic Block (CLB) of the AMD UltraScale architecture, including multiplexing structures [62].

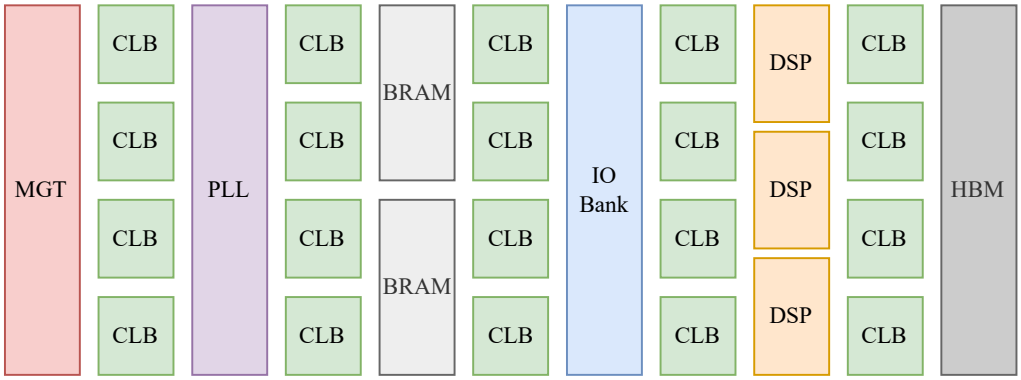


Figure 2.4: Simplified Floorplan of a Clock Region of an AMD UltraScale+ Device, derived from [63, 64].

Hard blocks occupy predefined locations within the FPGA and impose strong constraints on placement and connectivity. In DAQ systems, key functions such as timing distribution, high-speed data reception, and deterministic buffering are therefore inherently tied to specific silicon resources. As a result, architectural decisions concerning these functions must be made early in the development.

The programmable logic is interconnected through a hierarchical routing network that enables signal transport across the fabric. While this network provides substantial flexibility, it is a finite and shared resource whose utilisation is strongly influenced by the placement of hard blocks, clock-region boundaries, and locality constraints. In large or high-speed designs, timing closure limited by routing delay and congestion often becomes the dominant feasibility constraint, rather than exhaustion of logic resources.

Taken together, these architectural elements directly shape the achievable system-level characteristics of FPGA-based DAQ back-ends. The number and performance of MGTs determine the attainable aggregate data bandwidth and the structure of external interfaces, while on-chip memories such as BRAM and HBM constrain buffering depth and latency during trigger decision and data aggregation. The available CLB fabric and DSPs slices define the computational capacity for real-time processing tasks, including filtering, correlation, and feature extraction. Finally, the organisation and utilisation of routing resources govern timing closure and ultimately limit feasible design complexity. These couplings between silicon architecture and system-level parameters explain why FPGA-based DAQ designs require early architectural commitment.

2.2.3 Device Families and System Integration

According to [65], as of June 2025, approximately $85\,\%$ of the global FPGA market is dominated by AMD (formerly Xilinx) and Altera (formerly part of Intel), with market shares of $55\,\%$ and $30\,\%$, respectively. These two vendors are currently the only significant competitors in the high-end segment of the FPGA landscape, supplying devices for data centres, high-speed communications and advanced embedded applications. In contrast, Lattice Semiconductor and Microchip Technology (formerly Microsemi) primarily target the mid- to low-end sectors, offering cost-efficient and power-sensitive solutions for industrial, automotive and consumer electronics markets.

Selecting an FPGA involves balancing available logic resources, memory capacity, transceiver count, and cost. Table 2.1 summarises key characteristics of AMD UltraScale+ FPGA families, illustrating the scaling of resources across device classes. Altera offers devices in comparable categories, but as most processing boards built for CMS use AMD, this work will focus on this FPGA vendor.

Table 2.1: Overview of AMD UltraScale+ FPGA devices.

Resource	**Spartan**		**Artix**		**Kintex**		**Virtex**	
	Min	Max	Min	Max	Min	Max	Min	Max
CLB LUTs (k)	5	100	37	141	163	842	394	4096
Block RAM (Mb)	1.7	6.8	3.5	10.5	12.7	60.8	23.6	94.5
Ultra RAM (MB)	0	18	N/A		0	81	90	360
HBM (GB)	N/A		N/A		N/A		0	16
GTH Transceivers	0	8	0	12	0	44	0	0
GTY Transceivers	N/A		0	12	0	32	32	128
GTM Transceivers	N/A		N/A		N/A		0	48

GTH: $16.3\,\mathrm{Gbit\,s^{-1}}$; GTY: $32.75\,\mathrm{Gbit\,s^{-1}}$; GTM: $58\,\mathrm{Gbit\,s^{-1}}$ PAM4
Zynq UltraScale+ devices are not listed, as they are System-on-Chips (SoCs) and thus differ substantially from pure FPGA devices.

Modern FPGA devices are increasingly offered as part of SoC architectures [66], combining reconfigurable logic with embedded processors, dedicated acceleration engines, and high-bandwidth on-chip interconnects such as Network-on-Chip (NoC) fabrics. While such integration reduces board-level component count and can simplify certain system interfaces, it does not remove the underlying architectural constraints of the programmable fabric. Instead, these constraints are resolved at different integration boundaries, shifting rather than eliminating architectural commitment.

2.2.4 Applications of FPGAs in High-Energy Physics DAQ

Data acquisition systems in high-energy physics operate under constraints that exceed the capabilities of commercial off-the-shelf computing platforms. In particular, sub-nanosecond timing requirements with jitter budgets in the order of $1.5\,\mathrm{ps}$, experiment-specific backplane and optical connectivity, and sustained multi-terabit-per-second data rates cannot be met by standard server or accelerator hardware. These constraints motivate the development of custom processing platforms, such as the Serenity-S1 ATCA card, that are tailored to the electrical, mechanical, and timing requirements of large-scale detector back-ends.

Within this context, FPGAs play a central role in modern HEP experiments. In experiments such as ALICE [67], ATLAS [68], CMS [43], and LHCb [69] at the LHC, FPGAs and FPGA-based SoCs are deployed as the first stage of back-end processing, where detector data are received, buffered, formatted, and reduced in real time. They perform latency-critical tasks including event filtering, zero suppression, data aggregation, and feature extraction before transferring data to downstream computing farms composed of commercial server hardware [42].

The use of custom FPGA-based processing platforms introduces substantial system-level complexity. Firmware development requires specialised expertise, long validation cycles, and close coordination between hardware, firmware, and software teams. In large, geographically distributed collaborations, maintenance, debugging, and long-term support further increase the engineering effort associated with such systems. These characteristics are not incidental but are an inherent consequence of combining architectural programmability with strict real-time and integration constraints.

2.3 Modular Hardware in Electronics

Modern electronic systems – including DAQ systems – are rarely realised as monolithic artefacts. Instead, they are typically composed of multiple functional units whose separation enables reuse, parallel development, and controlled system evolution. In hardware design, however, the term *modularity* is applied across a wide range of abstraction levels, from complete computing platforms down to fine-grained circuit structures. Without explicitly distinguishing these levels, discussions of modular hardware risk conflating fundamentally different design approaches and obscuring their respective trade-offs.

2.3.1 Levels of Modularity

In this work, three levels of hardware modularity are distinguished according to the integration boundary at which functional separation is introduced:

- **System-level modularity**, where independently operable hardware units – such as plug-in cards or processing nodes – are integrated via standardised mechanical, electrical, and protocol interfaces. Typical examples include backplane-based platforms such as ATCA [70] or VPX [71], as well as bus-oriented ecosystems based on PCI Express [72], where interoperability and replacement are governed by external standards.
- **Board-level modularity**, where functional subsystems are implemented as replaceable assemblies within a larger electronic unit, commonly using mezzanine cards or daughter boards. While mechanical form factors may be standardised, electrical interfaces, timing assumptions, and power domains are often application-specific, and integration remains under the control of a single system design authority.
- **Intra-board modularity**, where functional separation is realised within a single printed circuit board through reusable schematic and layout blocks that expose stable electrical interfaces while sharing a common physical implementation context.

All three forms of modularity are well established in electronics engineering practice, yet they differ substantially in achievable performance, integration overhead, and flexibility under physical constraints. System- and board-level modularity primarily facilitate interoperability and scalability between independently manufactured units, but they introduce fixed interface boundaries that can limit bandwidth density, timing precision, and signal integrity. Fully monolithic board designs, by contrast, maximise physical optimisation but scale poorly in terms of development effort, validation complexity, and long-term maintainability.

High-rate data acquisition back-ends operate under particularly stringent constraints on latency, bandwidth, synchronisation, and signal integrity. These constraints limit the applicability of coarse-grained modular architectures without incurring unacceptable performance penalties, while simultaneously rendering purely monolithic approaches increasingly difficult to sustain over long project lifetimes. This tension motivates an examination of modularity at finer abstraction levels, where reuse and architectural consistency can be introduced without relinquishing control over physical implementation.

The following subsections therefore examine the benefits and limitations of modular hardware, draw on established modular design paradigms from related engineering disciplines, and position different forms of hardware modularity relative to existing industry standards. This establishes the context for modular design approaches that operate below the level of system platforms and

focus on modularity within individual processing boards under the constraints characteristic of high-performance DAQ systems.

2.3.2 Advantages and Disadvantages of Modular Hardware

Modular hardware architectures are commonly adopted in complex electronic systems to manage development effort, support technology evolution, and enable parallel work across teams [73, 74]. By decomposing a system into self-contained units with defined interfaces, individual modules can be developed, validated, and replaced independently. This approach improves maintainability and facilitates long-term system evolution, particularly in projects with extended operational lifetimes.

At the same time, modularity introduces additional interfaces – electrical, mechanical, thermal, and organisational – that must be carefully specified and maintained. Each interface represents a potential performance bottleneck and a source of integration risk. In high-performance electronics, interface overheads can directly translate into increased latency, reduced bandwidth, or degraded signal integrity [75, 76].

Table 2.2: Comparison between modular and monolithic hardware architectures.

Criterion	Modular Architecture	Monolithic Architecture
Design flexibility	High; modules can be reconfigured or replaced independently	Low; modifications require redesign of the entire system
Integration complexity	Requires careful interface management and system coordination	Minimal; subsystems integrated directly
Performance	May suffer from bandwidth and latency limits at module interconnects	Typically optimised for maximum throughput and timing
Maintainability	Excellent; modules can be serviced or upgraded individually	Limited; integrated design complicates maintenance
Scalability	Supports incremental expansion and technology upgrades	Fixed capacity determined at design time
Development efficiency	Reuse of proven modules enables shorter design cycles and parallel work	Entire system must be developed as a whole
Cost structure	Lower initial design cost but higher per-unit expense due to duplicated interfaces	Economical for large production volumes

Table 2.2 summarises the principal trade-offs between modular and monolithic hardware architectures as described in systems engineering literatue [73, 74]. Beyond technical considerations, these trade-offs also affect lifecycle management, verification effort, and cost structure.

While modularity offers clear advantages in flexibility, reuse, and maintainability, these benefits are not uniformly available across all abstraction levels. In particular, modularisation at coarse physical boundaries – such as pluggable boards or external interconnects – often incurs unacceptable penalties in timing precision, bandwidth density, and power efficiency for high-rate DAQ systems [75]. Conversely, fully monolithic designs maximise performance but scale poorly in terms of development effort, validation complexity, and long-term maintainability.

The central insight motivating this work is that these trade-offs are not binary. By applying modularity at a finer granularity – within a single PCB – many of the benefits of modular design can be retained while avoiding the dominant performance penalties associated with coarse-grained modular architectures. This intra-board approach enables reuse and architectural consistency without relinquishing control over physical implementation.

2.3.3 Modular Development in Other Fields

The concept of modular development is well established across engineering disciplines, most prominently in software engineering, systems engineering, and very-large-scale integration (VLSI) design. Each of these fields uses modularity as a strategy for managing complexity, improving maintainability, and enabling systematic reuse. Because modularisation addresses fundamental issues such as information hiding, change propagation, and verification, insights from these domains provide a theoretical foundation for modular schematic and PCB-level design.

A central contribution from software engineering is the distinction between *eliminating* and *encapsulating* complexity, as articulated by Osterhout [32]. Ousterhout emphasises that *deep modules* – those with substantial internal complexity but compact external interfaces – offer the highest payoff for long-term maintainability. This perspective highlights two principles that translate directly to hardware design: a module should hide internal design details behind a stable interface, and its external behaviour should remain unaffected by internal changes.

Complementary guidance arises from the SOLID methodology introduced by Martin [33]. Although conceived for object-oriented software, the SOLID principles formalise general design concepts that are domain-independent: single responsibility, interface clarity, separation of concerns, and the avoidance of unnecessary coupling. These principles articulate why modules should remain focused, replaceable, and extendable without modification – properties equally relevant for hardware blocks, power-tree elements, or communication interfaces.

In addition to these conceptual frameworks, the *Don't Repeat Yourself* (DRY) principle [34] provides a practical rationale for avoiding duplicated implementations. Duplication leads to divergence, inconsistent bug fixes, and fragmented documentation. DRY therefore motivates reusable schematic blocks, parametrised module variants, and version-controlled libraries.

Beyond software, the field of *Component-Based Design* (CBD) [35, 77] offers a close parallel to hardware modularisation. CBD defines systems as compositions of reusable components with well-specified interfaces and controlled variability. It is widely used in mechatronics, embedded systems, and cyber–physical systems. CBD emphasises interface contracts, configuration management, and verified reuse – concepts that map directly onto schematic modules, pin-level interface checks, and versioned component libraries in PCB design.

Systems engineering standards provide further formal justification. The INCOSE Systems Engineering Handbook [78] and ISO/IEC/IEEE 15288 [79] both require the definition of modular system elements, traceable configuration baselines, stable interface definitions, and reusable verification artefacts. These standards recognise that modularity is essential not only for managing complexity, but also for ensuring traceability and repeatability across a system's life cycle.

A parallel example arises in VLSI design, where hierarchical hardware description languages such as VHDL and Verilog encapsulate functionality into reusable blocks. Verified modules can be reused across projects or offered as intellectual property (IP) [80], demonstrating both technical and organisational advantages of modularity. Although schematic-level PCB design operates at a different abstraction level, the principles of encapsulation, verified reuse, and hierarchical composition remain applicable. This correspondence, however, is strongest in digital VLSI, where modular workflows are fully formalised; analog circuits largely remain tied to GUI-driven, geometry-centric design flows [81], underscoring the need for more systematic modularity at higher abstraction levels such as PCB and schematic design.

Across disciplines, modular development serves a common purpose: managing complexity through comprehensible, maintainable, and reusable system structures. The principles articulated by Ousterhout, SOLID, DRY, Component-Based Design, and Systems Engineering provide the conceptual and methodological basis for the Modular-Hardware Toolbox and its application to schematic- and PCB-level design (chapter 4).

2.3.4 System-Level Modularity in Hardware Standards

Industry standards play a central role in establishing modularity at the system level by defining common mechanical envelopes, electrical interfaces, and management infrastructures. Standards such as ATCA [70] exemplify this approach by enabling independently developed processing boards to be integrated into scalable, multi-vendor systems. Through standardised backplanes,

power distribution schemes, cooling concepts, and management protocols, such standards decouple system integration from payload design and support long-term operability at the system boundary.

System-level hardware standards intentionally delimit the scope of modularity they address. In order to remain broadly applicable and tractable, they define abstraction boundaries between hardware units – such as boards or endpoints – while deliberately leaving the internal organisation of those units unconstrained. ATCA, for example, specifies how boards interface with the surrounding system but remains agnostic to the internal architectural decomposition, schematic structure, and layout organisation of the payload itself. A similar principle applies to interface-centric standards such as PCI Express [72], which define scalable, high-performance communication semantics between endpoints without prescribing how functionality is structured within an endpoint.

As a consequence, existing hardware standards institutionalise modularity between hardware units, but not within them. The internal design space below the standard-defined abstraction boundary – including functional partitioning, reuse of circuit blocks, and consistency of schematic and layout structures – remains the responsibility of individual projects and design teams. This openness is a necessary characteristic of widely adopted standards, as it allows diverse applications and performance requirements to be accommodated without over-constraining implementations.

In modern, long-lived DAQ systems, development effort is increasingly dominated by limited personnel resources, extended iteration cycles, and the cumulative overhead of verification and maintenance. Within this context, the unstructured internal design space below the system-level abstraction boundary becomes a critical factor influencing development time and engineering throughput. Addressing this challenge does not require extending or replacing existing hardware standards; rather, it motivates the introduction of complementary design methodologies that apply modularity at a finer granularity.

The Modular-Hardware Toolbox developed in this thesis targets this unaddressed design space by applying modular design principles at the intra-board level. Instead of pluggable hardware units, modularity is realised through reusable schematic and layout building blocks with well-defined interfaces and associated verification status. This approach complements system-level standards by enabling reuse, architectural consistency, and scalable development processes within a single printed circuit board, without conflicting with the intended scope or abstraction boundaries of existing hardware platforms.

2.4 The CMS Experiment in HL-LHC

The CMS experiment provides the operational environment for the hardware developments presented in this work. As one of the principal detectors at the High-Luminosity Large Hadron Collider (HL-LHC), CMS imposes uniquely demanding requirements on data acquisition electronics in terms of bandwidth, latency, radiation tolerance and long-term maintainability. Understanding these experimental constraints is essential to motivate the design and architecture of the Serenity-S1 ATCA processing platform, which serves as a modular back-end solution deployable across multiple CMS subsystems. This section summarises the experiment's layout, the HL-LHC upgrade programme, and the resulting data acquisition requirements that define the context for the Serenity-S1 development.

2.4.1 CMS Experiment

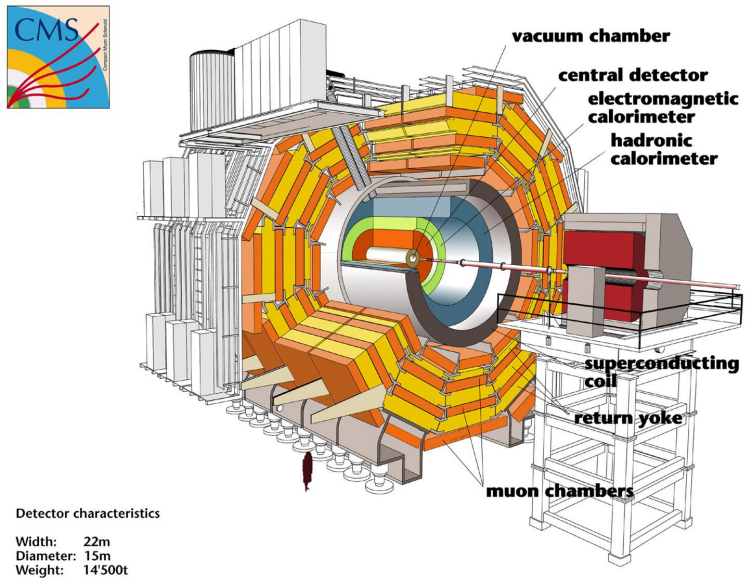


Figure 2.5: Cutaway view showing the outer four layers for detecting muons (interleaved with three layers of iron), the central calorimeters and the inner tracking system of the CMS detector [82].

One of four principal experiments at the LHC, which is currently the highest-energy particle accelerator in the world, is the CMS experiment. Situated near Cessy, France, along the Franco-Swiss border, CMS is positioned on the LHC's 27 km circular tunnel to perform proton–proton

collisions at centre-of-mass energies up to 14 TeV. As a general-purpose detector, CMS incorporates several subdetectors – tracking systems, calorimeters and muon chambers – each optimized to measure specific products of high-energy collisions, such as leptons, photons and jets [83, 84, 85].

This experiment is central to advancing our understanding of fundamental physics. Designed as a general-purpose, high-precision detector, CMS plays a crucial role in testing the Standard Model and searching for physics beyond known frameworks [86].

One of CMS's most significant achievements, so far, was its contribution to the discovery of the Higgs boson, a breakthrough in particle physics. In 2012, the CMS and ATLAS collaboration observed a new boson near 125 GeV, later confirmed as the Higgs boson, validating the mechanism of electroweak symmetry breaking as predicted by the Standard Model [87, 88].

Beyond the Higgs discovery, CMS is deeply involved in searches for supersymmetry. By analysing missing transverse energy signatures and high-mass final states, the experiment tests models that extend the Standard Model's particle spectrum, offering potential explanations for dark matter and other unresolved mysteries in particle physics [89]. Additionally, CMS explores physics beyond the Standard Model (BSM), investigating phenomena such as heavy resonances, extra dimensions and exotic particles that could indicate new fundamental forces or interactions [90].

2.4.2 CMS Subsystems

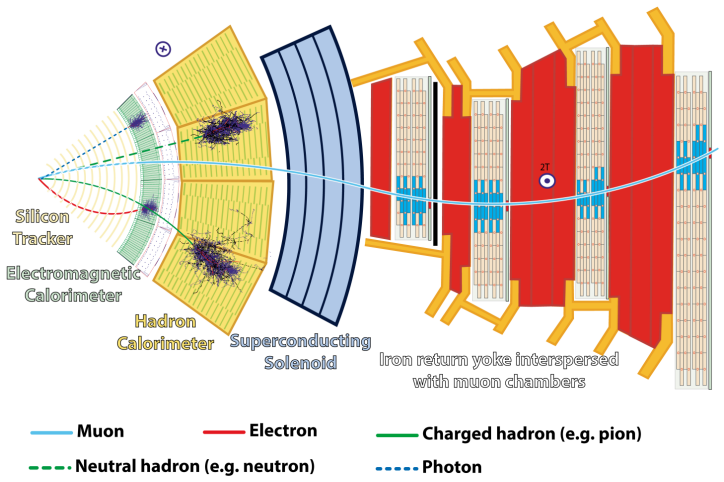


Figure 2.6: Transverse cross-section of the CMS detector, indicating its layered subdetectors and their measurement domains [91].

This subsection summarises the baseline detector structure of the CMS experiment and introduces the major subdetector systems that define the sources and characteristics of data entering the trigger and data acquisition chain. Phase-2–specific extensions and replacements are discussed separately in subsection 2.4.3.

The CMS detector features a multi-layered architecture, with each layer optimised to detect specific types of particles (see Figure 2.6). The innermost layer is the tracker, followed by the electromagnetic and hadronic calorimeters. Surrounding these is the superconducting solenoid, which gives the CMS detector its name and generates a magnetic field of $3.8\,\mathrm{T}$. This field bends the trajectories of charged particles via the Lorentz force, enabling precise momentum measurements. The outermost component is the muon system, positioned externally because muons can traverse the inner detector layers with only minimal interaction.

The tracker utilises silicon pixel and microstrip detectors to trace the trajectories of charged particles. The detector modules are arranged in a cylindrical geometry around the beam pipe in the barrel region, while disk-shaped structures are used in the forward and endcap regions. The tracker measures only the spatial position of charged particles; momentum reconstruction is performed by fitting trajectories in the magnetic field. Hit information from the tracker is transmitted to the detector back-end systems, where it is buffered and forwarded for further processing [20].

Calorimeters work by fully absorbing particles and thus enabling the measurement of their energy. There are two distinct types of calorimeters, homogenous calorimeters and sampling calorimeters. For homogenous calorimeters, the detector material is at the same time the absorbing and sensing material. Sampling calorimeters consist of alternating layers of passive absorbers and active detectors. Many calorimeters work using scintillators that emit light proportional to the energy deposited. This light is converted into electrical signals using photodiodes and finally read out by an ADC.

In CMS, the first calorimeter is the Electromagnetic Calorimeter (ECAL), a homogenous calorimeter. It is composed of lead tungstate ($PbWO_4$) crystals that act as a scintillator to measure the energy of electrons, positrons and photons. Surrounding the ECAL is the Hadronic Calorimeter (HCAL), which detects hadrons by absorbing their energy through interactions with dense materials. Depending on the region of the detector, brass or steel is used as passive absorbers with plastic scintillator tiles as active absorbers. In the Phase-2 CMS upgrade, the High-Granularity Calorimeter (HGCAL) replaces the ECAL and HCAL endcap detectors. It is a highly segmented silicon-based sampling calorimeter, with lead and steel absorbers, providing highly granular spatial information and timing with a $30\,\mathrm{ps}$ resolution.

Encircling these layers is a superconducting solenoid magnet with a diameter of $6\,\mathrm{m}$ that generates a $3.8\,\mathrm{T}$ magnetic field. This field is uniform along the barrel region but non-uniform in the endcap region and is applying the Lorentz force on charged particles, bending their trajectories.

The outermost layer is used to measure muons, as they traverse the inner layers of the CMS detector with only minimal energy loss. The Muon system uses Drift Tubes (DTs) and Resistive Plate Chambers (RPCs) in the barrel region. The drift tubes allow tracking of the muons while the RPCs provide precise timing. Due to higher occupation and radiation levels, drift tubes can not be used in the forward and endcap regions, hence Gas Electron Multipliers (GEMs) and Cathode Strip Chambers (CSCs) are used for tracking there.

2.4.3 Phase-2 Upgrade and HL-LHC Requirements

The High-Luminosity LHC (HL-LHC) upgrade aims to extend the LHC's discovery reach by increasing its instantaneous luminosity to approximately $7.5 \times 10^{34}\,\mathrm{cm^{-2}\,s^{-1}}$, corresponding to an integrated luminosity of $3000\,\mathrm{fb}^{-1}$ over a decade of operation [15]. This tenfold increase in collision rate enhances statistical sensitivity but also imposes unprecedented challenges on detector readout and data acquisition. At design luminosity, up to 200 simultaneous proton–proton interactions per bunch crossing (*pile-up*) must be reconstructed, demanding sub-nanosecond timing precision, high detector granularity, and multi-terabit-per-second data throughput.

CMS addresses these challenges through the Phase-2 upgrade programme [18], which introduces substantial modifications to the baseline detector systems described in the previous subsection:

- **Tracker:** A complete replacement with radiation-hard silicon modules providing precise tracking information at $40\,\mathrm{MHz}$. The new tracker integrates a hardware-based track trigger capable of track reconstruction within the L1T latency window.
- **Calorimetry:** Replacement of the endcap ECAL and HCAL with the HGCAL, a highly segmented silicon-based sampling calorimeter providing fine spatial granularity and precise timing for pile-up mitigation.
- **Timing Layer:** Introduction of the Minimum Ionizing Particle Timing Detector (MTD), which provides per-particle timing with a resolution of approximately $30\,\mathrm{ps}$, enhancing event reconstruction under high pile-up conditions.
- **Muon System:** Extension of forward coverage through additional RPC and GEM detectors to improve efficiency.

- **Trigger and DAQ:** A substantial increase in aggregate data throughput from approximately $2\,\mathrm{Tbit\,s^{-1}}$ in Run 2 to beyond $60\,\mathrm{Tbit\,s^{-1}}$ in Run 3 (Phase-2). By introducing larger on-detector and back-end buffers, the L1T latency is extended to $12.5\,\mu\mathrm{s}$, enabling more sophisticated multi-subsystem correlation algorithms [42].

The diversity and scale of these upgrades require a new generation of modular back-end electronics capable of high throughput, flexible configuration, and long-term maintainability across detector domains. The Serenity-S1 platform developed in this work directly addresses these needs, serving as a reusable processing architecture deployable in subsystems including the Tracker, HGCAL, Muon, and Trigger systems.

2.4.4 Trigger and DAQ Architecture

The CMS detector subsystems described above generate continuous data streams at rates that far exceed the capabilities of permanent storage and offline processing. At the HL-LHC bunch crossing frequency of $40.079\,\mathrm{MHz}$, only a small fraction of events can be retained for detailed analysis. The trigger and data acquisition (DAQ) system therefore provides a real-time selection and data reduction infrastructure that enables the experiment to operate within strict bandwidth and latency constraints while preserving sensitivity to rare physics processes.

Detector front-end electronics continuously transmit digitised data over high-speed optical links to off-detector back-end systems. These back-end processing nodes aggregate data from multiple detector modules, provide short-term buffering during trigger decision latency, and perform real-time processing steps required for event selection. Depending on trigger decisions, data are either discarded or forwarded as event fragments to higher-level processing stages. This data flow architecture must accommodate both continuous streaming operation and burst-like data transfer following trigger accepts, placing stringent requirements on buffering capacity and deterministic data handling.

In the Phase-2 CMS architecture, event selection is organised as a two-stage hierarchy consisting of a hardware-based L1T followed by a software-based HLT. The L1T operates as a hard real-time system, processing data synchronously with the LHC bunch crossings and reducing the input rate from $40.079\,\mathrm{MHz}$ to an average accept rate of approximately $750\,\mathrm{kHz}$. To enable more sophisticated trigger algorithms, the Phase-2 upgrade extends the maximum L1T latency to $12.5\,\mu\mathrm{s}$, necessitating deep buffering and fully deterministic processing pipelines in the back-end electronics [42].

Events accepted by the L1T are transferred to the HLT, which executes more complex reconstruction and selection algorithms on a farm of 110 commercial 48U server racks [43]. The HLT further reduces the event rate by roughly two orders of magnitude, resulting in a final output rate

of approximately $7.5\,\mathrm{kHz}$. With typical event sizes of $5.9\,\mathrm{MB}$, this staged reduction lowers the aggregate data rate from $2.7\,\mathrm{Pbit\,s^{-1}}$ at the detector output to $354\,\mathrm{Gbit\,s^{-1}}$, a level compatible with long-term storage [43].

A defining architectural change introduced with the Phase-2 upgrade is the integration of tracking information into the L1T. The Level-1 Track Trigger performs real-time reconstruction of charged-particle tracks directly from tracker front-end data at the full collision rate. The hit information is transmitted to the back-end, where tracks are reconstructed within a latency budget of a few microseconds. The resulting track parameters are then correlated with calorimeter and muon trigger information in the correlator stage to refine transverse-momentum and vertex selection before a global trigger decision is formed [42, 92].

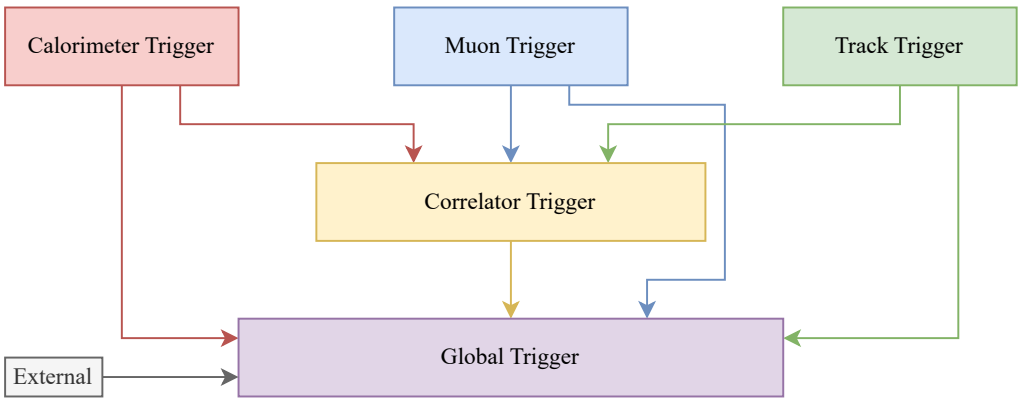


Figure 2.7: Overview of the Phase-2 CMS Level-1 trigger architecture, showing the calorimeter, muon, and track triggers feeding into the correlator and global trigger stages. Adapted from [42].

Within the Phase-2 trigger and DAQ architecture, the more than 1200 back-end processing cards serve as convergence points where data from multiple detector subsystems is received through approximately $50\,000$ front-end links, buffered, processed, and forwarded under strict real-time constraints [43]. These nodes must simultaneously support high optical I/O density, deep and predictable buffering, deterministic processing pipelines, and scalable inter-node communication. The architectural demands imposed by online tracking and extended latency windows further reinforce the need for flexible yet highly controlled hardware platforms at this level.

The CMS detector and the HL-LHC upgrade programme define one of the most demanding environments for real-time data acquisition in experimental physics. Extreme input data rates, tight latency constraints, and the requirement for long-term operational stability collectively motivate the use of modular, high-performance back-end electronics. The Serenity-S1 ATCA platform presented in this dissertation is a processing cards developed within this context, providing a scalable hardware foundation for Phase-2 CMS trigger and data acquisition systems.

3 Analysis of Requirements

This chapter establishes the requirements that define the scope, functionality, and evaluation criteria of the Modular-Hardware Toolbox developed in this work. It does so by analysing the evolution of data acquisition (DAQ) electronics in high-energy physics, identifying both technological and organisational challenges that shape modern back-end systems. The analysis begins with a review of existing architectures within the CMS Phase-2 upgrade and related experiments, traces broader trends in DAQ development, and extrapolates requirements for future collider systems. In addition to qualitative architectural insights, this chapter introduces a quantitative evaluation framework based on the Capability Maturity Model Integration (CMMI), which is adapted here to assess and benchmark PCB design process capability. Together, these analyses define not only what the Modular-Hardware Toolbox must enable, but also how its performance can be objectively evaluated in later chapters.

3.1 Review of Electronics for the CMS Phase-2 Upgrade

The CMS Phase-2 upgrade provides a recent and representative sample of state-of-the-art back-end electronics and therefore serves as an empirical basis for deriving requirements for a modular development methodology. Although the boards target different subsystems, they implement closely related functions and can be ordered by the degree to which modularity is realised: monolithic single-board designs such as BMTL-1 [26] and APx [27]; partially modular approaches such as Apollo [29, 93] and X2O [30]; and Serenity-S1 [2, 3], which applies modularity primarily at the schematic and layout level through reuse-enabled design workflows. The coexistence of multiple, functionally similar designs indicates that consolidation and reuse are limited not only by technical constraints but also by organisational conditions. This section reviews the Phase-2 boards to extract common architectural patterns, degrees of modularity, and collaboration constraints that motivate a unified, reusable hardware development framework.

3.1.1 Overview of ATCA Boards for Phase-2

Despite being developed by different collaborations and targeting distinct detector subsystems, the ATCA-based Phase-2 back-end boards follow a remarkably consistent architectural template: a high-end FPGA for data processing, high-speed optical transceivers for I/O, and a secondary controller – typically a Zynq UltraScale+-based System-on-Module (SoM) – for slow control and monitoring. Differences are primarily driven by subsystem-specific adaptations, such as optical lane count, timing distribution, or the power-delivery network.

The boards are introduced below in order of increasing modularity; Serenity-S1 is discussed first because it provides the reference implementation used throughout this dissertation. While physically monolithic, Serenity-S1 exemplifies modularity at schematic and layout level through reuse-enabled design blocks that support distributed development across institutes. The following paragraphs summarise the Phase-2 boards and their distinguishing features; subsection 3.1.2 then extracts common patterns and their organisational implications.

Serenity-S1 The *Serenity-S1* [2, 3] ATCA card, described in detail in chapter 5, integrates an AMD[1] Virtex UltraScale+ VU13P FPGA, 124 Samtec FireFly optical lanes, and an AMD Kria K26 SoM for board management on a single PCB. Within the Phase-2 upgrade, Serenity-S1 functions as a high-bandwidth detector back-end node, serving the Outer Tracker (OT), HGCAL, parts of the muon system, and the MTD, as well as the Beam Radiation, Instrumentation, and Luminosity (BRIL) and L1T systems. Although physically monolithic, Serenity-S1 was developed using modular design principles enabled by the Modular-Hardware Toolbox, facilitating distributed schematic and layout work across institutes.

BMTL-1 The *BMTL-1* ATCA card [26] is dedicated to muon trigger processing in the barrel region. A single VU13P FPGA handles data from DTs and RPCs, with Samtec FireFly modules providing optical links and a Zynq UltraScale+ SoM managing control functionality. Like Serenity-S1, all components are integrated on a single PCB.

APx Building on the same monolithic principle, a similar single-FPGA architecture is also adopted in the *Advanced Processor (APx)* [42, 27], which targets calorimeter trigger applications. It integrates a VU13P device with 124 lanes to Samtec FireFly connectors, board management via a Kria K26 SoM, and a Zynq-7000-based intelligent platform management controller (IPMC). Older versions used the Embedded Linux Mezzanine (ELM) [94] instead of the Kria SoM.

[1] Xilinx was acquired by AMD in 2022. FPGAs released before the acquisition are referred to as AMD devices for consistency.

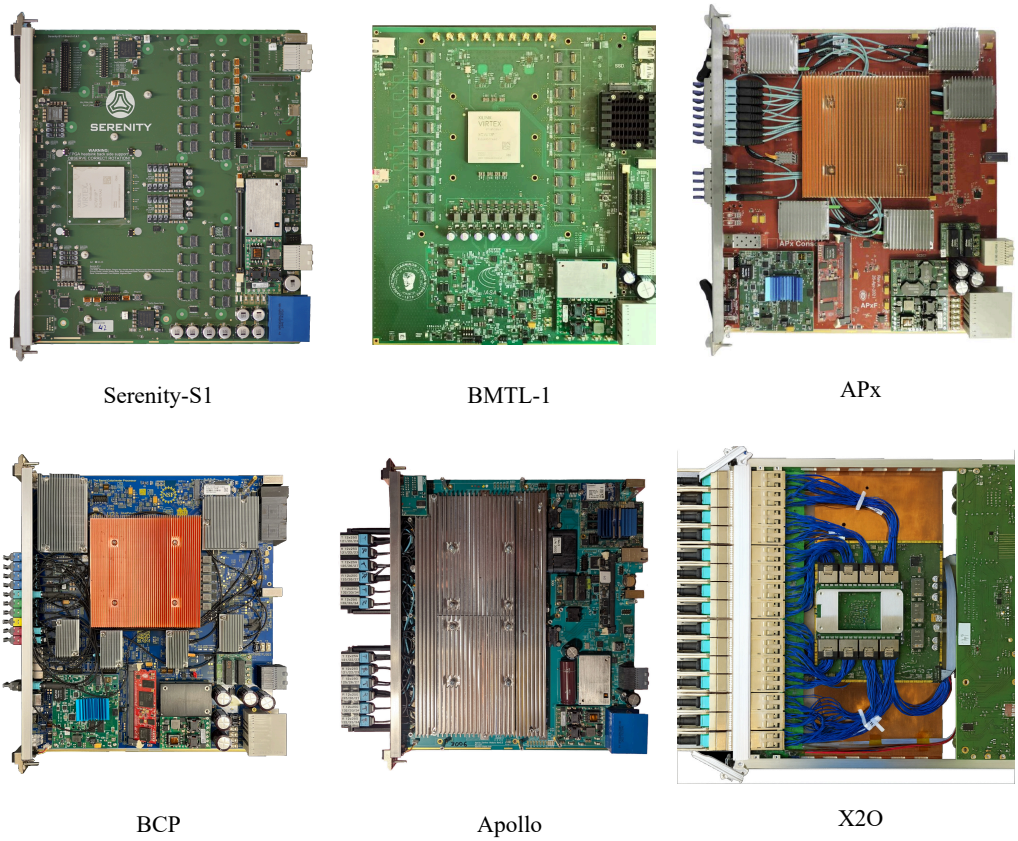


Figure 3.1: Comparison of CMS Phase-2 ATCA node cards, arranged by increasing modularity from monolithic to multi-module designs.

BCP The *Barrel Calorimeter Processor (BCP)* [28] also relies on a monolithic PCB and is closely related to the APx card, as both cards initially used the ELM. However, it features asymmetric data transmission and provides four high-speed lanes through an RTM module. Just like the APx, the BCP will also switch to the Kria K26 SoM in future revisions.

Taken together, Serenity-S1, BMTL-1, APx, and BCP show that CMS Phase-2 node cards predominantly realise a monolithic PCB integration, while the following two cards, the Apollo and X2O consist of interconnected modules. This allows these cards to easily adapt to different use cases.

Apollo The *Apollo* [93, 29] card represents a shift towards modularity, adopting a two-module architecture reminiscent of the Apollo spacecraft design. The *service module* hosts power conversion, clock distribution, and control functions via a Zynq UltraScale+ SoM, while the interchangeable *command module* integrates two VU13P FPGAs and FireFly optical interfaces for high-speed data processing. Designed primarily for the CMS track trigger and inner tracker DAQ, there is also a variation of the Apollo command module that was developed for ATLAS [95]. Apollo's modular approach allows parallel development between collaborating institutes even for multiple experiments and demonstrates how mechanical and electrical partitioning can facilitate distributed work.

X2O The concept of modularity is taken further with the *X2O platform* [30], which breaks down the ATCA card into three modules: a power module, an optical module, and the Octopus processing module. The power module offers backplane connectivity, regulated $12\,\mathrm{V}$ distribution, and board control via an AMD Kria K26 SoM. Notably, this board implements the IPMC functionality into the Kria SoM as an application running on Linux [96]. The optical module accommodates 30 Quad Small Form Factor Pluggable (QSFP) modules, interfaced to the processing device by high-speed twinaxial cables. The Octopus module hosts a lidless-packaged VU13P FPGA and a Lattice MachXO2 FPGA for slow control and board management. Structural rails and a common heatsink integrate the modules mechanically and thermally. X2O is intended to be used in the DAQ of the endcap muon subsystems (CSCs and GEMs) and the global muon trigger. Building on this modularity, the newer *Kraken* processing module incorporates an AMD Versal Premium VP1802 SoC, extending total theoretical throughput to $6.72\,\mathrm{Tbit\,s^{-1}}$ [96]. The throughput is only limited by the QSFP transceivers used on the optical module, which remained identical between Octopus and Kraken. This evolution exemplifies the possibilities and limitations of modular scalability in both hardware and performance.

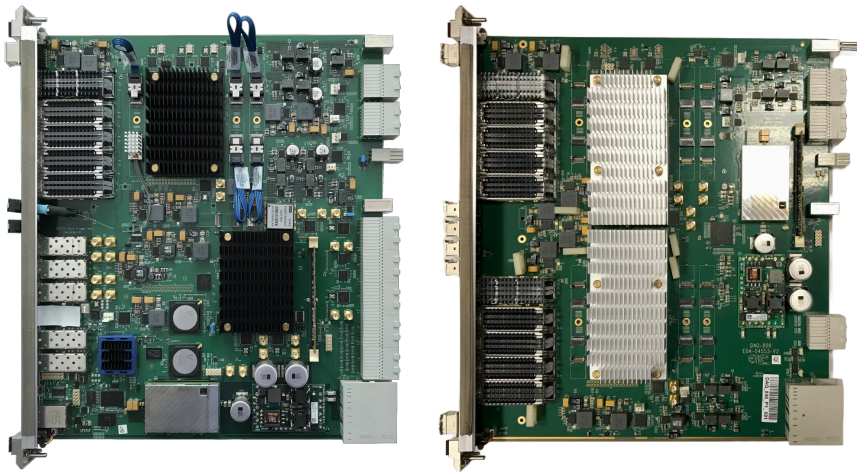
DTH400 and DAQ800 While the boards above serve as node cards, the *DAQ and Timing Hub (DTH)* [97, 98, 99] occupies the hub slot of the ATCA crate. It collects data from the back-end processing cards and sends it – if accepted by the L1T – to the HLT. Furthermore, the DTH400 distributes Phase-2 Timing and Control Distribution System (TCDS2) signals to each node via the backplane. The DTH400 employs two AMD Virtex UltraScale+ VU35P FPGAs – one dedicated to DAQ and one to TCDS2 – both integrating HBM to buffer and forward large data volumes. Board management is located on an Rear Transition Module (RTM) using a Kria K26 SoM. Optical communications utilise Samtec FireFly modules for back-end links and QSFPs for the Data-to-Surface (D2S) network. The DAQ bandwidth of up to $400\,\mathrm{Gbit\,s^{-1}}$ motivates the specific designation DTH400; a companion DAQ800 variant, replaces the TCDS2 FPGA to double the DAQ circuit, hence doubling throughput. The boards are depicted in Figure 3.2.

Table 3.1: Overview of the ATCA-based FPGA Boards in CMS Phase-2

Board	FPGA(s)	Board Management	IPMC
Serenity-S1	VU13P	Kria K26 SoM	OpenIPMC
Apollo	2× VU13P	Enclustra ME-ZX1 SoM	OpenIPMC
APx	VU13P	Kria K26 SoM	ZYNQ-IPMC
BCP	VU13P	Kria K26 SoM	ZYNQ-IPMC
BMTL-1	VU13P	Enclustra ME-XU5 SoM	CERN IPMC
X2O (Octopus)	VU13P	Kria K26 SoM	IPMC in Kria
X2O (Kraken)	VP1802	Kria K26 SoM	IPMC in Kria
DTH400	2× VU35P	Kria K26 SoM	CERN IPMC
DAQ800	2× VU35P	Kria K26 SoM	CERN IPMC

Table 3.2: Connectivity of ATCA-based FPGA boards in CMS Phase-2. High-speed connections to relevant endpoints are listed individually; connections to board management or backplane infrastructure are omitted.

Board	Tx-Bandwidth	Rx-Bandwidth	Connection
Serenity-S1	$3.10\,\mathrm{Tbit\,s^{-1}}$	$3.10\,\mathrm{Tbit\,s^{-1}}$	FireFly optical modules
Apollo	$2.60\,\mathrm{Tbit\,s^{-1}}$	$2.60\,\mathrm{Tbit\,s^{-1}}$	FireFly optical modules
	$1.30\,\mathrm{Tbit\,s^{-1}}$	$1.30\,\mathrm{Tbit\,s^{-1}}$	Inter-FPGA
APx	$3.10\,\mathrm{Tbit\,s^{-1}}$	$3.10\,\mathrm{Tbit\,s^{-1}}$	FireFly optical modules
BCP	$1.80\,\mathrm{Tbit\,s^{-1}}$	$3.00\,\mathrm{Tbit\,s^{-1}}$	FireFly optical modules
	$0.04\,\mathrm{Tbit\,s^{-1}}$	$0.04\,\mathrm{Tbit\,s^{-1}}$	SFP on RTM
BMTL-1	$1.58\,\mathrm{Tbit\,s^{-1}}$	$2.34\,\mathrm{Tbit\,s^{-1}}$	FireFly optical modules
X2O (Octopus)	$3.00\,\mathrm{Tbit\,s^{-1}}$	$3.00\,\mathrm{Tbit\,s^{-1}}$	QSFP optical modules
DTH400	$0.60\,\mathrm{Tbit\,s^{-1}}$	$0.60\,\mathrm{Tbit\,s^{-1}}$	FireFly (back-end)
	$0.50\,\mathrm{Tbit\,s^{-1}}$	$0.50\,\mathrm{Tbit\,s^{-1}}$	QSFP (Data-to-Surface)
DAQ800	$1.20\,\mathrm{Tbit\,s^{-1}}$	$1.20\,\mathrm{Tbit\,s^{-1}}$	FireFly (back-end)
	$1.00\,\mathrm{Tbit\,s^{-1}}$	$1.00\,\mathrm{Tbit\,s^{-1}}$	QSFP (Data-to-Surface)



DTH400 DAQ800

Figure 3.2: CMS Phase-2 ATCA hub cards, DTH400 and DAQ800.

Across the CMS Phase-2 ATCA platforms, a common architectural baseline is evident: high-end FPGA processing, high-speed optical interfaces, and an embedded management SoM based on an AMD Zynq UltraScale+ SoC (see Table 3.1 and Table 3.2). The main variations concern the power delivery network and the clock distribution of the cards.

3.1.2 Comparative Analysis of Board Architectures

Figure 3.3 illustrates an abstracted functional view of the ATCA systems described in subsection 3.1.1. The diagram highlights common roles – high-speed data processing, optical I/O, clocking, power delivery, and board management – while deliberately omitting implementation-specific details such as physical partitioning or exact interconnect topology.

At the implementation level, notable deviations exist. Apollo distributes processing across two FPGAs connected by a high-bandwidth inter-FPGA fabric, X2O separates power, optics, and processing into physically distinct modules, and Serenity-S1 implements low-jitter clock routing from the RTM to satisfy BRIL and MTD precision requirements below $15\,\mathrm{ps}$ [100, 101]. Nevertheless, these differences represent variations in integration strategy rather than fundamentally distinct architectural concepts.

Across all Phase-2 designs, the same functional elements recur and are combined in comparable ways. Subsystem-specific requirements are addressed through adaptations of clocking, power delivery, and I/O density, while the overall architectural structure remains highly convergent.

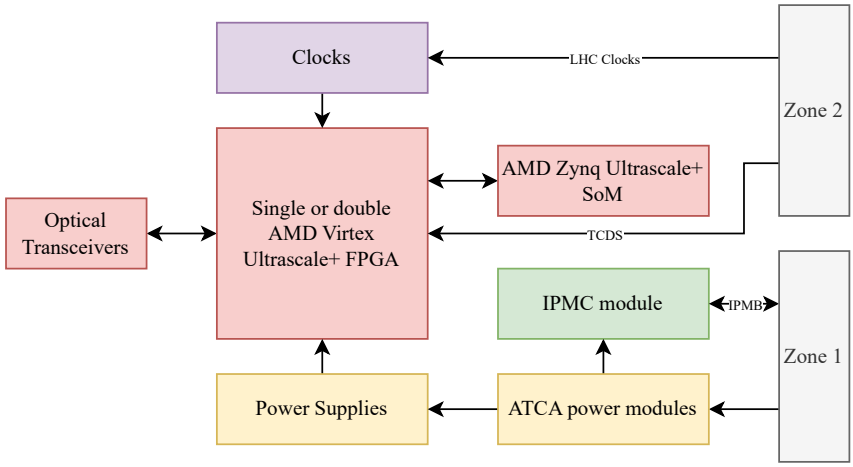


Figure 3.3: Abstracted functional structure of ATCA-based node-cards in CMS Phase-2, illustrating common functional blocks.

This convergence suggests that a reduced set of base designs, or a unified modular framework, could in principle support a broad range of back-end functions within CMS.

Given the strong architectural convergence and overlapping performance envelopes, the persistence of multiple, functionally similar Phase-2 boards cannot be explained primarily by technical necessity. Instead, it reflects organisational and funding structures within large-scale collaborations, where institutes are incentivised to deliver independent hardware contributions. In the absence of a shared hardware governance framework, parallel developments emerge as a stable equilibrium: they maximise participation and local ownership, even when they duplicate engineering effort.

3.1.3 Collaborative Development of ATCA Boards

While the Phase-2 boards share a common architectural baseline, they emerged from markedly different collaboration models shaped by the organisational structure of the CMS experiment. These models determine how work is divided across institutes, how ownership is established, and how effectively design effort can be shared. Three dominant collaboration modes can be identified:

1. independent institute-led board development,
2. physically modular co-development across institutes,
3. distributed co-design of monolithic boards.

Despite their functional similarity, the Phase-2 boards did not arise from centralised platform planning. Instead, they reflect the funding and responsibility structure of international HEP collaborations, in which national agencies expect visible, attributable deliverables from participating institutes. Joint ownership of a single hardware platform therefore offers limited incentive, while independent developments maximise local responsibility and recognition. In this context, duplication of functionally similar boards is tolerated as the cost of broad participation, rather than treated as an inefficiency to be eliminated.

Collaborative hardware development is also constrained by the capabilities of electronic design automation tools. Depending on the tool, schematic and layout collaboration required strict serialisation of work – one team finishing before another could proceed. This technical limitation reinforced monolithic design ownership and discouraged distributed layout development. High-end tools like Mentor Graphics Expedition or Cadence Allegro allow for concurrent editing of PCB layout. The adoption of reuse-enabled workflows allows independently routed schematic blocks to be merged systematically, even in tools that do not support concurrent editing, thereby enabling true multi-institute co-design.

Modular PCB partitioning enables parallelism by allowing institutes to work independently on well-defined functional domains – such as power-delivery, clock-tree, or high-speed I/O design – according to their respective expertise and laboratory infrastructure. This model aligns with institutional deliverable structures while maintaining overall system coherence. For instance, one institute may focus on signal-integrity and clock-jitter validation using high-bandwidth instrumentation, while another optimises the power-conversion network using electronic loads. Such modular work division scales effectively with board complexity and fosters cross-institute knowledge exchange without imposing centralised control.

The Apollo project exemplifies physically modular co-development: its separation into command and service modules enabled parallel design by different institutes while maintaining strict interface boundaries. Serenity-S1 demonstrates that comparable benefits can be achieved without physical modularisation. Through the use of reusable schematic and layout blocks, its design was partitioned across institutes despite being implemented as a single PCB. Serenity-S1 thus represents a hybrid model – monolithic in hardware, yet modular and distributed in process.

A closely related situation exists in large-scale ASIC development, where escalating non-recurring engineering costs and verification effort have long necessitated modular IP reuse, strict interface contracts, and mature process governance. While PCB design differs in fabrication cycle and physical constraints, the underlying drivers – complexity, collaboration at scale, and the cost of late-stage errors – are comparable. This parallel underscores that modularity is not merely a convenience but a prerequisite for sustainable hardware co-development at scale.

As future HEP detectors demand greater integration density, bandwidth, and timing precision, effective collaboration becomes a design constraint in its own right. Under these conditions, modularity is not optional: it is a key enabler for distributing work across institutes without fragmenting the resulting system. A shared modular hardware toolbox, composed of reusable and interface-compatible design blocks, offers a pragmatic mechanism to reconcile technical convergence with distributed governance. This insight directly motivates the methodological requirements developed in the following sections.

3.2 Evolution of DAQ Systems in High-Energy Physics

This section traces the architectural evolution of DAQ systems from early, subsystem-specific implementations to the highly integrated platforms foreseen for the HL-LHC and beyond. Using CMS as the primary case study and complementing it with examples from other experiments, the discussion identifies technological and organisational trends that directly inform the requirements for a scalable, reusable hardware development methodology.

3.2.1 Legacy DAQ Systems in the CMS Experiment

The data acquisition (DAQ) architecture used by the CMS experiment during Run 1 (starting in 2008) provides the historical baseline for understanding subsequent system evolution. Its design exemplifies an era in which subsystem-specific constraints and limited digital processing capability enforced a fragmented, highly specialised hardware landscape. Although technically effective, this generation of DAQ hardware embedded structural assumptions – tight coupling between detector subsystems and bespoke back-end boards – that would later constrain architectural unification.

During Run 1, most CMS DAQ systems employed the VMEbus standard, with dedicated hardware for each detector subsystem. This specialisation was not merely organisational but technologically necessary. Several factors contributed to this fragmentation:

- **Heterogeneous signal formats:** Subsystems employed incompatible readout technologies. The tracker relied on analog optical transmission, whereas calorimeter and muon systems performed local digitisation and transmitted digital data over copper or optical links. Each interface required dedicated receiver ASICs and custom decoding logic in the back-end.
- **Limited integration capability:** FPGAs available in the mid-2000s lacked the logic density and transceiver performance needed to implement complete DAQ pipelines on a

single device. As a result, back-end boards distributed functionality across multiple FPGAs and auxiliary components [24].

- **Stringent latency budgets:** The Level-1 Trigger (L1T) decision time of only $3.2\,\mu s$ necessitated fixed-function, low-latency ASICs in the front-end [102], leaving little flexibility for reconfiguration or firmware-based control.
- **Lack of applicable Large Electron-Positron Collider (LEP) experience:** The Large Electron-Positron Collider (LEP) detectors operated at data rates of only $8\,\mathrm{Mbit\,s^{-1}}$ [103, 104, 105], offering no scalable readout model for LHC conditions. As a result, CMS subsystems developed their DAQ hardware largely independently.

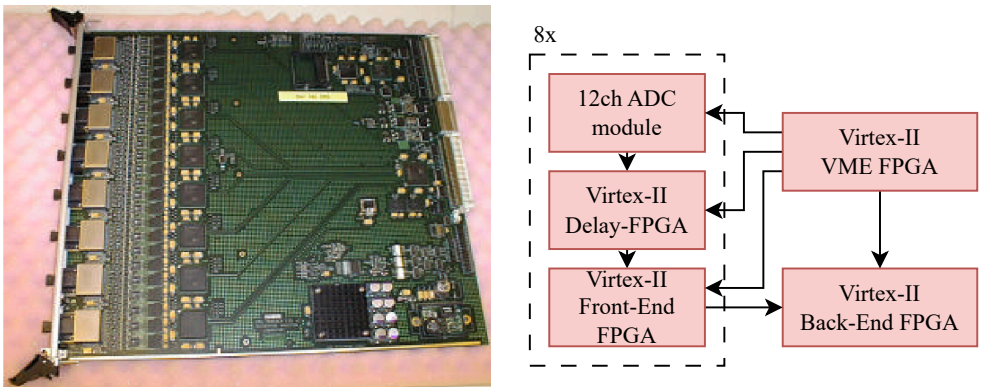


Figure 3.4: CMS Tracker FED VME card, representative of early subsystem-specific readout architectures employing multiple FPGAs and custom ASICs.

These technological constraints had architectural consequences. Because each subsystem required dedicated decoding, timing, and buffering hardware, the overall DAQ landscape became vertically integrated: each detector group maintained its own readout boards, firmware, and control software. This fragmentation hindered hardware reuse and created a strong path dependence in subsequent upgrade efforts. In effect, the physical limitations of VME and early FPGAs entrenched a modularity model based on *detector boundaries* rather than *functional abstraction*.

A decisive change occurred during the first long shutdown (LS1, 2014 – 2015) and the extended year-end technical stop (EYETS, 2016), when CMS began deploying new back-end electronics based on the Micro Telecommunications Computing Architecture (µTCA) standard. By this time, AMD 6- and 7-series FPGAs had introduced significantly larger logic resources and integrated multi-gigabit transceivers, while commercial optical modules such as the Avago MiniPOD and PPOD provided standardised I/O interfaces up to $10\,\mathrm{Gbit\,s^{-1}}$ [106, 107, 108, 109]. These

developments enabled a new generation of back-end boards capable of implementing complete DAQ and trigger pipelines entirely in a single FPGA.

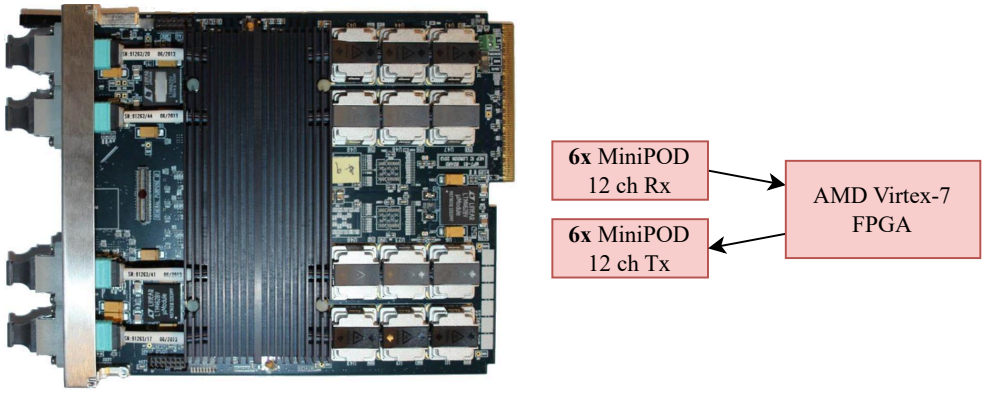


Figure 3.5: FC7 µTCA board, representative of the Run-2 back-end architecture.

The resulting µTCA boards – such as MP7, µHTR, TwinMux, and CTP6 – adopted a common architectural pattern: a central high-performance FPGA coupled with arrays of optical transceivers for flexible interconnects. As summarised in Table 3.3, the performance and composition of these boards were remarkably similar, reflecting an emergent convergence toward a general-purpose back-end platform rather than a collection of subsystem-specific designs.

Board	Optical Modules	Speed	Lanes (IO)	Processing FPGA
MP7	Avago MiniPOD	$10.3\,\mathrm{Gbit\,s^{-1}}$	72x TX 72x RX	AMD Virtex-7
CTP-6	Avago PPOD	$6.4\,\mathrm{Gbit\,s^{-1}}$	12x TX 48x RX	2x AMD Virtex-6
µHTR	Avago PPOD	$5\,\mathrm{Gbit\,s^{-1}}$	12x TX 24x RX	2x AMD Virtex-6
	SFP+	$10\,\mathrm{Gbit\,s^{-1}}$	2x TX 2x RX	
TwinMux	Avago MiniPOD	$10.3\,\mathrm{Gbit\,s^{-1}}$	12x TX 12x RX	AMD Virtex-7
	Smiths Interconnect SNAP12	$480\,\mathrm{Mbit\,s^{-1}}$	72x RX	

Table 3.3: Comparison of representative CMS Run-2 µTCA back-end and trigger cards, all featuring a central FPGA with high-speed serial I/O.

The transition from the parallel VME architecture to the serial µTCA backplane was enabled by a new generation of devices that provided substantially greater logic density and integrated multi-gigabit I/O. Conceptually, Run 2 electronics established a de facto architectural template that directly anticipated the ATCA-based systems of the CMS Phase-2 upgrade (section 3.1).

3.2.2 DAQ Developments in Other Experiments

The architectural evolution observed in CMS reflects a broader transformation across contemporary high-energy physics experiments. Although each collaboration faced distinct detector geometries and luminosity regimes, their DAQ systems converged toward similar design principles:

1. consolidation of distributed logic into powerful, integrated processing nodes,
2. adoption of high-speed serial interconnects and standardised form factors such as µTCA, ATCA, and PCIe, and
3. increasing reliance on software- and firmware-defined functionality.

A common enabling factor across these developments is the migration from analog to digital readout at the detector front-end [110], which standardises back-end interfaces and facilitates the reuse of processing platforms across subsystems and experiments. The following examples illustrate how these principles have been realised in ATLAS, LHCb, ALICE, and Belle II.

ATLAS: From Parallel VME to Serial ATCA and PCIe Architectures ATLAS began Run 1 with a highly partitioned architecture. Event building for the trigger relied on PCI-based Read-Out Buffer INterface (ROBIN) cards [111], while the Region-of-Interest Builder [112] implemented event routing across multiple VME boards, each hosting up to 22 FPGAs. This reflects the findings from early CMS electronics (see subsection 3.2.1). During Run 2, the PCI-based ROBIN cards were replaced with C-RORC PCIe cards [113], originally developed for ALICE, since the PCI standard had become obsolete [114]. These were later complemented by FELIX PCIe cards [31], which continue to be developed, with modernised AMD Versal-based versions foreseen for Run 4 [115].

The trigger system of ATLAS underwent a similar progression. The first Central Trigger Processor (CTP) was essentially realised as a set of VME cards interconnected through the backplane [116]. In Run 3, more specialised hardware was introduced in the form of the electron and jet Feature EXtractors (eFEX and jFEX), based on the global Feature EXtractor (gFEX) module, an ATCA blade equipped with four Virtex-7 FPGAs [117, 118, 119]. Looking toward the Phase-2 upgrade, the Global Common Module (GCM) [120] is being introduced as a central element of the ATLAS trigger. The GCM integrates two AMD Versal FPGAs with optical connectivity provided by Samtec FireFly modules, while dedicated ATCA feature extractor boards continue to perform subsystem-specific tasks.

LHCb: Commercial Alignment and Triggerless Readout In contrast to CMS and ATLAS, LHCb operated under lower luminosity and smaller data volumes, enabling an early shift toward software-based triggering. In Run 1, it employed TELL1 VME cards [25] and operated without a dedicated hardware trigger, instead filtering events in a two-stage HLT running on CPUs. This was feasible given the manageable event rates. In Run 2, LHCb introduced the TELL40 system using the ATCA standard with AMC modules, which supported optional hardware-trigger functionality [121, 45]. In Run 3, the collaboration adopted PCIe40 cards [122, 123], enabling GPU-based trigger processing and thereby taking advantage of commercial compute platforms and the ability to perform more complex and precise triggering algorithms on complete sets of data [45]. For Run 4, LHCb is deploying PCIe400 boards [124], which extend this concept by providing greater processing capacity and supporting advanced real-time selection algorithms.

ALICE: Software-Centric DAQ and Shared Hardware Development ALICE, similar to LHCb, has maintained a strong focus on software-based event processing. Since Run 1, ALICE has used PCI-based readout cards to stream data from front-end electronics to a large-scale computing farm. Its D-RORC PCI card [125] was later succeeded by the C-RORC PCIe card and, more recently, the CRU [126] for Run 3 and Run 4, which is based on the LHCb's PCIe40. These cards interface directly with the detector readout and feed data into ALICE's software-driven DAQ system.

Belle II: Cross-Experiment Technology Transfer Beyond the LHC Even outside the LHC, similar trends are visible. For example, the Belle II experiment replaced its VME-based COPPER board [127] with the PCIe40 card already employed in LHCb and ALICE [128] and the ONSEN system [129] that implements four µTCA cards on an ATCA card. This cross-implementation of hardware technologies underscores the consolidation of design practices across the high-energy physics community.

Across experiments, several consistent architectural pressures can be identified:

- **Interface consolidation:** Digitisation at the front-end and the adoption of serial links have reduced subsystem-specific back-end requirements, enabling greater reuse of processing platforms.
- **Rising integration density:** Increasing channel counts and bandwidth place growing demands on signal integrity, power delivery, and thermal management, making verification and design discipline critical.

- **Heterogeneous processing:** Trigger and event-building functions are increasingly distributed across firmware, CPUs, and accelerators such as GPUs, requiring well-defined hardware–software interfaces.
- **Timing and synchronisation:** Precision timing has emerged as a first-class system constraint, influencing clock distribution, backplane design, and board architecture.

Although implementation strategies differ, these pressures reflect a shared move toward integrated, scalable systems whose complexity is no longer dominated by raw bandwidth alone. Instead, architectural coherence, verification effort, and collaborative scalability become limiting factors for future DAQ designs.

3.2.3 Application of Trends into Future Systems

From Run 1 to the HL-LHC era, two long-term trends in DAQ design can be identified. The first is the migration from fragmented, subsystem-specific boards towards more unified back-end designs. This shift was enabled by the steady increase in FPGA resources and I/O capabilities, which made it possible to implement complete DAQ and trigger algorithms in firmware on a single device. The second is the transition from parallel-bus architectures such as VME toward serial interconnect standards including µTCA, ATCA, and PCIe. Using commercial optical modules or the PCIe form factor, the systems integrate well with the commercial High Performance Computing (HPC)-based HLT. Together, these trends reflect a move toward more scalable, integrated, and commercially aligned DAQ systems.

These qualitative developments are consistent with the quantitative trajectory shown in Figure 3.6, which compares the number of logic cells in successive FPGA generations with the readout channel count of the CMS tracker.

Figure 3.6 illustrates the growth of AMD high-end FPGAs between 2002 and 2021 [130, 131, 132, 133, 134, 135, 59, 136], shown together with the increase in readout channels of the CMS inner tracker [137, 138, 139, 140, 16]. The FPGA logic cell count is used here as a representative metric for processing capability, since it scales broadly with the complexity of algorithms that can be implemented in firmware. Likewise, the tracker channel count serves as a proxy for detector granularity and, more generally, for the scale of DAQ demands. Although each quantity captures only one aspect of system complexity, taken together they provide a useful measure of how detector requirements and processing capacity have co-evolved.

While detector granularity will continue to increase, the rate of FPGA performance scaling has been considerably faster. If this trend persists, future DAQ systems may employ fewer, more powerful processing nodes, potentially realised as single-FPGA or SoC-based back-end boards.

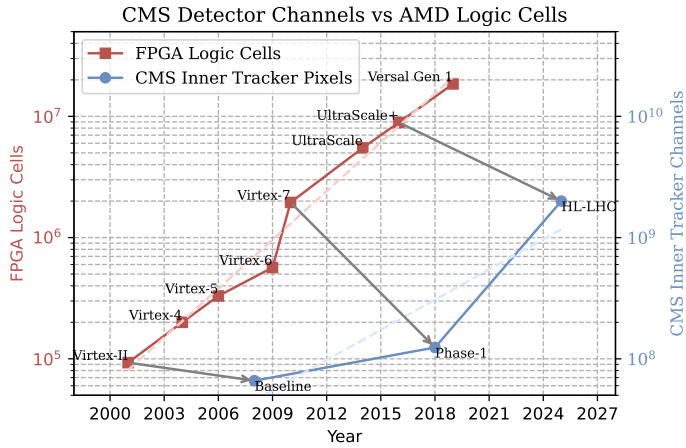


Figure 3.6: CMS inner tracker pixel count and AMD logic cell count over time. Dashed lines indicate linear trends; arrows identify FPGA families used in tracker back-end systems.

However, increases in FPGA I/O bandwidth are predominantly achieved through higher per-lane data rates rather than a proportional growth in link count [133, 134, 135, 59, 136]. Since the number of optical links is largely determined by detector segmentation and front-end architectures, a reduction in the number of back-end boards is only feasible if more detector channels can be aggregated per link. Such aggregation would require corresponding advances in front-end electronics and optical transmission technologies, for example through higher-speed serialisation or emerging silicon photonics solutions [141]. Under these conditions, future systems could reduce the number of cards required for an experiment and thereby simplify system integration.

3.3 Future Requirements Beyond HL-LHC

To derive requirements that remain valid beyond the CMS Phase-2 upgrade, it is necessary to consider the characteristics of next-generation HEP experiments currently under study. While the physics goals and accelerator technologies differ, these projects impose common pressures on DAQ systems that directly affect hardware architecture, integration, and development methodology.

3.3.1 Future High-Energy Physics Accelerators

In order to know, what features must be supported by the modular hardware toolbox that enables next-generation DAQ systems in HEP experiments, the upcoming experiments need to be analysed. Due to the necessary decades of planning to build these experiments, a lot of information is already

available. Essentially, three major alternative collider facilities are being discussed and prioritized within the next decades in HEP:

- International Linear Collider (ILC) in Japan
- Future Circular Collider (FCC) in Switzerland and France
- Circular Electron Positron Collider (CEPC) in China

The ILC is a proposed $20\,\mathrm{km}$ to $50\,\mathrm{km}$ linear accelerator designed to collide electrons and positrons at center-of-mass energies up to $500\,\mathrm{GeV}$, with potential upgrades to $1\,\mathrm{TeV}$. Developed by an international consortium, its Technical Design Report (TDR) was finalized in 2013 [142]. The ILC design includes two detectors (ILD and SiD) and places significant emphasis on cost optimization, risk mitigation, and site flexibility, with Japan (Kitakami region) considered a leading host candidate.

CERN's FCC program [143] proposes a $91\,\mathrm{GeV}$ to $365\,\mathrm{GeV}$ $\mathrm{e^+e^-}$ collider (FCC-ee) in a $91\,\mathrm{km}$ tunnel near Geneva, optimized for precision measurements of the Z, W, Higgs bosons, and the top quark. Its design prioritizes energy efficiency and sustainable construction. The FCC's modular roadmap allows seamless transition to a future $100\,\mathrm{TeV}$ hadron collider (FCC-hh). The FCC-ee plans four detectors alongside the interaction point, while there are currently three concepts. The Compact LInear Collider (CLIC)-Like Detector (CLD) is a development based on the International Large Detector (ILD) detector that has been adapted to a circular collider. It features a silicon tracker and a highly-granular calorimeter. The Innovative Detector for Electron-positron Accelerator (IDEA) is a completely new detector concept that replaces the silicon tracker with a drift wire chamber and uses a scintillation- and Cherenkov-based dual-readout calorimeter. The A Lepton-Lepton collider Experiment with Granular Read-Out (ALLEGRO) detector concept describes a noble liquid electromagnetic calorimeter cooled to cryogenic temperatures. The FCC program currently schedules first $\mathrm{e^+e^-}$ collisions for the second half of the 2040s [143].

The CEPC is China's flagship project to build a circular $\mathrm{e^+e^-}$ collider with a $100\,\mathrm{km}$ circumference, designed as a Higgs factory with $91\,\mathrm{GeV}$ to $240\,\mathrm{GeV}$ center-of-mass energy, upgradeable to proton-proton collisions in the future (SPPC). The CEPC is planning to build two detectors on opposite sides of the accelerator. The so-called baseline detector adopts the CLD concept and the other detector is likely adopting the IDEA detector concept. While the CEPC and the FCC-ee share numerous similarities, CEPC plans initial collisions as early as the mid 2030s. Independent experts are, however, predicting similar timescales to first physics for FCC-ee and CEPC [144].

All of these facilities are planned as electron–positron ($\mathrm{e^+e^-}$) colliders [145, 146, 147], while beam commissioning for the future hadron collider FCC-hh is foreseen only for the 2070s [148]. In contrast to hadron colliders such as the LHC, which collide composite particles and therefore

exhibit large total interaction cross-sections, e^+e^- colliders bring elementary particles into collision, resulting in substantially smaller interaction probabilities. Consequently, the mean number of interactions per bunch crossing at e^+e^- colliders is far below 1, and the majority of bunch crossings produce no detectable activity. The effective detector event rate is therefore governed by the physics interaction rate rather than by the bunch-crossing frequency. This fundamental difference explains the large discrepancy in maximum event rates between the CMS detector at the HL-LHC and the CEPC baseline detector shown in Table 3.4, despite comparable bunch-crossing frequencies. While HL-LHC events are dominated by high pile-up and large detector occupancies, e^+e^- collider events are sparse, isolated, and recorded at significantly lower rates.

Table 3.4: Projected running parameters of the HL-LHC CMS and CEPC baseline detectors.

	CMS (HL-LHC)	**CEPC Baseline Detector**
Event rate rate (maximum)	40.079 MHz	100 kHz
Event size	8.4 MB	20 MB
Raw data rate	$2.7\,\mathrm{Pbit\,s^{-1}}$	$16\,\mathrm{Tbit\,s^{-1}}$

Table 3.4 contrasts projected parameters for the CMS detector at the HL-LHC [43] with those of the baseline detector concept for CEPC [149]. While the expected raw data rate of the CEPC baseline detector is considerably lower than that of CMS, this comparison reflects only aggregate bandwidth and does not capture other critical constraints such as link topology, timing distribution, or system integration effort. Nevertheless, it indicates that future electron–positron collider experiments are unlikely to demand fundamentally new back-end bandwidth scales beyond those already encountered in HL-LHC developments.

This observation does not imply that DAQ development can remain static. Even when aggregate bandwidth requirements stabilise, challenges related to system integration, maintainability, and interaction with rapidly evolving computing ecosystems persist. Moreover, future readout and processing strategies are expected to exploit heterogeneous resources and increasingly sophisticated algorithms, shifting the primary development burden from raw throughput toward architectural coherence and verification effort. These trends are reflected in recent ECFA and Snowmass roadmaps, which are examined in the following subsection.

3.3.2 ECFA and Snowmass Projections

The European Committee for Future Accelerators (ECFA) is an initiative to coordinate the preparation for the technological challenges of future particle accelerators. The American initiative is

called the Snowmass Process, named after the original venue location. Both initiatives are evaluating the current state of HEP and are suggesting and evaluating future experiments, which they publish in reports. ECFA has recently released the Detector Research and Development (DRD) roadmap [141]. This roadmap is organised into dedicated work packages that address critical detector and data acquisition needs across the community. One of the pillars of this effort is DRD7, which focuses on back-end electronics, DAQ, and real-time data processing systems. The DRD7 collaboration organises its activities into seven work packages (WP7.1–WP7.7) that align with the ECFA Detector R&D roadmap on critical electronics topics [150]. Similarly, the Snowmass report is structured into frontiers, one of which is the instrumentation frontier (IF) [151, 152], which is comparable to the DRD7. Major points in the DRD7 roadmap and the IF of the Snowmass report with respect to the Trigger and DAQ systems are:

- Both committees are projecting *no-backend* or *streaming* DAQ systems as major development, as commercial servers can keep up with the data rate in near future detectors. The detector back-end will then either only be used as a smart switch or removed completely. For this to work, research is ongoing for 100 GbE links to the front-end, which can withstand the high radiation environments in the detector.
- Using Machine Learning (ML) in trigger decisions is also a topic that is pushed by both committees, as ML algorithms can easily be parallelised on commercial hardware like GPUs.
- Quantum detectors will be more widespread, which demands either the DAQ systems at room temperature to perform the ADC and DAC, or the front-end to be working in cryogenics.
- Lastly, the Snowmass report acknowledges the fact that the expertise in DAQ systems must be retained, in order to have experts when building the next generation of DAQ systems.

As the reports state, the adaptation of DAQ systems to emerging technologies is of central importance for future detector systems. In particular, the increasing use ofArtificial Intelligence (AI) and ML accelerators is expected to play a significant role in the processing chain. AI-based algorithms for detector readouts are already in use [153, 154], and it is anticipated that such techniques will become an integral part of the next generation of DAQ systems. Consequently, system designs should not circumvent this trend but should instead integrate hardware platforms, such as the AMD Versal architecture, which are specifically tailored for AI/ML applications.

Simultaneously, detector technologies themselves are undergoing rapid advances. Silicon-based detectors are increasingly capable of providing higher spatial granularity, which leads to increasing bandwidth requirements for the next generation of DAQ. While front-end links in HEP are currently still limited by the radiation-tolerant transceivers on the detector, silicon photonics is

rapidly advancing, which would enable a great increase in front-end bandwidth. As for the back-end, line rates of $224\,\mathrm{Gbit\,s^{-1}}$ with PAM-4 encoding are already available in latest hardware [155] and should therefore be considered in future systems.

Furthermore, novel concepts such as quantum sensors enable improved energy resolution. The use of quantum sensors further necessitates the relocation of ADCs and DACs back to room-temperature electronics in the back-end system, introducing new architectural considerations for the DAQ chain. With the RFSoC, AMD provides SoCs that include the ADC and DAC alongside an FPGA and a Processing System (PS), which are already applied to this exact use-case [156, 157, 158].

Future DAQ systems must therefore evolve to support:

- the embedding of AI/ML-enabled processing
- the increasing data transmission bandwidth
- the seamless integration of ADCs and DACs

The trends identified by ECFA and Snowmass point to clear architectural pressures on future back-end systems. Streaming or no-backend concepts demand robust, reproducible high-speed networking. The growing use of AI/ML in the trigger chain requires heterogeneous compute fabrics that combine programmable logic with specialised accelerators. Increased detector granularity drives bandwidth beyond current design envelopes, while quantum-sensor readout introduces mixed-signal constraints that necessitate ADC and DAC integration.

Taken together, these developments imply that next-generation back-end electronics will be substantially more complex, more heterogeneous, and more tightly constrained by bandwidth and integration density. For the Modular-Hardware Toolbox, this means the methodology must scale accordingly: it cannot merely provide templates, but must offer reusable, validated design modules that encapsulate the complexity of high-speed transceivers, heterogeneous compute blocks, and mixed-signal digitisation – thereby reducing verification effort, accelerating development, and supporting the transition toward the DAQ architectures projected by ECFA and Snowmass.

3.4 Design Process Capability

The quality and reliability of complex hardware systems depend critically on the capability of the underlying design processes rather than on individual design decisions alone. While software engineering has long employed formal process-evaluation frameworks, comparable systematic approaches for PCB-based hardware development remain limited. To address this gap, this section

introduces a capability assessment framework tailored to PCB design and verification workflows, which serves both as a benchmarking instrument and as a structured basis for methodological improvement.

3.4.1 Capability Maturity Model Integration

The CMMI [159] has become a widely adopted framework in industry for benchmarking engineering processes [160]. Its strength lies in providing a consistent language for describing process reliability and identifying incremental improvement steps. Within CMMI, a distinction is made between *capability levels*, which describe the reliability of individual processes, and *maturity levels*, which apply to organisations as a whole [161, 162]. Since the Modular-Hardware Toolbox targets PCB design workflows rather than organisational governance, only the concept of capability levels is adopted here as a guiding abstraction.

CMMI defines four Capability Levels (CLs):

- CL0 – Incomplete: Specific goals are not satisfied; generic goals are absent.
- CL1 – Performed: Work products are generated and specific goals are satisfied.
- CL2 – Managed: The process is planned and executed according to organisational policies, though variations between projects may remain.
- CL3 – Defined: Processes are standardised at the organisational level and may be tailored to individual projects as required.

Applied in the context of PCB design, capability levels provide a structured means of identifying weaknesses in current workflows and targeting them for improvement. These improvements can then be addressed directly by the modular hardware toolbox.

3.4.2 CMMI Model for PCB Design

Although the general CMMI variants provide useful guidance, they were originally developed for software engineering, service delivery, and acquisition processes, and therefore cannot be directly transferred to PCB workflows. The following three variants are established in industry:

- CMMI for Development (CMMI-DEV) [163], applied to software and system development,
- CMMI for Services (CMMI-SVC) [164], applied to service delivery and support,

- CMMI for Acquisition (CMMI-ACQ) [165], applied to supply chain and procurement processes.

The limited adoption of CMMI-like frameworks in PCB design is largely attributable to the historically smaller team sizes, shorter iteration cycles, and tool-driven workflows typical of hardware development, which have reduced perceived incentives for formal process modelling compared to large-scale software projects.

To address this gap, a dedicated capability model for PCB design is introduced in this work and summarised in Table 3.5. The model is derived by mapping the typical PCB lifecycle – including architecture definition, component library management, schematic entry, layout, verification, documentation, and risk management [166, 167] – onto relevant engineering and support process areas of CMMI-DEV. The resulting aspects therefore reflect established practice rather than ad hoc categorisation.

For practical applicability, the model evaluates each aspect across three capability levels: *ad hoc*, *structured*, and *managed*. This three-level scale collapses the finer granularity of CMMI capability levels into a form that is suitable for small and medium-sized hardware teams, while retaining sufficient resolution to identify systematic weaknesses. A similar reduction has been applied successfully in domain-specific capability models for ASIC development verification [168]. Each aspect in Table 3.5 is assessed using the three-level capability scale defined above, ranging from ad hoc practices to managed and systematically enforced workflows.

The model captures both technical practices, such as placement, routing, and verification, and organisational aspects, including documentation and risk management. By evaluating these domains across the three capability levels – ad hoc, structured, and managed – it provides a pragmatic framework that is sufficiently detailed to identify systemic deficiencies while remaining simple enough for consistent application across hardware projects.

3.4.3 Current Design Process Capability

To establish a baseline, the capability model was applied to a set of recent and ongoing PCB projects at IPE using a structured questionnaire. In all cases, the assessment was conducted in close interaction with the engineers responsible for the respective projects, allowing clarification of terminology, alignment on the interpretation of capability levels, and consistent application of the model across projects.

This guided evaluation was essential because capability assessments are inherently sensitive to subjective interpretation, particularly when distinguishing between structured and managed practices. Restricting the study to projects within a single institute ensured methodological

Table 3.5: PCB design capability model derived from CMMI-DEV and IPC-2221B

Aspect	Level 0: Ad hoc	Level 1: Structured	Level 2: Managed
Project Template	No templates; each project starts from scratch.	Reuse of partial templates; some consistency between projects.	Standardised, version-controlled templates reused across projects.
Architecture Definition	Architecture defined informally; evolves during design.	Preliminary block diagrams with informal design reviews.	Formal architecture documents with defined interfaces and review procedures.
Component Database	Inconsistent, local libraries. Multiple sources.	Shared symbol libraries.	Centralised, validated libraries with automated checks and traceability.
Schematic Entry	Manual entry. No guidelines. No reviews.	Schematic reviews performed.	Centralised, validated schematics with traceability.
Placement	Manual placement without clear rules.	Placement follows internal guidelines; peer-reviewed for critical components.	Placement optimised against defined metrics (SI, PI, routing complexity).
Layout	Routing performed ad hoc with inconsistent rules.	Basic DRC/DFM checks applied; manual corrections.	Systematic rule compliance with SI/PI simulation.
Design Review	None.	Ad hoc reviews.	Structured peer reviews with checklists and follow-ups.
Design Verification	Manual lab test after assembly.	Lab tests following a plan, some simulation used ad hoc.	Automated lab tests and simulation-based verification linked to requirements.
Software	No integration considered during hardware design.	Hardware based on known processing modules, enabling partial software reuse.	Defined HW/SW interfaces; co-designed with software frameworks.
Documentation	Minimal notes scattered across files.	Central repository with inconsistent structure; reviewed	Standardised templates; version-controlled, reviewed, systematically updated.
Risk Management	No tracking.	Known risks discussed informally.	Risk log maintained with mitigation strategies.

consistency and comparability of results. The selected projects therefore constitute a controlled case study that is sufficient for identifying recurring process-level weaknesses and for evaluating the potential impact of the Modular-Hardware Toolbox. The analysis does not aim to provide a statistically representative survey of the wider CERN electronics community, but rather to establish a reliable reference baseline against which methodological improvements can be assessed.

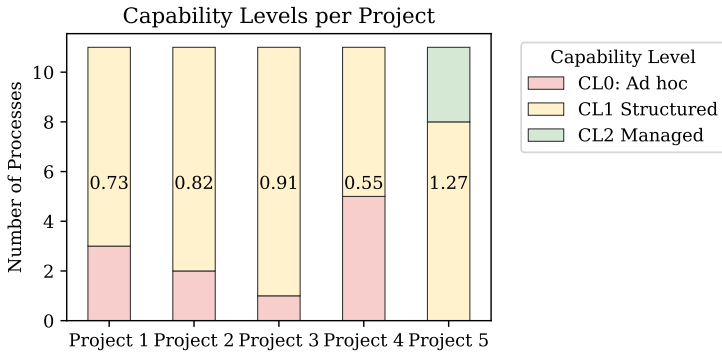


Figure 3.7: Project capability levels at IPE, with bars annotated by average capability level.

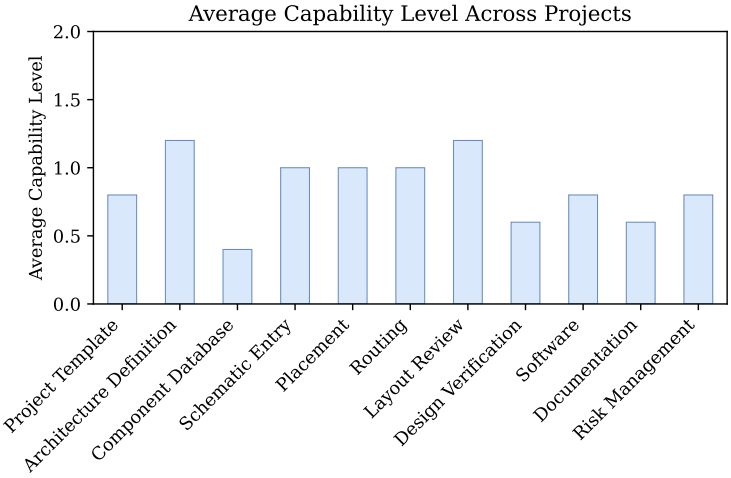


Figure 3.8: Average capability levels at IPE derived from the PCB capability model. The vertical axis denotes the mean capability level (0 = ad hoc, 1 = structured, 2 = managed) averaged across evaluated projects for each aspect.

Five projects (ECHo [53], Tandem-L, PANDA-AMC [169], TRISTAN Tile-Main-Board, ZCU216 DirectRF [157]) that do not yet use the modular hardware toolbox were evaluated according to the capability model in Table 3.5. The results in Figure 3.7 show that only one project demonstrates

predominantly structured processes, with some processes reaching the managed level. By contrast, most projects are situated between CL0 and CL1, indicating that many processes are either ad hoc or only partially structured.

The aggregated capability results in Figure 3.8 show consistent shortcomings in component database management, design verification, and documentation workflows – domains in which projects repeatedly rebuild similar structures with varying quality, limited traceability, and little reuse. These weaknesses stem not from isolated design errors but from the absence of a shared technical foundation that standardises how schematics, footprints, verification artefacts, and documentation are created and maintained. Incremental fixes at the project level cannot resolve this structural fragmentation.

The Modular-Hardware Toolbox should directly address this root cause by providing a clearly defined set of reusable, version-controlled design assets – that is, templates, configuration sets, and automation scripts that form the technical baseline of every hardware project. Concretely, these assets must include:

- **Component assets:** a central, validated component database comprising symbols, footprints, 3D models, parametric metadata, lifecycle status, and automated compliance checks for symbol–footprint consistency.
- **Project-structure assets:** standardised project templates with predefined directory structures, layer stacks, design-rule configurations, manufacturing-output generators, and mandatory locations for schematics, layout, documentation, and release artefacts.
- **Verification assets:** reusable test plans, simulation harnesses, Signal Integrity (SI)/Power Integrity (PI) guideline sets, automated rule-check scripts, and board-level review checklists that integrate verification tasks into the development workflow.
- **Documentation assets:** README and release-note templates, changelog formats, structured documentation trees (e.g. Markdown), and CI scripts that automatically generate documentation bundles at each release.

These assets provide more than convenience. By inheriting the same validated components, design rules, verification steps, and documentation structures from the outset, projects no longer depend on ad hoc setups or designer-specific conventions.

3.5 Electronic Design Automation and Hardware Reuse

The capability analysis in section 3.4 identifies recurring deficiencies in PCB design workflows, including limited reuse, weak traceability, and a high reliance on manual intervention. While these shortcomings are often discussed in organisational terms, they are also shaped by the capabilities and limitations of the electronic design automation tools used to implement the design process. In practice, many aspects of process capability – such as modular reuse, controlled variation, interface validation, and automation – can only be realised if they are explicitly supported by the underlying EDA environment. This section therefore examines contemporary EDA tools from the perspective of how effectively they enable the process capabilities required for scalable, reusable hardware development.

Unlike software, which mostly used text-based programming languages and adopted modular methodologies decades ago, conventional PCB tools (e.g. Altium Designer [170], Cadence Allegro [171], Siemens EDA PADS/Xpedition [172, 173], KiCad [174]) remain predominantly GUI-driven. Recently, however, PCB-level Hardware Description Languages (HDLs) such as JITX [175, 176], Polymorphic Blocks [177], PHDL [178] and atopile [179] have emerged, signalling a rising interest into software-like workflows for PCB-design. To assess how well these approaches support modular design, it is first necessary to establish the key requirements.

3.5.1 Essential EDA Capabilities for Reuse

The capability assessment in Figure 3.8 highlights recurring deficiencies that extend beyond individual weak domains. Many PCB design processes lack standardisation, systematic reuse, and controlled variation – issues that are well understood in systems engineering and component-based design. The following capabilities therefore serve as benchmarks for a scalable and maintainable modular PCB design methodology. They reflect established principles of component reuse, interface control, configuration management, and abstraction, as introduced in subsection 2.3.3.

In contrast to software engineering, which relies on text-based programming languages and has adopted modular development methodologies for decades, conventional PCB design tools (e.g. Altium Designer [170], Cadence Allegro [171], Siemens EDA PADS/Xpedition [172, 173], KiCad [174]) remain predominantly GUI-driven. Logical structure and physical realisation are often tightly coupled, which complicates reuse, controlled variation, and automation across projects.

A useful point of comparison is *digital* Very-Large-Scale Integration (VLSI) design, where strict separation between behavioural description, synthesis, and physical implementation enables

systematic reuse, parameterisation, and scalable automation. PCB-based back-end electronics in HEP experiments have evolved from mixed-signal designs toward predominantly digital systems, suggesting that selected concepts from digital hardware description and workflow separation may be transferable to PCB development at the architectural level.

In this context, PCB-level HDLs such as JITX [175, 176], Polymorphic Blocks [177], PHDL [178], and atopile [179] have emerged, reflecting growing interest in software-like design workflows for electronics. While these approaches differ in maturity and scope, they illustrate potential mechanisms for enabling modularity, parameterisation, and automation in PCB design. To evaluate such tools meaningfully, the underlying capabilities required for modular reuse must first be defined.

Essential capabilities

1. **Reusable schematic and layout blocks:** Subcircuits should be reusable at both schematic and layout level, reducing duplicated effort and limiting the number of independent implementations that must each be verified. Hierarchical PCB design practice routinely demonstrates that reuse improves consistency and accelerates development.

2. **Parametrised modules:** Modules must support controlled variation without duplication. For example, a DC–DC converter module may be instantiated with different passive values to generate $5\,\mathrm{V}$ or $12\,\mathrm{V}$. Parameterisation prevents divergence between variants and ensures that updates and bug fixes propagate consistently.

3. **Versioned module libraries:** Configuration management is a core principle of systems engineering (ISO/IEC/IEEE 15288). Without explicit versioning, reused modules quickly diverge across projects, making it impossible to track which implementation is manufactured, tested, or deployed. Versioned libraries therefore turn informal reuse into a controlled, auditable process.

Advanced capabilities

4. **Interface consistency checks:** Automated validation of pin functions and signal types reduces integration errors at module boundaries. Basic checks prevent electrically incompatible connections (e.g. two driven outputs), while more advanced checks verify bus semantics (e.g. avoiding `SCL`/`SDA` swaps in I^2C). Such interface-control mechanisms are standard in component-based design and crucial for safe module integration.

5. **Automation and scripting support:** APIs and scripting interfaces enable scalable instantiation of modules, generation of design variants, and integration with continuous-integration

workflows. This capability becomes essential once the number of projects, variants, or release cycles increases, as automation is the only reliable way to keep module libraries and verification steps applied consistently.

3.5.2 EDA Software Analysis

While some of the above mentioned capabilities are standard in enterprise environments, others (notably parametrisation, version control, and automation) remain unevenly supported. In the following comparison, feature support refers to native, production-ready functionality intended for systematic reuse across projects, rather than ad hoc workarounds or third-party extensions.

Table 3.6: Modular design feature support in selected EDA tools.

Feature	Altium	Allegro	PADS	Xpedition	KiCad
Schematic and Layout Reuse	Yes	Yes	No	Yes	No
Parametrisation	Limited	Partial	No	Yes	Yes
Versioned Modules	Yes	Yes	Partial	Yes	Partial
Interface Checks	Limited	Yes	Manual	Yes	Limited
Scripting/API	Yes	Yes	Yes	Yes	Yes

A comparison of conventional EDA tools is shown in Table 3.6. Altium Designer is widely used in small to medium enterprises due to its intuitive interface and relatively moderate licensing costs. Its main strength in the context of modularity lies in “Design Reuse Blocks,” which allow version-controlled reuse of schematic and layout sections across projects. These blocks are managed via Altium 365, which integrates reuse seamlessly but introduces dependency on a cloud service. Parametrisation, however, remains limited and largely manual.

Cadence Allegro and Siemens Xpedition dominate the high-end market, particularly in large teams and regulated industries. Both offer robust constraint-driven design and strong collaboration features, including real-time co-editing. Their reuse methodology is less integrated than Altium’s, relying on independent schematic and placement files. This, however, simplifies integration with Git repositories, providing natural version control.

Mentor PADS, now part of Siemens EDA, occupies a middle ground between Altium and enterprise tools. It supports hierarchy and team workflows but lacks polished layout reuse and real-time collaboration, making it less suited for highly modular workflows.

KiCad, as a free and open-source alternative, offers broad accessibility. It supports hierarchical schematics and parametrisation but lacks integrated layout reuse. Community plug-ins offer

partial solutions, and upcoming versions aim to address these limitations. Git-based versioning is straightforward due to KiCad's text-based file formats, and a Python interface enables scripting. However, collaboration is limited to Git merges (albeit supported by third-party merge tools).

Overall, Allegro and Xpedition provide the most complete modularity support, but their cost and complexity restrict accessibility. Altium and PADS occupy the middle ground, while KiCad remains the most accessible but least feature-rich option.

As electronic systems grow in complexity, text-based PCB-HDLs are emerging to bring modularity, reuse, and automation directly into the design flow [175, 177]. These approaches bypass manual schematic entry, generating design files programmatically for use in conventional EDA environments. A comparison of JITX and Polymorphic Blocks, the two most capable solutions is given in Table 3.7.

Table 3.7: Comparison of modular design features in PCB-level hardware description languages.

Feature	**JITX**	**Polymorphic Blocks**
Schematic and Layout Generation	Yes	Schematic only
Parametrisation	Yes	Yes
Versioned Modules	Yes	Partial
Interface Checks	Yes	Type-checked
Scripting and API Access	Yes	Python-native

JITX is a commercial solution that adopts a code-first design paradigm with a dedicated domain-specific language. It integrates schematic capture, layout generation, and constraint enforcement, supporting parametrisable blocks and supply-chain-aware optimisation. Designs can be exported to Altium Designer or KiCad for editing, but the tool requires a significant shift in workflow, which may hinder adoption in institutes with established GUI-driven practices.

Polymorphic Blocks, by contrast, is an open-source project hosted in Python. It provides type-safe abstractions and parametrisable circuit blocks that compile into graph-based circuit representations. These can be exported as schematics for Altium and KiCad, although no layout engine is included. By operating exclusively at the schematic level, Polymorphic Blocks may offer a pragmatic intermediate step toward modularisation without requiring immediate adoption of fully code-generated layouts. Otherwise, being a HDL, it has excellent reuse capabilities, while posing a completely new workflow on the engineers.

Overall, current EDA tools provide only partial support for modular PCB design. Enterprise solutions offer extensive functionality but remain inaccessible to many academic teams, while mid-range and open-source tools trade modularity for usability and cost. Emerging PCB-HDLs

promise software-like reuse and automation but require significant workflow changes and are still maturing. For the Modular-Hardware Toolbox, this landscape motivates a tool-agnostic approach that defines reusable assets at an abstraction level compatible with existing GUI-driven tools, while remaining adaptable to future code-centric design methodologies.

3.6 Summary of Requirements

The analyses presented in this chapter converge onto five underlying needs that guide the design of the Modular-Hardware Toolbox. Each requirement is grounded in the empirical evidence and organisational observations developed throughout this chapter.

Consistency (R1) The survey of Phase-2 back-end boards in section 3.1 and the comparative analysis in subsection 3.1.2 reveal very different approaches across otherwise similar designs. This is further reinforced by the capability assessment in section 3.4, which identifies heterogeneous symbol and footprint libraries, informal schematic structures, and non-standardised release artefacts as recurring weaknesses. These findings establish consistency as a fundamental requirement for a methodology intended to reduce duplicated effort and improve reproducibility.

Reuse and Modularity (R2) Two strands of the requirement analysis motivate modularity. First, the architectural comparison in section 3.1 shows that independently developed ATCA boards follow a nearly identical functional structure, suggesting that reusable schematic and layout blocks are both feasible and advantageous. Second, the collaboration study in subsection 3.1.3 demonstrates that modular partitioning is essential for distributed development across institutes, enabling parallel work while maintaining well-defined interfaces. Together, these observations support reuse and modularity as a central requirement for scalable hardware development.

Automation and Error Reduction (R3) The capability evaluation in section 3.4 highlights the prevalence of manual, error-prone tasks across multiple domains, including footprint creation, design-rule enforcement, documentation, and verification. Low capability levels in these areas indicate structural rather than incidental weaknesses. Automation therefore emerges as a key requirement for reducing human error, increasing reliability, and ensuring that essential steps are applied consistently across projects. It further accelerates the development of complex boards as predicted in subsection 3.3.2.

Traceability (R4) Throughout the requirement analysis, fragmented information and missing configuration control appear as systemic issues. The discussions in section 3.4 and subsection 3.4.2 identify inconsistent component libraries, ad hoc documentation structures, and weak linkage between verification artefacts and design intent as persistent shortcomings. These problems justify the need for centralised component metadata, structured documentation, and version-controlled artefacts, motivating traceability as a core requirement.

Scalability and Collaboration (R5) The organisational analysis in subsection 3.1.3 shows that collaboration is constrained not by technical divergence but by funding structures, institute-specific deliverables, and limitations in concurrent design tooling. At the same time, the technological evolution outlined in section 3.1 and subsection 3.1.2 – including rising system complexity, heterogeneous compute architectures, and increasing integration density – demands more coordinated workflows. These combined pressures motivate scalability and collaboration as a structural requirement for future hardware development methodologies.

4 Modular-Hardware Toolbox

Chapter 3 established a set of functional and organisational requirements for modern PCB and hardware design workflows. These requirements arose from the increasing complexity of high-energy physics electronics, the need for reproducible design processes, and the desire to reduce duplicated effort across multiple projects. In particular, the analysis identified the following challenges as central: inconsistent project structure, heterogeneous symbol and footprint libraries, repeated manual creation of design files, insufficient traceability, and the lack of systematic reuse of validated design artefacts. The *Modular-Hardware Toolbox* presented in this chapter is a direct response to these challenges. It provides a structured collection of design artefacts that together form a coherent environment for hardware development.

4.1 Architecture of the Modular-Hardware Toolbox

The Modular-Hardware Toolbox formalises a structured environment for designing, verifying, and maintaining electronic hardware. Its architecture is grounded in the requirements derived in Chapter 3, which identified the need for consistent project foundations, reusable building blocks, automated generation of artefacts, unified component metadata, and systematic verification processes. To meet these requirements, the toolbox is organised into six subsystems that together provide a coherent methodological and technical framework.

4.1.1 Toolbox Scope and Design Rationale

The toolbox separates concerns along the full hardware development workflow. Project-level infrastructure defines how new designs are initiated and documented, supporting consistency (R1) and collaboration (R5). Reusable design modules encapsulate functional behaviour at schematic and layout level, enabling modularity and reuse (R1, R2). Framework templates extend this concept to larger subsystems. Automated generators translate structured data into fabrication artefacts and documentation, improving reproducibility and reducing human error (R3). A consistent component database ensures traceability (R4) and consistency across projects,

even within collaborations (R5), while verification and release infrastructure ensures that designs meet predefined quality and manufacturability criteria (R1, R3, R4).

The resulting structure is intentionally tool-agnostic: it specifies responsibilities, artefact boundaries, and validation steps rather than relying on proprietary Electronic Design Automation (EDA) features. Three operational principles constrain the toolbox design. First, *modularity* requires reusable building blocks with explicit interfaces and integration constraints. Second, *standardisation* constrains variability through templates, naming conventions, and shared libraries to keep designs comparable and reviewable across projects. Third, *automation* makes release artefacts reproducible by generating them deterministically from versioned sources rather than manual tool state.

Modularity Modules are the fundamental reusable entities within the toolbox. They provide the primary abstraction for encapsulating functional circuitry at schematic and layout level. For subsystem-level abstractions that exceed the scope of a single module – such as high-pin-count devices or standard-defined mechanical contexts – the toolbox introduces framework templates, described in Section 4.2.3.

Standardisation Standardisation reduces variability and strengthens reproducibility by enforcing common project templates, naming conventions, metadata structures, validated layer stacks, and design-rule packs. These measures support consistency (R1) and scalable collaboration (R5) by ensuring that design artefacts remain compatible across projects and teams. Standardisation also extends to visual conventions, abstraction mechanisms (such as harness definitions), documentation structures, and review procedures.

Automation Automation reduces human error and ensures deterministic artefact generation. The toolbox employs structured data sources to generate fabrication files, bills of materials, and release bundles. This approach improves reliability and supports systematic error reduction (R3). By embedding process logic into scripted generators, the toolbox enforces consistent release procedures across projects and supports repeatable design flows independent of individual users or tools.

4.1.2 Subsystems Decomposition

The Modular-Hardware Toolbox is decomposed into the following six subsystems, each fulfilling a distinct architectural responsibility. A high-level overview of this decomposition is shown in Figure 4.1.

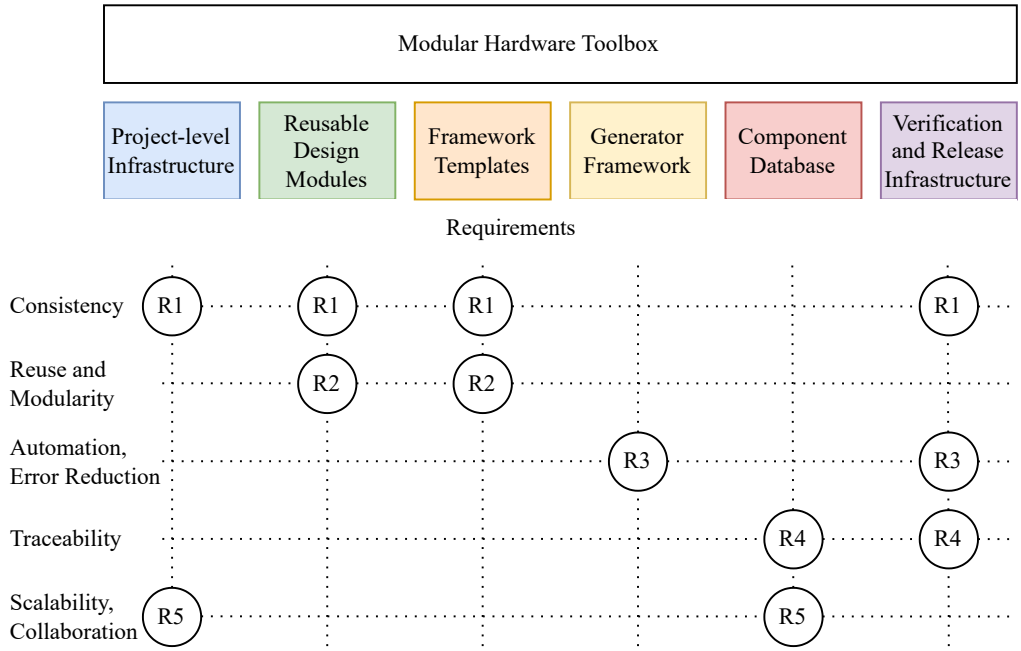


Figure 4.1: Architecture of the Modular-Hardware Toolbox. The upper section lists the artefacts contributed by each subsystem; the lower section maps subsystems to the design requirements they address.

1. **Project-Level Infrastructure:** Defines templates, validated layer stacks, design-rule packs, documentation conventions, and version-control structure.
2. **Reusable Design Modules:** Provides functional building blocks at schematic and layout level, including power, clocking, conversion, processing, and interface modules.
3. **Framework Templates:** Capture subsystem-level scaffolding for complex or standard-defined designs, such as FPGAs or ATCA-compliant boards.
4. **Generator Framework:** Automates the creation of fabrication artefacts and documentation from machine-readable inputs.
5. **Component Database:** Serves as the single source of truth for component metadata, enabling traceability and multi-project consistency.
6. **Verification and Release Infrastructure:** Provides rule checks, review workflows, continuous integration pipelines, and deterministic release bundles.

The relationship between the requirements from Chapter 3 and the toolbox subsystems is summarised in Figure 4.1. Each subsystem directly addresses a specific subset of these requirements, ensuring traceability from analysis to architecture. Framework templates were introduced as

an additional subsystem to bridge reusable modules and system-level infrastructure, providing validated scaffolding for complex or standard-constrained designs.

4.2 Subsystems of the Modular-Hardware Toolbox

The architectural principles introduced in Section 4.1 define how the Modular-Hardware Toolbox is structured. Each subsystem implements one aspect of the architecture – from project foundations to reusable modules, framework templates, automation, metadata management, and verification. Together, these subsystems establish a comprehensive environment for modular, repeatable, and collaborative hardware development.

4.2.1 Project-Level Infrastructure

Project-level infrastructure provides the foundation upon which all designs created with the Modular-Hardware Toolbox are built. It ensures that new projects start from a validated, consistent baseline and that designers across institutes and projects operate within a uniform environment.

Standardised Project Templates Each new design begins from a standardised project template that defines the essential artefacts required for development. The template includes a predefined top-level schematic sheet, an associated PCB document, an organised directory structure for source files, metadata, and documentation, as well as placeholders for assembly drawings and release notes. The template also integrates version control conventions suited for Git, ensuring that design history, reviews, and change tracking remain reproducible and transparent. By enforcing a consistent starting point, the project template minimises ambiguity and significantly reduces onboarding effort for new collaborators.

Validated Layer Stacks and Design Rules To ensure manufacturability and electrical robustness, the toolbox provides validated layer stack definitions and design-rule packs. These rule sets cover standard four- to eight-layer boards and include controlled-impedance configurations proven in previous institutional projects. For high-performance systems, proven stacks from complex prior designs can be selected to serve as a reliable starting point, reducing risk and shortening the path to first prototypes. Designers further benefit from established baseline constraints – such as minimum trace widths, spacing, via dimensions, and differential-pair rules – without needing to perform repeated low-level validation.

Centralised Component Library Integration Project-level infrastructure is tightly coupled to the component library introduced in Section 4.2.5. The project template points to a version-controlled repository of schematic symbols, footprints, simulation models, and parametric data. By relying on a single source of truth, the toolbox avoids divergence between project-specific libraries.

Documentation and Review Infrastructure Documentation structures, schematic sheet conventions, style checks, and assembly drawing templates are embedded into each template. Together, these elements establish a reproducible, validated starting point for hardware designs.

4.2.2 Reusable Design Modules

Reusable design modules form the functional core of the Modular-Hardware Toolbox. They encapsulate validated electronic functionality at both schematic and layout level and provide well-defined interfaces that allow them to be integrated into larger system designs. This subsystem addresses requirements R1 (consistency) and R2 (reuse and modularity) and – when combined with automation and verification – supports scalable development across multiple projects and institutes.

Definition and Structure of Modules A reusable module integrates three elements into a coherent artefact:

- a **schematic implementation** capturing logical connectivity and design intent,
- a **layout implementation** capturing the physical realisation under standardised constraints,
- an explicit **interface and constraint contract** defining ports, placement rules, and integration assumptions.

Scope and Role Within the Toolbox Modules represent reusable building blocks such as power converters, PLLs, ADC/DAC stages, or standard FPGA integration patterns. Their purpose is to encode widely used circuit functions in a generalised, validated form.

To support functions whose scope exceed the boundaries of a single module, the toolbox introduces a complementary subsystem abstraction: *framework templates*, which provide validated architectural scaffolding for complex or standard-defined subsystems. These are described in the next subsection.

4.2.3 Framework Templates

Framework templates extend the concept of reusable modules by capturing subsystems whose functional complexity or interconnect density cannot be represented as a single self-contained module. They address design contexts where current EDA tools lack support for deeply hierarchical module composition and where subsystem-wide correctness requires a pre-validated architectural scaffold.

Framework templates cover two complementary classes of design context:

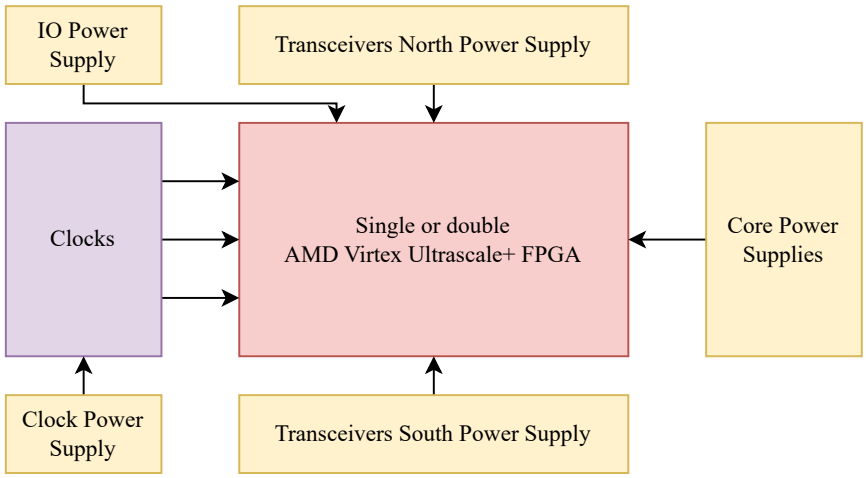


Figure 4.2: Device-centric framework template project for an AMD VU13P FPGA, derived from Serenity-S1.

Device-Centric Frameworks High-pin-count devices such as FPGAs require multiple power rails, configuration interfaces, clock sources, and auxiliary circuits. Representing such devices through a single module interface would lead to excessively large, but sparsely used port sets. A framework instead instantiates the complete subsystem – mechanics, management interfaces, and support circuitry – directly within the template. Designers then connect only the signals relevant to their system.

Standards-Constrained Frameworks Framework templates encode the mechanical and electrical requirements of standards such as ATCA, µTCA, and PCIe. These include connector locations, keep-out zones, mandatory management circuits (e.g. IPMC), and form-factor constraints. Embedding these elements ensures that every design begins from a compliant baseline.

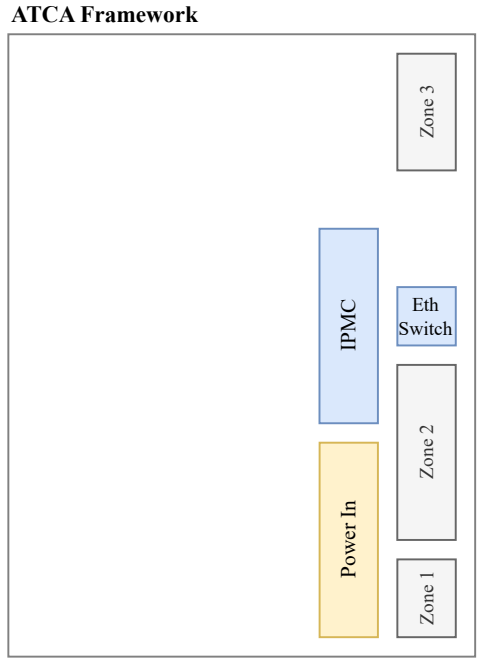


Figure 4.3: Standards-constrained framework template project for the ATCA standard, derived from Serenity-S1.

As part of this work, two framework templates were derived from the Serenity-S1 development: an ATCA framework capturing the standard's mechanical and management requirements (Figure 4.3), and a Virtex UltraScale+ VU13P FPGA framework (Figure 4.2). These demonstrate how frameworks complement modules in practice.

4.2.4 Generator Framework

The generator framework automates the creation of artefacts that would otherwise require extensive manual effort and are prone to inconsistencies. It addresses automation and error reduction requirements (R3) and forms an essential component of the toolbox by enforcing deterministic output and embedding process knowledge into the design flow.

Footprint Generators Footprint creation is automated using IPC-7351B-compliant generators, avoiding manual interpretation of data sheets and ensuring consistency across designs.

Fabrication and Documentation Generators Generators produce:

- bills of materials (BOMs),
- pick-and-place files,
- assembly drawings and stack-up documentation,
- Gerber or ODB++ outputs,
- schematic PDFs and design reports.

These adhere to naming conventions, folder structures, and revision rules defined by the project template.

4.2.5 Component Database

The component database serves as the single source of truth for component metadata and establishes the foundation for consistent, traceable design across projects. Symbols, footprints, and parametric information are centrally maintained, avoiding divergence between project-specific libraries and enabling consistent module development across institutes.

A centralised database streamlines collaborative development by ensuring that all contributors rely on the same component definitions. It enables distributed teams to develop and validate modules independently while maintaining coherence across the system-level design. This is particularly significant in large-scale scientific and industrial collaborations.

As part of this work, a unified component library was established to support the participating institutes within the Helmholtz community and associated research partners. The library provides a shared, version-controlled foundation for multi-institute collaboration and serves as a practical implementation of the toolbox principles. To remain compatible with the two most commonly used EDA environments in these institutes, all component data are stored in a tool-agnostic database structure from which both Altium Designer and KiCad libraries can be generated. The database leverages Altium's native symbol and footprint formats – which KiCad supports from version 9 onward – thereby enabling a single source of truth for symbols, footprints, and parametric metadata across both toolchains.

Each component entry is uniquely defined by its manufacturer part number, ensuring unambiguous referencing across modules and projects. The database is maintained in a Git repository, which provides transparent versioning, review capabilities, and traceability of all changes. Planned extensions include automated compliance tests for symbols and footprints, drawing on existing

open-source validation tools, to further strengthen consistency and quality across the shared library.

4.2.6 Verification and Release Infrastructure

Verification and release infrastructure provides the quality-assurance backbone of the toolbox.

Verification Infrastructure ERC and DRC rule sets embedded in the project template ensure correct connectivity, valid spacing and clearance, and adherence to differential-pair and component-placement rules. This supports consistency (R1) by ensuring that all designs are checked against the same rule set.

Structured review checklists support traceability (R4) by ensuring that design intent, parameter choices, and integration constraints are explicitly validated. These checklists also promote consistent design practices (R1) and reduce the risk of undocumented dependencies.

Continuous integration pipelines, where supported, automate style checks, ERC/DRC runs, generator invocations, and library consistency checks adapted from KiCad's library check [180]. This early-stage automated validation reduces integration risk (R3) and ensures that updates to libraries or modules do not silently violate established design conventions.

Release Infrastructure Release processes generate deterministic artefact bundles including fabrication data, assembly drawings, Bill of Materialss (BOMs), Pick-and-Place files, and design documentation. These support traceability (R4) by providing complete, versioned release snapshots and contribute to consistency (R1) across manufacturing hand-offs.

4.3 Detailed Design of Reusable Hardware Modules

While Section 4.2.2 defined the role and conceptual structure of reusable modules, this section provides a detailed methodology for constructing such modules in practice. It describes how schematic and layout implementations can be designed to maximise portability, clarity, and reusability. The principles presented here were refined through extensive prototyping and applied during the development of the Serenity-S1 card presented in the next chapter.

4.3.1 Selecting Suitable Candidates for Reusable Modules

Before describing the schematic and layout implementation of reusable modules, it is necessary to determine which circuit functions merit modularisation in the first place. Only functions whose reuse provides measurable benefit across multiple designs should be encapsulated as modules. The following criteria establish when modularisation is justified. A function qualifies as a suitable module candidate if it meets one or more of the criteria below; the categories are independent rather than cumulative.

1. Functions That Recur Across Projects and Require Significant Design Effort A circuit is a strong candidate for modularisation when it appears repeatedly in different designs and requires substantial effort to implement or validate. This includes functions that involve non-trivial data sheet interpretation, tight interaction with other subsystems (e.g. power stages, clocking paths, high-speed interfaces), or verification steps that extend beyond routine checks. Encapsulating such functions avoids repeated rediscovery of constraints, reduces verification workload, and lowers the probability of latent design errors.

2. Functions With Stable Topology and Parametrisable Operating Points Circuits whose internal structure remains stable across a broad range of operating conditions are well suited to modularisation. In these cases, reuse requires only predictable parameter adjustments – typically resistive dividers, filtering networks, or configuration pins – while the underlying topology remains unchanged. Provided that the permissible parameter space is validated for safe operation, a single module can serve multiple designs without modification. Such modules provide general applicability while retaining controlled flexibility.

3. Functions With Non-Obvious Internal Constraints Some circuits depend on auxiliary requirements that are not obvious from their primary function and are easily missed in ad hoc implementations. Typical examples include mandatory pull-ups on open-drain or status pins, strap resistors defining operating modes, enable and sequencing dependencies, or stability networks whose correctness depends on component type and placement. Such omissions may not be detected by standard ERC/DRC but can render a circuit partially or entirely non-functional, as observed for missing pull-ups on power-supply status outputs during the development of the Serenity-S1. Encapsulating these functions as reusable modules makes the hidden constraints explicit and consistently enforced, reducing integration errors and cognitive load at system design.

4. Functions That Provide Limited Abstraction Value Should Not Be Modularised Not every circuit benefits from module status. Functions that are too simple (e.g. connectors with

basic passives), highly specialised for a single design, or subject to widely varying topologies across applications are best implemented directly at project level. Avoiding these cases prevents unnecessary library growth and maintains a clear boundary between reusable building blocks and project-specific circuitry.

4.3.2 Schematic Implementation

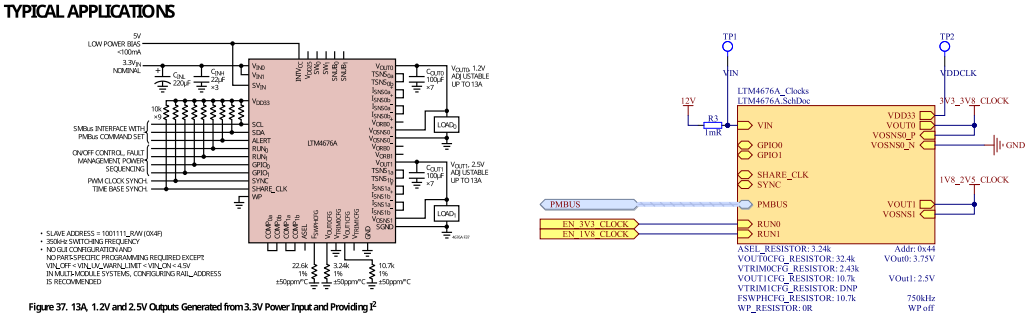


Figure 4.4: Example transformation from a manufacturer reference schematic [181] to a reusable schematic module in Altium Designer, illustrated for the LTM4676A.

Schematic design within the Modular-Hardware Toolbox emphasises clarity, abstraction, and long-term maintainability. A set of structured design principles guides module construction, adapted from established software design practices (see subsection 2.3.3). To make these principles tangible, the LTM4676A switched-mode power supply is used throughout this section as a representative example of how schematic modules are constructed within the toolbox. Figure 4.4 shows the typical application diagram from the data sheet and the finished module of the LTM4676A.

In the following, the principles are explained in detail and Figure 4.5 illustrates the concepts using the example based on the LTM4676A.

Single Responsibility Each module should implement a single primary function. This reduces coupling between unrelated subsystems and ensures that modules remain comprehensible and easy to integrate. Examples include a voltage-regulation stage, a jitter-cleaning block, or a single ADC–front-end chain. In practice, the LTM4676A module used in Serenity-S1 implements only the power module, resistors to parametrise the module and to allow operation and capacitors at the power inputs and outputs; downstream rails and consumer circuits are intentionally kept outside the module to preserve a clear responsibility boundary.

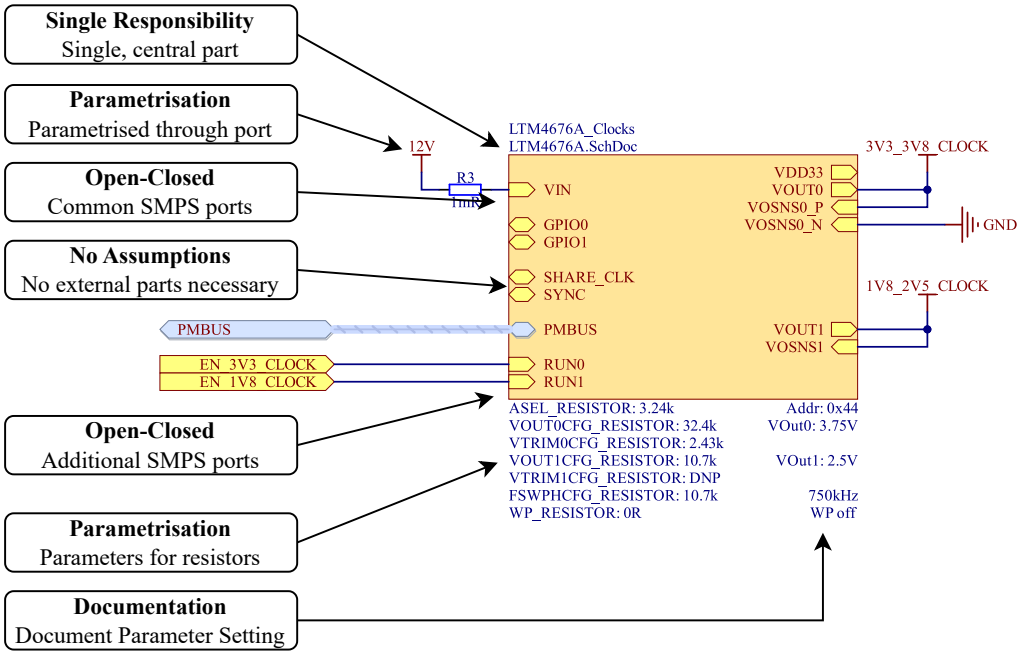


Figure 4.5: Mapping of the SOLID design principles onto schematic-level implementation using an LTM4676A switched-mode power supply as an example.

Open–Closed Structure Modules must permit extension without modification. This is achieved by defining minimal functional ports and offering structured variants for optional features (e.g. additional monitoring interfaces). Such variants must be documented but not forced onto all users. This preserves generality while supporting system-specific extensions. For example, the same power module may be used with or without PMBus telemetry: the base interface exposes only enable, power-good, and rail outputs, while a documented variant adds the PMBus signals for systems that require remote supervision.

Interface Segregation Interfaces must expose only the functionality strictly required for integration. Optional or application-specific signals are excluded unless they are common across many systems. This ensures that modules are not artificially tied to single projects. In the Modular-Hardware Toolbox, debug-only signals and board-specific configuration pins are therefore not part of the standard module interfaces but are handled at the system-integration level.

Dependency Abstraction Serial interfaces such as PMBus, SPI, and I^2C are represented through generic harness definitions included in the project template. These harnesses abstract implementation details and maintain consistent naming and signal grouping across modules. As a

result, modules that implement, for example, I^2C-controlled clock chips or SPI-controlled ADCs can be swapped or combined without redefining their control interfaces.

Parametrisation Modules should support configuration through parametrisation, primarily via resistive networks. Parametrisation enables reuse across a broad set of operating points, but the entire configuration space must remain safe: component ratings, stability margins, and filtering requirements must be valid for all supported parameter values. In the LTM4676A module, feedback and sense resistors are explicitly marked as parameters, and their permissible value ranges are documented on the schematic sheet. This allows the same module to supply different output voltages without altering its internal structure.

Documentation Practices Module schematics include embedded design notes, allowable parameter ranges, integration assumptions, and references to associated layout elements. Colours and sheet layouts follow project-wide conventions to ensure readability [182, 183]. DIN A3 format offers sufficient space for clear module representation [184].

Abstraction and Hierarchy Hierarchical schematics are used sparingly: modules serve as the primary level of abstraction, while system-level schematics arrange modules either (a) in a physical layout orientation or (b) according to functional dataflow. Both views provide complementary insights and aid cross-team communication.

4.3.3 Layout Implementation

Layout reuse requires careful abstraction of physical constraints to ensure portability across fabrication processes and target systems. The toolbox therefore defines layout practices that prioritise manufacturability and reusability over project-specific optimisation. These practices were also applied to the LTM4676A and can be observed in Figure 4.6.

Layer Usage Routing inside modules is restricted primarily to top and bottom layers. Inner-layer routing is avoided during module creation because layer counts and stack-up structures vary significantly across projects. When additional routing is required, the module connections are terminated with vias that system integrators may extend on inner layers during board-level routing.

Placement Constraints Placement respects widely used backplane standards such as ATCA [70], µTCA [185], and PCIe [72], where bottom-side component height limits – typically $2.7\,\mathrm{mm}$ to

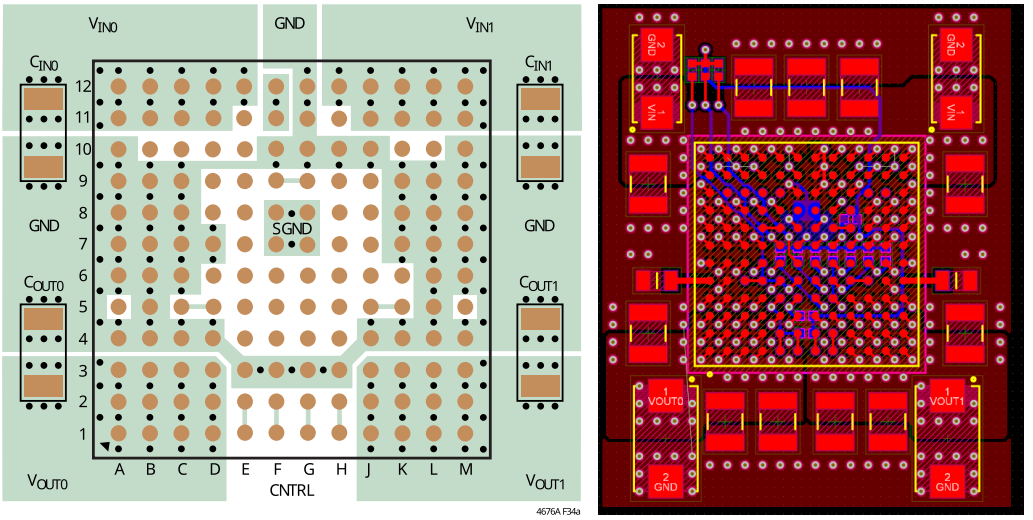


Figure 4.6: Example reference layout from the LTM4676A data sheet [181] and the corresponding reusable layout module derived within the toolbox.

3.4 mm depending on the framework – are enforced. Parametrised resistors and adjustment networks are preferentially placed on the top side to ease access during bring-up and debugging.

Fabrication Rules Modules follow relaxed, broadly manufacturable design rules [186]. HDI constraints, blind/buried vias, and stacked via structures are avoided, as they severely restrict portability. Instead, through-hole vias and standard geometries are used. The recommended via sizes in Table A.1 provide a consistent baseline across modules and ensure compatibility with a broad range of PCB vendors.

Geometric Regularity and Ports Modules follow rectangular or near-square outlines and adopt local grids aligned with the pitch of their primary component. This regularity facilitates packing multiple modules into dense regions of complex boards, such as the power and clocking areas of Serenity-S1, where several reusable building blocks must be arranged efficiently around the FPGA and backplane connectors.

4.4 Applicability and Assessment

The Modular-Hardware Toolbox aims to provide reusable design artefacts that reduce development time, increase design reliability, and support collaborative workflows across multiple projects.

This section evaluates its applicability in practice and assesses how effectively the toolbox fulfils the requirements identified in Chapter 3. The discussion draws on experience gained from applying toolbox concepts under realistic project constraints during the Serenity-S1 development.

4.4.1 Applicability Across Design Contexts

The applicability of reusable modules depends on how effectively they can be integrated into different design environments without substantial rework. The toolbox is particularly well suited to research-oriented and low- to medium-volume designs, where development speed, risk reduction, and reproducibility provide clear value.

In research environments, module reuse reduces repeated design effort and mitigates errors by encapsulating validated functionality. Layout reuse is possible when boards share compatible constraints; when these differ, schematic reuse alone remains valuable by providing validated connectivity, configuration ranges, and integration guidance.

In contrast, high-volume industrial products often require cost-optimised custom layouts, which limits the direct reuse of the toolbox's layout modules. However, modules still support initial prototyping, feasibility studies, and early design phases where time-to-first-hardware is critical.

Toolchain portability remains a practical limitation. While the toolbox methodology is tool-agnostic, the implementation described in this thesis relies primarily on Altium Designer. Migration to other platforms such as KiCad or Cadence Allegro is feasible but may require adaptation, particularly for layout-level reuse.

4.4.2 Workflow Impact and User Accessibility

The toolbox modifies the traditional hardware design workflow by shifting effort from project-specific implementation to reusable module creation. Once a module is validated, subsequent projects benefit by integrating proven circuitry directly, without repeated data sheet interpretation or manual layout recreation.

In the conventional workflow shown in Figure 4.7, designers repeatedly consult data sheets when creating symbols, footprints, schematics, and layouts, often at non-consecutive stages. This creates gaps in understanding, increases cognitive load, and raises the probability of design errors. In contrast, the modular workflow encapsulates data sheet knowledge during module creation, reducing the need for repeated interpretation (see Figure 4.8).

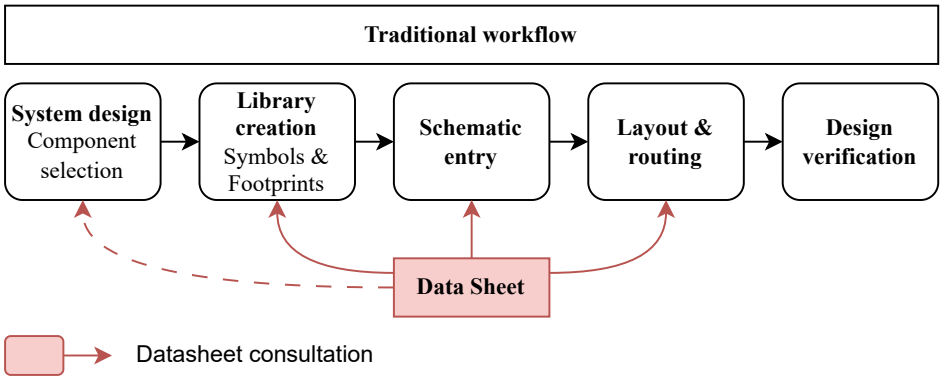


Figure 4.7: Traditional hardware design workflow, highlighting repeated data-sheet consultation as a source of error and inefficiency.

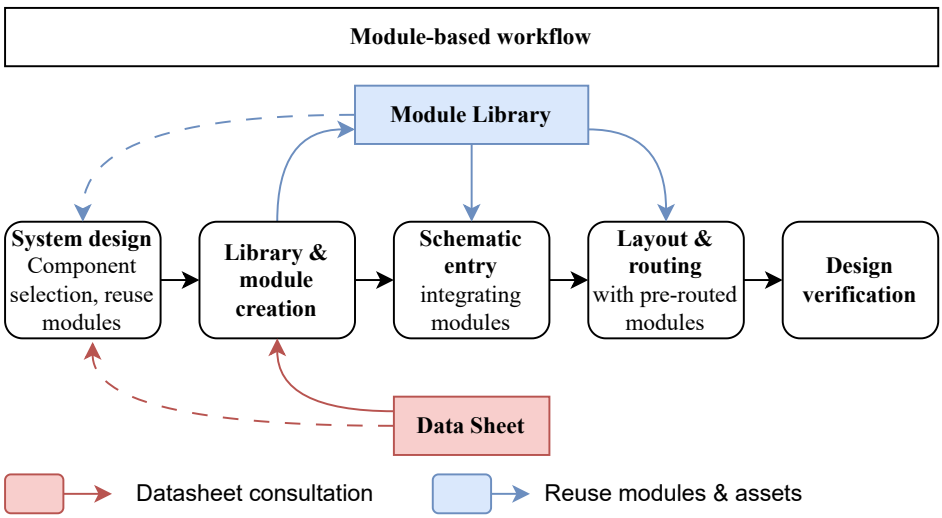


Figure 4.8: Module-based hardware design workflow, in which data-sheet knowledge is embedded during module creation rather than repeatedly re-evaluated.

Collaborative workflows also benefit. With modules clearly defined and stored in version control, teams can divide work along subsystem boundaries and validate modules independently. This reduces integration risk and shortens development cycles in distributed environments.

4.4.3 Challenges and Mitigation Strategies

Despite their benefits, reusable modules introduce challenges that require careful management.

Platform Consolidation vs. Controlled Diversity Reuse creates pressure to standardise around a limited set of component families, but the optimal degree of consolidation depends on how strongly a component couples to shared cross-project infrastructure. Processing platforms such as FPGAs typically require substantial non-local investment (power architecture, clocking strategy, configuration and management, firmware toolchains, verification environments, and long-term maintenance). Supporting multiple, similarly capable device families therefore duplicates this infrastructure and fragments development effort. A survey of FPGA devices used across recent institutional projects, summarised in Appendix A.2, shows a concentration on a limited set of processing platforms. In contrast, many analogue and mixed-signal functions (e.g. ADCs) impose significant local design effort, but often require comparatively limited shared firmware and verification infrastructure beyond the immediate interface, making controlled diversity more tolerable when driven by application constraints. For a reusable toolbox, this motivates deliberate consolidation where it yields multiplicative reuse benefits, while allowing diversity where requirements are genuinely application-specific.

Over-Specialisation Modules that encode highly specific design choices become difficult to reuse. To remain portable, modules must be general in function, support typical configuration ranges, and avoid integrating application-specific optimisations unless these are common across multiple designs.

Hidden Assumptions Modules must not rely on undocumented dependencies such as external pull-ups, termination networks, or filtering stages. Such omissions lead to fragile designs. The toolbox mitigates this through structured design contracts and mandatory review checklists.

Toolchain and Fabrication Dependence Layout reuse depends on fabrication capabilities of the PCB manufacturer. By adhering to relaxed fabrication rules and avoiding HDI features, the toolbox maximises cross-project compatibility.

Component Diversity Uncontrolled proliferation of passive-component values across modules increases procurement complexity and reduces consistency. The toolbox mitigates this by encouraging broad use of fully-specified components from the centralised library described in subsection 4.2.5.

5 Serenity-S1 ATCA Card

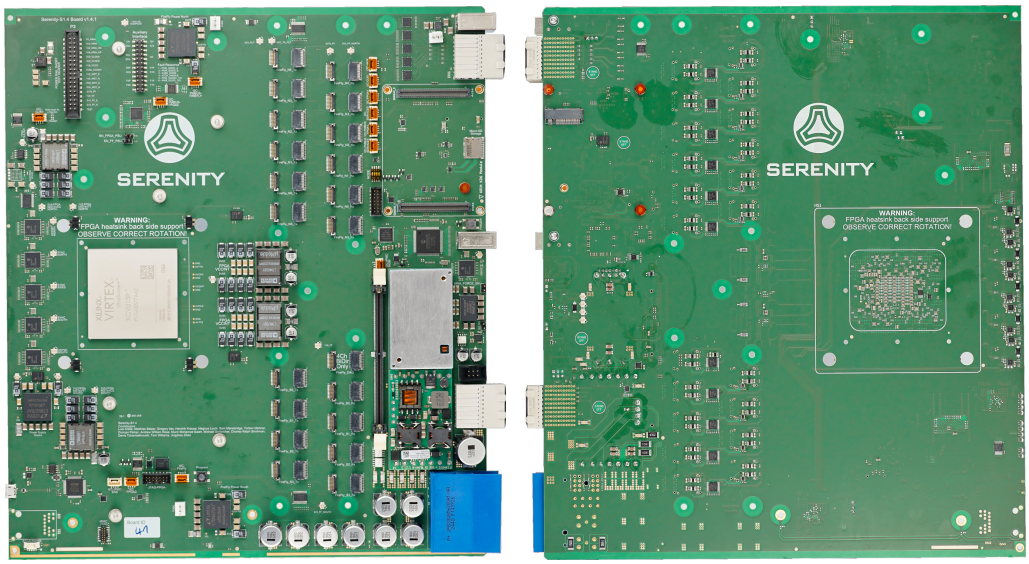


Figure 5.1: Serenity-S1 PCB front (left) and back (right) views.

The Serenity-S1 – shown in Figure 5.1 – constitutes a principal processing platform of the CMS Phase-2 back-end. Approximately 730 units are planned for production, with about 620 destined for installation in the detector back-end and the remainder allocated to spares and development. This makes Serenity-S1 the most widely deployed processing board of the upgrade and the workhorse of the CMS data acquisition and trigger system. The card integrates high-density optical I/O, deterministic low-latency processing, and embedded board management within a single ATCA form factor, enabling deployment across multiple subdetectors with distinct bandwidth and timing requirements. It also serves as the first large-scale realisation of the modular hardware toolbox, providing a production environment in which its design principles are applied and assessed.

5.1 Serenity ATCA Cards

The Serenity-S1 represents the first large-scale deployment of the modular hardware toolbox in a production environment, providing a high-performance and adaptable platform for multiple CMS subdetectors.

The development of Serenity-S1 can be traced back to early prototyping efforts undertaken during the global semiconductor shortage of 2020–2023, which required flexible approaches to component selection and verification. In response to limited availability, several power supplies and clock generators were evaluated in parallel as independently developed modules, later integrated into the board [187]. These constraints shaped the design methodology and strengthened the emphasis on modular prototyping and standalone validation. Serenity-S1 and the Modular-Hardware Toolbox evolved in a mutually reinforcing manner: practical needs from the board's development informed the structure of the Toolbox, while the emerging methodological framework guided subsequent design decisions. The Toolbox is presented in chapter 4, and the interplay between the two developments is discussed in chapter 6.

Table 5.1: Serenity-S1 usage in CMS subsystems.

Subsystem	**Quantity**
Outer Tracker (OT)	216
Beam Radiation, Instrumentation, and Luminosity (BRIL)	1
High-Granularity Calorimeter (HGCAL)	302
Level-1 Trigger (L1T)	49
MIP Timing Detector (MTD)	24
Drift Tubes (DT)	10
Resistive Plate Chambers (RPC)	20

While the Serenity-S1 was initially developed for the OT DAQ and Timing Card (DTC), the board's capabilities have since motivated its adoption across several subdetector systems in the CMS experiment, as can be seen in Table 5.1. Serenity-S1 provides substantial input/output flexibility, large FPGA processing capacity, and support for precision timing distribution, making it suitable for a wide range of applications. Consequently, Serenity-S1 has been adopted as a common processing platform across several CMS Phase-2 subdetectors, where it supports functions ranging from front-end data reception (e.g. OT, HGCAL, MTD) to trigger-primitive generation and back-end data routing (both HGCAL).

In summary, Serenity-S1 reflects both the modular design practices developed in this work and the concrete demands of CMS Phase-2, including high link density, strict latency constraints. These factors together established the card as a central component of the upgraded back-end and as a key component of the CMS Track Trigger and HGCAL subsystems.

5.1.1 Requirements

To ensure that Serenity-S1 could meet the diverse demands of CMS subsystems, its design was driven by a set of common and subdetector-specific requirements. These requirements encompass bandwidth, link density, latency, and precision timing, and they form the baseline against which the design and validation of the card must be assessed. A summary of subdetector-driven requirements is provided in Table 5.2, while the common system-level criteria are outlined below.

Table 5.2: Bandwidth and latency requirements of CMS subdetectors for the Serenity-S1 ATCA card [188, 189].

Serenity-S1 Use Case	**FE Links**[1] $5/10\,\mathrm{Gbit\,s^{-1}}$ lpGBT	**BE Links**[2] $16/25\,\mathrm{Gbit\,s^{-1}}$ CSP	**DAQ Links**[3] $25\,\mathrm{Gbit\,s^{-1}}$ Slink Rocket	**Latency**
OT DTC	72 Tx/Rx	48 Tx	4 Tx/Rx	0.4 µs
HGCAL DAQ	108 Tx/Rx	—	12 Tx/Rx	—
HGCAL TPG-S1[4]	120 Rx	108 Tx	4 Tx/Rx	0.4 µs
HGCAL TPG-S2[4]	—	12 Tx, 108 Rx	4 Tx/Rx	2.2 µs
BRIL	48 Tx	24 Tx/Rx	4 Tx/Rx	5 µs[6]
L1T[5]	—	120 Tx/Rx	4 Tx/Rx	—
MTD	108 Tx/Rx	12 Tx/Rx	—	5 µs[6]
DT	—	—	4 Tx/Rx	5 µs[6]
RPC	—	108 Tx, 60 Rx	4 Tx/Rx	5 µs[6]

[1] Front-End (FE) Links are using Low-power Gigabit Transceiver (lpGBT) links [190] to communicate with the detector front-end.
[2] Back-End (BE) Links are using CMS Standard Protocol (CSP) links [191] to communicate with other cards within the detector back-end.
[3] DAQ Links are using Slink Rocket links [43, 192] to communicate with the DTH400 and DAQ800 hub cards.
[4] HGCAL TPG-S1 and TPG-S2 are the two stages of Trigger Primitive Generation (TPG) at HGCAL.
[5] The L1T comprises multiple functionalities with different latencies [193].
[6] 5 µs is the maximum latency for signals to reach the correlator trigger [193].

Serenity-S1 must support a high density of FE and BE bi-directional optical links. In addition to the FE/BE links, the card must provide four dedicated DAQ links, with the option to repurpose BE links for additional DAQ capacity. Table 5.2 shows the bandwidth requirements from all

subdetectors using the Serenity-S1. The highest bandwidth is required at the L1T, where the system must support 120 bidirectional links, all capable of operating at $25\,\mathrm{Gbit\,s^{-1}}$.

The board must also support the TCDS2 [43] connection to both hub slots of the ATCA shelf. Synchronized precision timing is essential at the FE interfaces, reflecting the requirements of subdetectors such as HGCAL ($60\,\mu\mathrm{s}$ [194]) and MTD ($30\,\mu\mathrm{s}$ [23]). Meeting these budgets requires a consistent clock-tree jitter below $1.5\,\mathrm{ps}$. Several subsystems also impose strict latency constraints, requiring deterministic, low-latency paths for trigger and timing-sensitive applications, particularly in high-throughput environments such as HGCAL and L1T (see Table 5.2).

Power and thermal constraints further shape the design envelope. The VU13P FPGA consumes up to $170\,\mathrm{W}$ on the core, and another $44\,\mathrm{W}$ at the transceivers. The power delivery network must deliver low-impedance rails with voltage variation within $10\,\mathrm{mV}$ at the FPGA's core and transceiver supply pins, while the thermal design should maintain junction temperatures below $100\,^{\circ}\mathrm{C}$ [195]. Furthermore, the FireFly modules' temperatures should keep below $50\,^{\circ}\mathrm{C}$ to improve longevity [196, 2].

Collectively, these requirements define the design envelope of Serenity-S1 and establish the criteria against which its validation is assessed in the following sections.

5.1.2 Serenity Development Cards

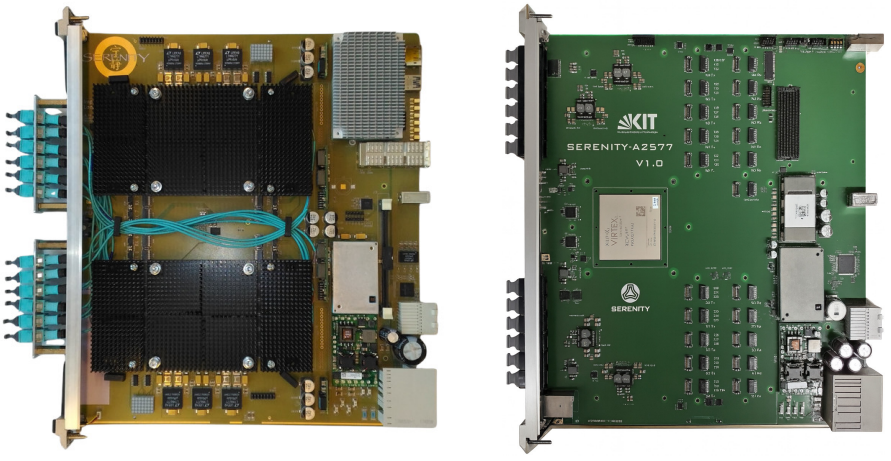


Figure 5.2: Serenity-Z development card [197] from Imperial College (left) and Serenity-A2577 development card [198] from KIT (right).

The Serenity-S1 evolved from two development boards, which are shown in Figure 5.2 and are designed to explore and validate architectural concepts. The first of these was the Serenity-Z,

developed at Imperial College [199]. To accommodate the high number of links specified in subsection 5.1.1, Serenity-Z employed two FPGAs mounted on interchangeable Samtec Z-Ray interposers. This architecture offered a high degree of flexibility by allowing the reassignment of I/O channels according to the specific use case. However, practical limitations emerged: the Z-Ray interposer's signal integrity was insufficient for the required bandwidth, reducing its speed rating from $25\,\mathrm{Gbit\,s^{-1}}$ to $14\,\mathrm{Gbit\,s^{-1}}$ [200] – a critical limitation for the Phase-2 upgrade, which requires $25\,\mathrm{Gbit\,s^{-1}}$.

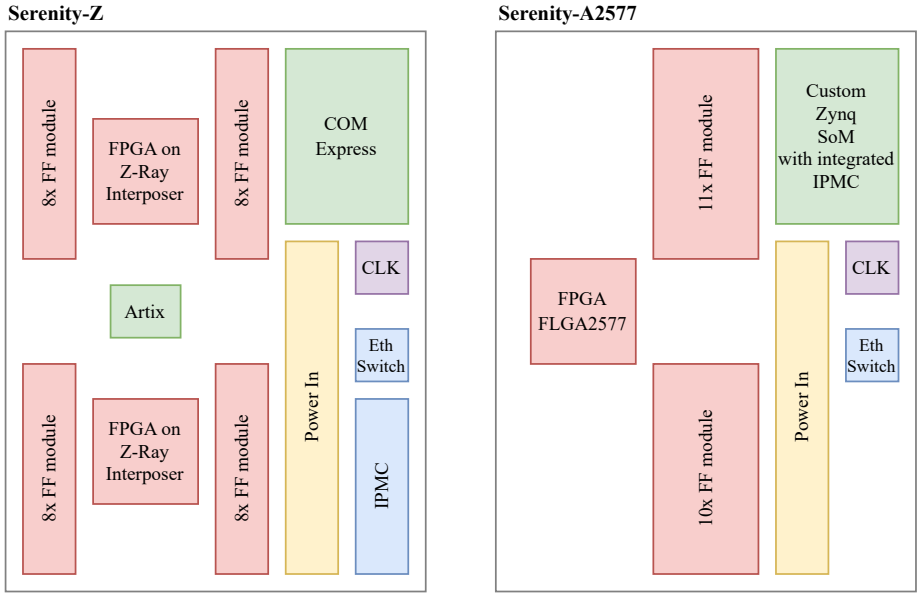


Figure 5.3: Block-level comparison between the Serenity-Z (left) and Serenity-A2577 (right) development cards.

In parallel, KIT initiated the development of the Serenity-A2577 [198], which already embodied several of the key features of Serenity-S1. This version integrated an AMD Virtex UltraScale+ VU13P FPGA directly onto the board, thereby avoiding the limitations of interposer-based integration. At the time of its development, this was the only device in AMD's catalogue offering the required transceiver density. The optical modules were deliberately arranged to minimise thermal coupling with the FPGA, extending the expected lifetime of the transceivers.

For control and management, the Serenity-A2577 replaced the x86-based COM Express architecture used in Serenity-Z with a custom-built mezzanine based on a Zynq UltraScale+ SoC [198, 1]. The Zynq Mezzanine could include the glue logic, for which the Aritx FPGA was needed in the Serenity-Z in the programmable logic as well as the IPMC functionality on the real-time cores of the processing system. This required special power-supply sequencing in the Zynq UltraScale+

SoC, because the real-time cores must be active in standby mode, while the rest is powered down to stay in the standby power budget, essentially demanding the custom build.

Collectively, these developments provided the foundation for Serenity-S1 by evaluating and validating key architectural concepts and exposing limitations that guided the production design. There were two cards planned for the Serenity main production, a Serenity-S1 with a single FPGA, which would conceptually be similar to the Serenity-A2577 and a Serenity-D1 with two FPGAs, which adopted the dual-FPGA concept of the Serenity-Z. For slow control, a module based on an Zynq UltraScale+ SoC should be used, eliminating the Artix FPGA, but the IPMC should not be integrated, in order to allow for a commercial module and simplify software development. To accommodate for the limited bandwidth of the Samtec Z-ray interposers, the Serenity-D1 would use soldered FPGAs as well, effectively fixing the configuration, however this version was ultimately discarded once subdetector requirements confirmed that Serenity-S1 alone would be sufficient.

In conclusion, the Serenity-Z and Serenity-A2577 demonstrated both the potential and the limits of early approaches, and their lessons were directly incorporated into the production card design.

5.1.3 Serenity Production Cards

The hardware design of the production cards commenced in spring 2022 as a collaborative effort between KIT and Imperial College, later joined by IHEP Beijing [2]. During this phase, the designs of the Serenity-Z and Serenity-A2577 were merged, combining their strengths into a unified architecture that addressed the performance and flexibility demands outlined in subsection 5.1.1. During the architecture design both cards – the Serenity-S1 and the Serenity-D1 – were still an option. Hence the board was planned to accommodate a common service area, that would satisfy both use cases and will be explained in more detail in 5.2.2.

A key change accompanying the transition to production was the migration to Altium Designer, replacing the earlier use of Mentor PADS for the prototyping cards. CERN proposed the switch to a tool that is available to all collaborators. Hence Altium Designer was selected, which required the entire design to be re-entered. While labour-intensive, this shift improves long-term maintainability and asset sharing across related projects.

Development followed the modular workflow introduced in Chapter 4. The schematic and layout were partitioned into well-defined hardware modules, with each designer responsible for a distinct subsystem: the ATCA power-entry circuitry, the Kria-based control section, the FPGA and supporting circuits, the clock and timing distribution network, the FireFly connectors, and the individual power supplies. These modules were developed independently on local design

copies and subsequently merged into the central repository. This modular workflow simplified coordination and reinforced design consistency across the system.

Because Altium Designer does not support concurrent editing within a single project, access to the central design repository was restricted to one contributor at a time. During the early development phase, this limitation was mitigated by the modular partitioning of the design: individual subsystems were developed and validated independently and integrated through short, controlled merge windows. During the later routing and integration phase, when cross-module connections became more frequent, a time-windowed handover workflow was adopted to maintain continuous progress. While this introduced additional coordination overhead, it demonstrates that modular design practices can decouple development tasks sufficiently to sustain parallel progress even under restrictive tool constraints.

Preparations for large-scale production progressed through a formal tender process covering PCB fabrication and assembly. Pre-production of 84 units commenced in August 2025, providing the basis for final validation and production-quality testing. The main production will follow in multiple batches throughout 2026 to meet the installation schedule for the CMS Phase-2 back-end [201].

5.2 Design and Architecture

The Serenity-S1 ATCA card is partitioned into two main domains: the service area and the payload area. This approach enables the reuse of common infrastructure across multiple Serenity-family cards, while providing flexibility to accommodate version-specific components in the payload region. The design choices presented here reflect both the requirements outlined in Section 5.1.1 and the lessons learned from previous Serenity development boards (see subsection 5.1.2). The overall design of the Serenity-S1 is similar to that of other ATCA cards developed for this upgrade, as was shown in subsection 3.1.1, which indicates the maturity of the design.

5.2.1 Serenity Framework

The overarching framework concept was established, when both Serenity boards (Serneity-S1 and Serneity-D1) should be built. It separates the service area – which includes recurring circuits that both boards would need – from the payload area. The service area implements power entry, board management, and ATCA-standard interfaces. The payload area integrates the main FPGAs and optical modules, together with their supporting power and clock distribution networks. This structure ensures that the core infrastructure remains reusable across all Serenity cards, while

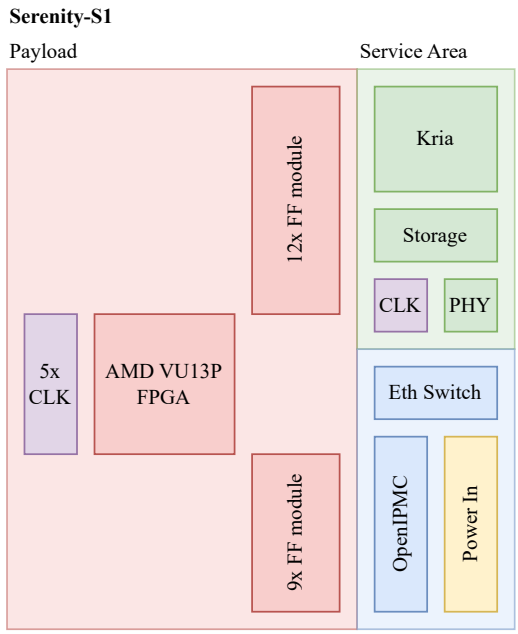


Figure 5.4: High-level block diagram of the Serenity-S1, indicating the separation between service and payload domains.

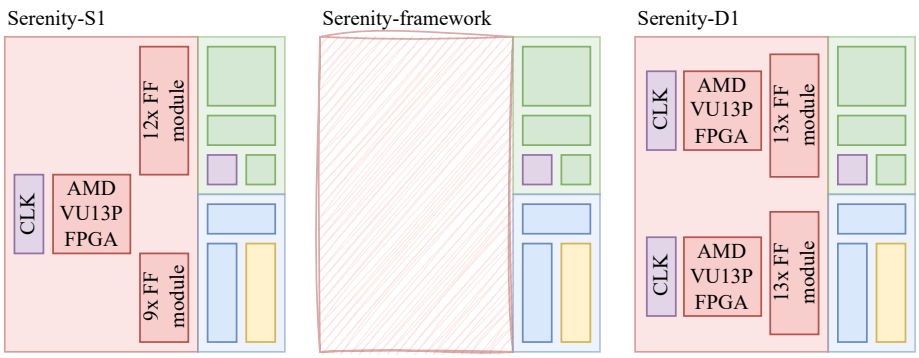


Figure 5.5: Serenity framework forming the common basis for the Serenity-S1 and Serenity-D1 cards.

allowing the payload to be adapted to specific needs. Figure 5.5 illustrates this concept using the example of the Serenity-D1.

The separation between service and payload domains was implemented using the module boundary definitions established during early development, allowing individual subsystems to be designed, verified, and iterated independently. These boundaries later formed the basis for the corresponding Modular-Hardware Toolbox workflows.

5.2.2 Service Area

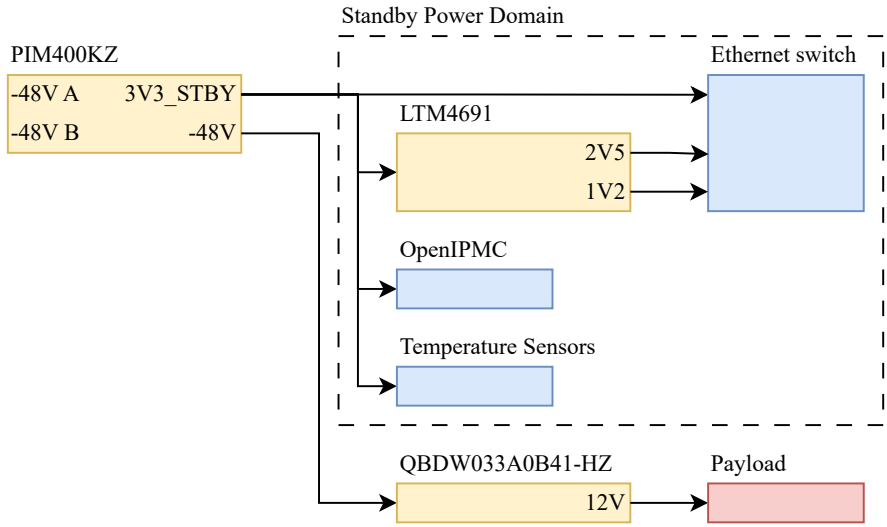


Figure 5.6: Block diagram of the standby power supply on the Serenity-S1.

The service area contains all ATCA-related electronics, most prominently the power-entry stage shown in Figure 5.6, which consists of a power-input module and an isolated DC/DC converter, the IPMC, and the networking infrastructure. The redundant $-48\,\mathrm{V}$ shelf supply is converted into a $3.3\,\mathrm{V}$ standby rail and a $12\,\mathrm{V}$ payload rail by an OmniOn PIM400KZ [202] and an OmniOn QBDW033A0B-HZ [203] quarter-brick converter, respectively. Both modules provide telemetry through $\mathrm{I^2C}$ to the IPMC. In accordance with the ATCA standard [70], the card remains in standby mode until the IPMC negotiates the power state with the shelf manager. Consequently, the OpenIPMC [204] and an Ethernet switch must be powered in standby to allow configuration and monitoring via Telnet. The IPMC is also responsible for temperature monitoring and is equipped with four on-board sensors and a remote sensor that measures the FPGA die temperature.

Board-level management is provided by a Kria K26 System-on-Module (SoM), based on the AMD Zynq UltraScale+ SoC. Its programmable logic instantiates glue logic such as $\mathrm{I^2C}$ and PMBus controllers, JTAG access, and the AXI Chip2Chip link [205] to the payload FPGA, as illustrated in Figure 5.7. Running AlmaLinux, the Kria SoM acts as the central slow-control processor and can be accessed either through Ethernet or via serial-over-LAN. The following peripheral circuits support the SoM:

- a microSD card for optional operating-system storage (the Serenity-S1 uses the Kria's on-board eMMC for the deployed system),

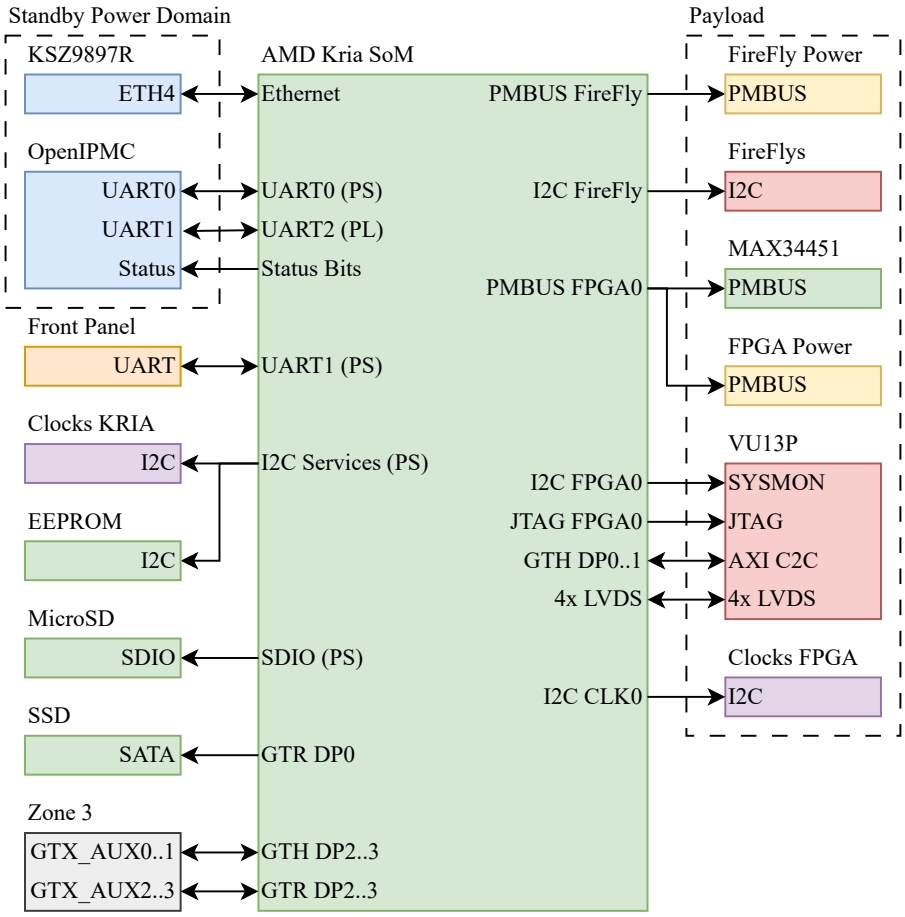


Figure 5.7: Block diagram of the Kria System-on-Module (SoM) connections on the Serenity-S1.

- a SATA solid-state drive for additional data storage,
- an EEPROM storing the Serenity-S1 board ID and revision,
- a programmable local oscillator providing the reference clock for AXI Chip2Chip, and
- a clock generator supplying the remaining reference clocks.

Although the Kria SoM and its peripherals are part of the service area, they are powered from the payload domain, as their consumption exceeds the available standby power and they are not required until payload control begins. This does not introduce issues with uninitialised devices at power-up: a dedicated power sequencer in the payload domain holds all payload rails disabled until the Kria explicitly enables them in the correct sequence.

Many circuits in the service area – particularly the power-entry stage, standby supply, IPMC infrastructure, and Kria-based control section – were implemented as standalone hardware modules with defined electrical and interface specifications. This modular approach allowed these subsystems to be developed, reviewed, and validated independently before integration, and later enabled their reuse within other Serenity-family cards.

In summary, the service area forms the operational backbone of Serenity-S1, implementing all ATCA-mandated management functions and providing a stable, reusable control infrastructure that operates independently of the payload.

5.2.3 Payload Area

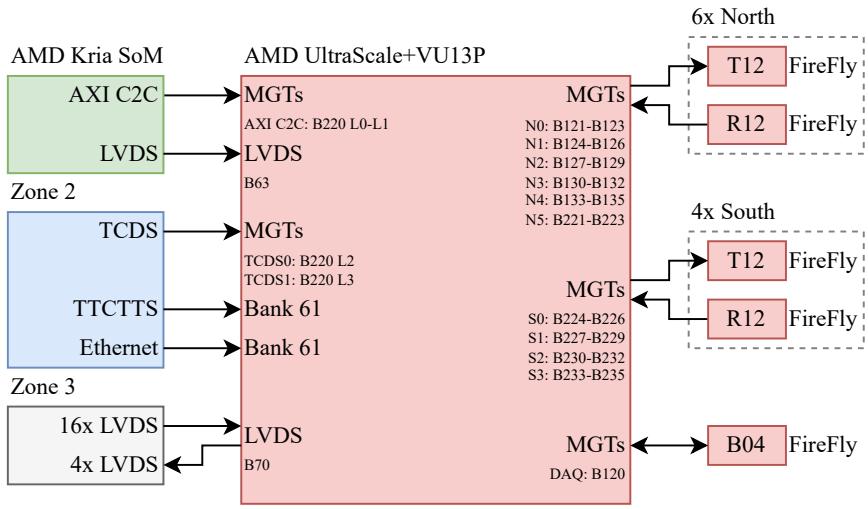


Figure 5.8: Block diagram of the FPGA data-path connections on the Serenity-S1.

The payload area hosts the AMD Virtex UltraScale+ VU13P FPGA and connectors for optical transceiver modules. A total of 21 Samtec FireFly modules are mounted, providing ten 12-channel transmitters, ten 12-channel receivers, and one four-channel bidirectional interface. Each channel supports data rates up to $25\,\mathrm{Gbit\,s^{-1}}$, yielding an aggregate throughput of approximately $3.1\,\mathrm{Tbit\,s^{-1}}$ in each direction. Figure 5.8 shows the data links connected to the FPGA on Serenity-S1.

The payload area likewise employed a modular design strategy. The high-current regulators, FireFly transceiver blocks, clock-generation chain, and FPGA support circuits were each implemented as separate modules with clearly defined interfaces. This structure simplified iterative redesign

during the S1.1–S1.3 prototyping phase, where several power supplies and clock components could be replaced without architectural changes, and ensured consistency across the more than twenty transceiver blocks.

The FPGA is configured and controlled by the Kria SoM via a JTAG and AXI Chip2Chip link. Synchronisation is achieved through the TCDS2 [43, 206], distributed from the DTH400 over the ATCA backplane and fanned out by a hierarchy of ZL30274 jitter-cleaners. This clock tree delivers synchronous and asynchronous clocks with sub-$1.5\,\mathrm{ps}$ jitter, meeting CMS timing requirements [2, 23].

Power distribution in the payload area is visualised in Figure 5.9. All voltages are generated from the intermediate $12\,\mathrm{V}$ rail and converted locally into 16 dedicated supplies. Two parallel LTM4700 converters deliver up to $200\,\mathrm{A}$ to the FPGA core, while additional regulators serve the transceivers, FireFly modules, and auxiliary logic. Linear regulators with high power-supply rejection are employed for the jitter-critical clock networks, ensuring low-noise operation. To coordinate sequencing and fault handling, a power supervisor and sequencer monitors all payload rails and controls power-up and power-down sequences. The system is able to react to supply faults or crate-level events by initiating a controlled shutdown, making use of the energy stored in hold-up capacitors. This design protects critical components and ensures compliance with ATCA reliability requirements.

5.2.4 Engineering Challenges

The Serenity-S1 had to address multiple engineering challenges arising from the coexistence of high-speed serial I/O and high-current power delivery on a single ATCA card. Data transmission requirements of up to $25\,\mathrm{Gbit\,s^{-1}}$ per channel and currents of up to $200\,\mathrm{A}$ for the FPGA core demanded a carefully optimised printed circuit board (PCB) stack-up and thermal management strategy.

In addition to these electrical and mechanical constraints, the development coincided with the global semiconductor shortage (2020–2023), which constrained the availability of several critical components [207, 208]. Although many risks were mitigated through early purchasing, the low-jitter clock-generation device originally selected for FPGA reference-clock conditioning became unobtainable without a reliable delivery horizon. To avoid a late dependency on an uncertain delivery schedule, an alternative jitter-cleaner (Microchip ZL30274) was identified and qualified ahead of system integration.

Because no commercial evaluation platform was available at the time, a dedicated evaluation board was produced around the reusable clock module intended for Serenity-S1, complemented by a minimal wrapper providing power entry and measurement interfaces (e.g. SMA outputs).

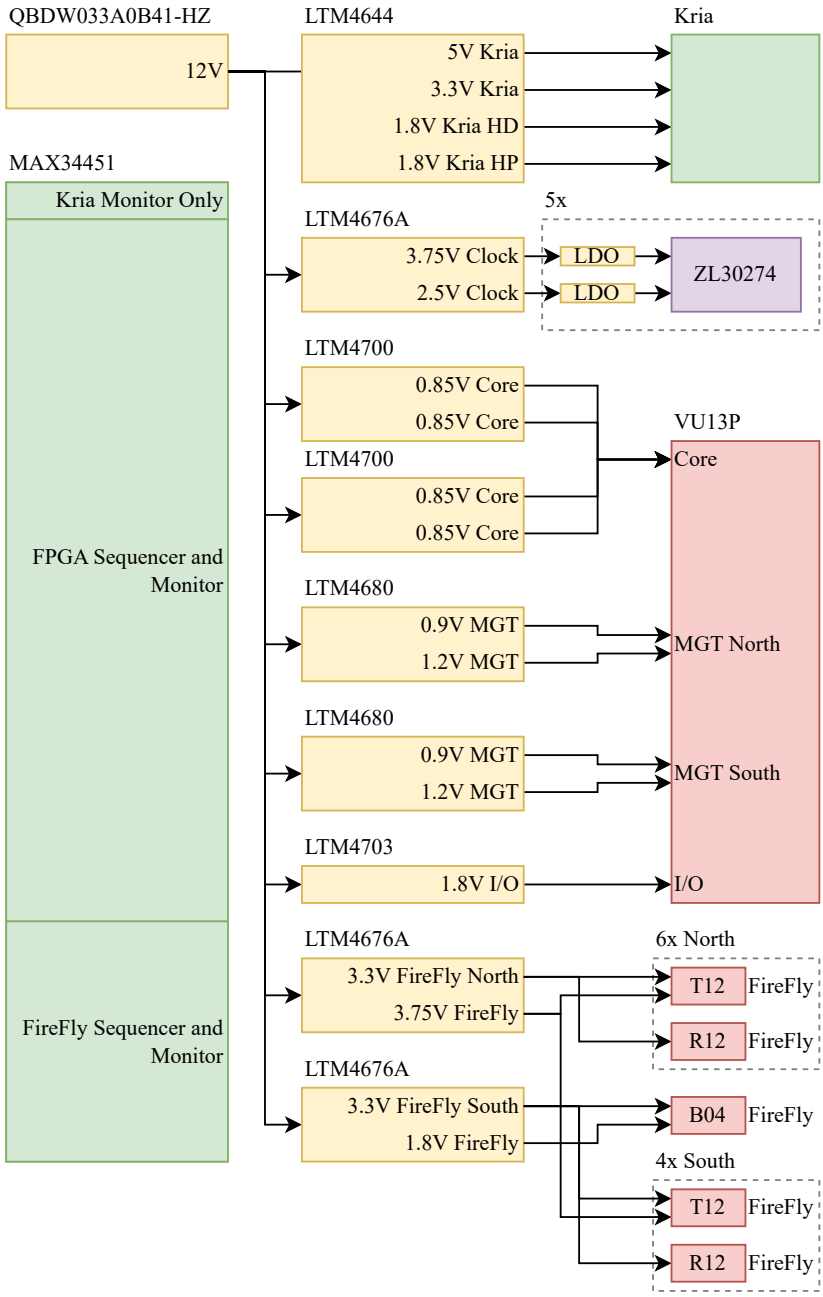


Figure 5.9: Power distribution network of the Serenity-S1. The MAX34451 controls the power sequencing of the FPGA and the FireFly power supplies individually.

This enabled stand-alone characterisation of functional behaviour and jitter performance under controlled supply conditions and supported CERN-based qualification for CMS use [187]. Beyond resolving an immediate supply constraint, this episode illustrates a practical benefit of the Modular-Hardware Toolbox: reusable modules can be instantiated with limited additional design effort in lightweight test PCBs, enabling parallel validation and reducing the likelihood that late component substitutions propagate into system-level redesign.

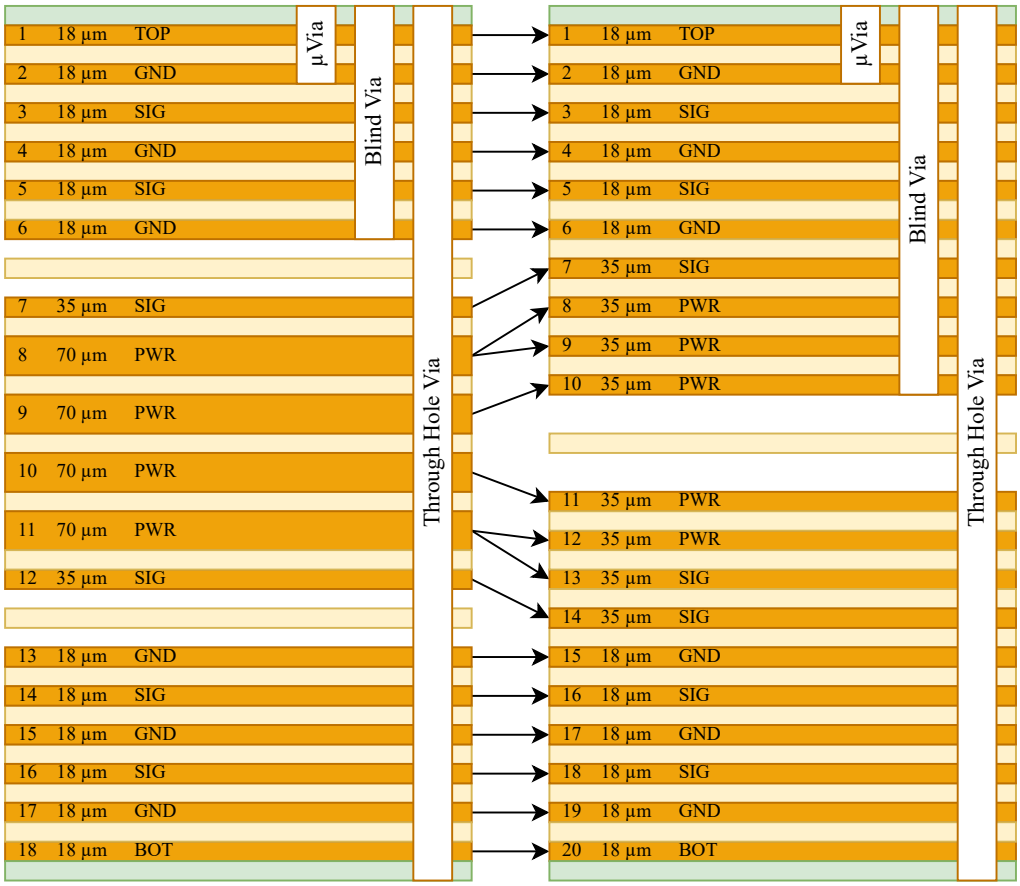


Figure 5.10: Layer stacks of the Serenity-S1. The 18-layer stack (left) is fabricated in three sub-stacks, and the 20-layer stack (right) is fabricated in two sub-stacks.

The PCB design employs four impedance-controlled inner layers for high-speed signals on layers 3, 5, 14, and 16[1]. These are enclosed by uninterrupted ground planes on layers 2, 4, 6, 13, 15, and 17 to ensure stable return paths and to limit crosstalk. Microvias and back-drilled vias are used

[1] Layer numbering follows the 18-layer stack shown in Figure 5.10.

to minimise stubs and maintain signal integrity. Layers 8–11 form copper-rich, low-resistance distribution paths for high-current rails. Overall, the 18-layer configuration provides sufficient routing resources and power-distribution capability for the board's mixed-signal requirements.

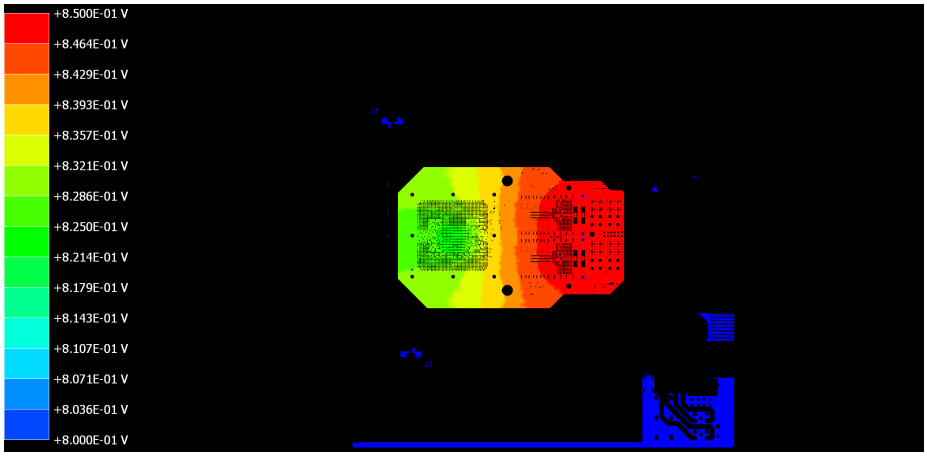


Figure 5.11: DC simulation of the Serenity-S1 showing the VCCINT plane on layer 8 at 150 A

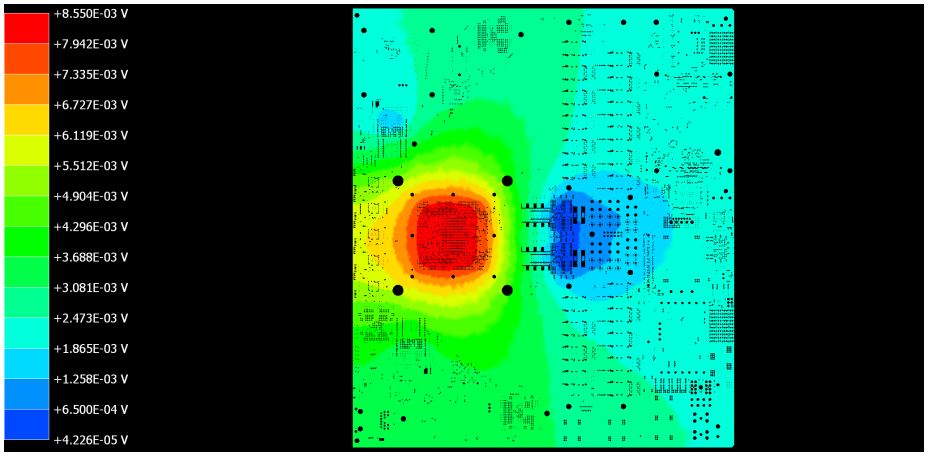


Figure 5.12: DC simulation of Serenity-S1, showing the GND plane on layer 13 as a representation of all ground planes.

A higher layer count was deliberately avoided. Adding further layers could increase the board thickness beyond the $2.4\,\mathrm{mm}$ limit defined by the ATCA mechanical specification [70] and would substantially raise fabrication cost. To retain manufacturability by a wide range of PCB manufacturers, the stack-up was chosen such that it can be implemented either as an 18-layer triple sub-stack or, alternatively, as a 20-layer dual sub-stack (see Figure 5.10). In line with

CERN's recommendation to use halogen-free materials, the prepregs EMC EM-890 [209] and TUC ThunderClad 2A Sp [210] were identified as suitable options.

To validate the stack-up, DC simulations for all power rails were performed using Ansys SIwave [211]. For example, the VCCINT simulation at the projected current of $150\,\mathrm{A}$ in Figure 5.11 shows a voltage drop of $25\,\mathrm{mV}$, compensated by remote feedback from the regulator. The drop across the FPGA itself is as low as $10\,\mathrm{mV}$, which is acceptable. Figure 5.12 confirms that sufficient return paths exist in the ground planes, with a maximum drop of only $8\,\mathrm{mV}$.

Thermal management on Serenity-S1 is constrained by the ATCA form factor and the vertical airflow provided by the crate. The ATCA specification limits component height to $21.33\,\mathrm{mm}$, necessitating low-profile heatsinks, and the inlet-air temperature inside CMS ATCA crates is typically in the range of $26\,^{\circ}\mathrm{C}$ to $30\,^{\circ}\mathrm{C}$. These boundary conditions define the thermal headroom available to the board and are particularly relevant for the FireFly optical modules, which must remain below $50\,^{\circ}\mathrm{C}$ to achieve their required multi-year operating lifetime.

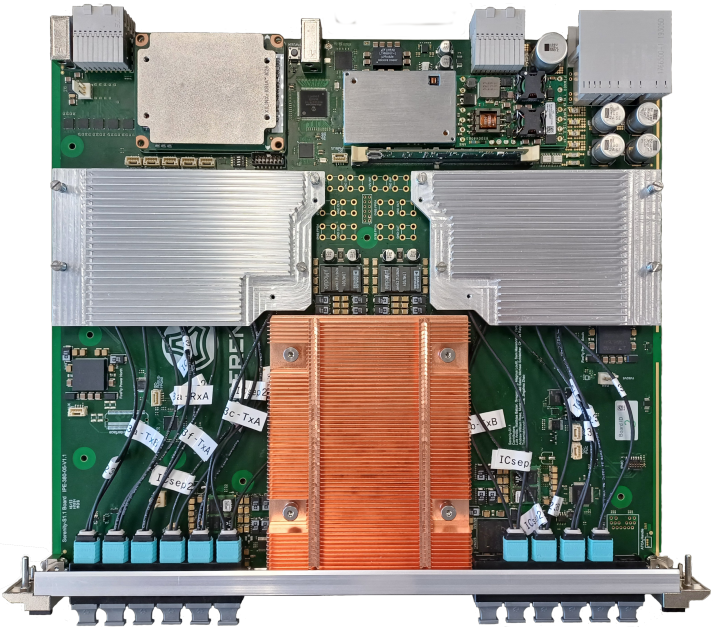


Figure 5.13: Serenity-S1 equipped with custom aluminium heat sinks for the FireFly modules and a custom copper heat sink with vapour chamber for the FPGA.

Within these limits, Serenity-S1 adopts a component placement strategy optimized for the airflow profile. The FireFly modules are positioned offset from the FPGA so that they receive unheated inlet air rather than air warmed by the FPGA. Measurements on the Serenity-A2577 prototype confirmed that the resulting increase in trace length does not compromise signal integrity (BER <

10^{-12}) [2]. To ensure efficient heat extraction, custom low-profile heatsinks – shown assembled in Figure 5.13 – were developed for both component groups: aluminium heatsinks for the FireFly devices and a vapour-chamber copper heatsink for the FPGA to enhance heat spreading while respecting ATCA height constraints. These measures provide stable thermal operation of the board within the expected inlet-air conditions.

Taken together, the routing, power-distribution, thermal, and component-qualification constraints of Serenity-S1 illustrate the tight coupling between electrical, mechanical, and supply-chain boundary conditions on a high-performance ATCA card. The selected stack-up provides controlled-impedance routing for more than one hundred high-speed links while accommodating low-resistance planes for the FPGA's high-current rails, and DC simulations confirm adequate voltage margin under projected load. Thermal placement and custom low-profile heatsinks sustain these electrical capabilities within the inlet-air temperatures and mechanical limits of the ATCA environment. In parallel, the modular implementation enabled rapid stand-alone qualification of substituted clocking components during the semiconductor shortage, reducing integration risk.The resulting design satisfies the signal-integrity, power-integrity, and thermal-margin constraints identified for the CMS Phase-2 upgrade, while remaining manufacturable and compliant with ATCA specifications.

5.2.5 Firmware and Software Architecture

The Serenity-S1 integrates a layered firmware and software architecture that separates low-level hardware management from high-level user interfaces. Figure 5.14 shows this layered architecture. At its foundation, the Kria SoM runs a minimal AlmaLinux operating system and performs a full network boot according to the Split Boot v2 scheme [4]. This approach simplifies maintenance and ensures consistent deployment across systems.

Low-level control is provided by the Serenity Management Shell (SMASH) [212], a modular framework that mirrors the hardware and FPGA structure through configuration scripts. SMASH supports multiple interfaces – including GPIO, I^2C, SPI and JTAG, providing abstracted access to power supplies, FireFly modules, PLLs, the FPGA and other peripherals. On the Serenity-S1, ten independent I^2C buses are instantiated in the programmable logic of the Kria SoM, with access through IPbus [213].

Within the processing FPGA, the Extensible Modular Processing (EMP) framework [43, 214, 215] provides the firmware infrastructure for high-speed I/O, data buffering, and control. Algorithm-specific logic is isolated in a dedicated EMP-payload, which is embedded into the EMP-infrastructure. Control and monitoring registers are exposed via IPbus, with the command-line

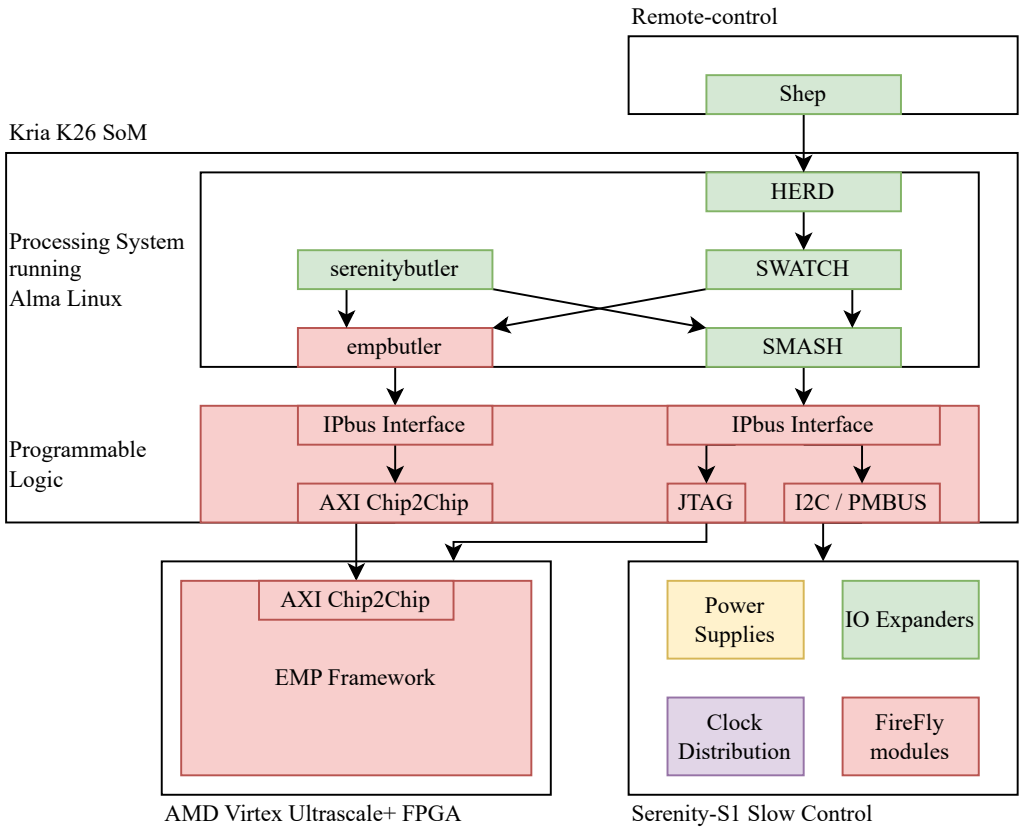


Figure 5.14: Layered firmware and software architecture of the Serenity-S1 platform.

tool empbutler serving as the primary utility for tasks such as resetting the framework, configuring links, or injecting and capturing data buffers.

Above these layers, additional software provides higher-level abstraction and integration. The serenitybutler package [216] builds on SMASH and empbutler to expose simplified commands for routine operation. At experiment level, SWATCH [217] standardises monitoring and control across CMS hardware, using plug-ins to provide a common interface to Serenity-specific features. Finally, the HERD server running on the Serenity-S1 enables remote access to these plug-ins and integrates with the Shep GUI, forming the ShepHERD online control framework [218].

Taken together, this layered architecture ensures reliable low-level control, robust firmware infrastructure, and user-friendly integration into the global CMS data acquisition system, making Serenity-S1 both scalable and maintainable.

5.3 Tests and Results

The first Serenity-S1 prototypes (revision S1.1) became available in December 2023, enabling initial validation of the power distribution network and high-speed serial transceivers. Early measurements revealed excess noise on the MGT power rails, which degraded link performance. Successive revisions addressed this by replacing the original regulators with low-noise alternatives, leading to revision S1.3, which forms the basis for the results presented in this section. The production revision (S1.4) from April 2025 introduced only minor adjustments to slow-control resets and mounting holes, with no impact on overall performance.

5.3.1 Power supplies

The Virtex UltraScale+ VU13P FPGA can consume currents exceeding $200\,\mathrm{A}$ on VCCINT, even though in practical used, the current is projected to be around $150\,\mathrm{A}$ along with low-noise rails for its multi-gigabit transceivers. In early prototypes, significant voltage ripple on the MGT supplies was traced to the original regulators. This was mitigated by replacing the TPSM8D6B24 module, specified with $20\,\mathrm{mV_{PP}}$ [219] with an LTM4680 with an output ripple of only $10\,\mathrm{mV_{PP}}$ [220]. Similarly, the FPGA I/O supply (VCCO) introduced noise into neighbouring power planes in revisions S1.1 and S1.2, and was replaced by an LTM4703 [221] from revision S1.3 onwards. These replacements were simplified by the modular design approach, as only the modules had to be replaced.

Table 5.3: Ripple measurements of FPGA power supplies on the Serenity-S1.

Power Rail	**Measured** Peak-to-Peak [mV]	**Limit** Peak-to-Peak [mV]
0.85V VCCINT	15.163	50.000
0.9V MGTAVCC North	47.192	54.000
0.9V MGTAVCC South	46.019	54.000
1.2V MGTAVTT North	38.979	72.000
1.2V MGTAVTT South	40.216	72.000
12V Payload Supply	283.960	undefined
1.8V VCCIO	11.765	108.000

Ripple and transient performance were characterised using micro coaxial connectors placed close to the power supplies, enabling straightforward oscilloscope measurements. Table 5.3 summarises the measured ripple values. Across all payload rails, the peak-to-peak ripple remained

below the limits recommended by AMD for Virtex UltraScale+ devices [195]. Although the observed ripple exceeds the regulator-level specifications, the measurement is taken directly at the converter output; the downstream decoupling network – comprising the capacitors located at the FPGA and the power-plane capacitance – further attenuates the high-frequency components. Figure 5.15 illustrates the ripple measurement in both time and frequency domains. The spectral representation confirms that the dominant ripple component corresponds to the power supply's switching frequency.

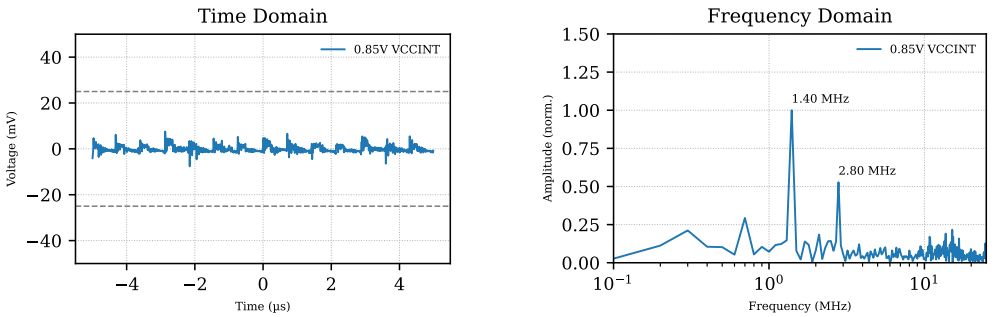


Figure 5.15: Measurement of the LTM4700 output ripple on the 0.85 V VCCINT rail in the time domain (left) and frequency domain (right). The dashed horizontal lines indicate the allowable peak-to-peak ripple limit, and the frequency spectrum is normalised to the effective 1.4 MHz switching frequency resulting from four interleaved 375 kHz phases.

Worst-case transient behaviour was assessed with dedicated firmware that simultaneously enabled all MGTs and implemented oscillating logic to generate a variable static load on VCCINT. Table 5.4 shows the minimum and maximum voltages measured during these transients. The load introduced by the oscillating logic was programmable, and tests were performed over a range of 120 A to 152 A. A sharp step from 152 A to 10 A triggered the over-voltage protection of the LTM4700, which had been conservatively set at 910 mV to ensure a large safety margin.

In practice, such extreme load steps will not occur in the CMS application, since the firmware performs sequenced power-up and power-down, dividing large transients into multiple smaller ones. Overall, the measurements confirm that the Serenity-S1 power supplies meet specifications and maintain stable operation even under the worst-case conditions created for the test. This low-noise power delivery is a critical prerequisite for reliable performance of the high-speed lanes.

Table 5.4: Voltage transient behaviour of FPGA power supplies on the Serenity-S1 under dynamic load conditions.

Power Rail	Step	**Measured [V]**		**Limit [V]**	
		Min	Max	Min	Max
0.85V VCCINT	10.0 A to 144.0 A	0.809		0.680	1.000
	144.0 A to 10.0 A		0.906	0.680	1.000
0.9V MGTAVCC North	0.2 A to 6.6 A	0.883		−0.500	1.000
	6.6 A to 0.2 A		0.928	−0.500	1.000
0.9V MGTAVCC South	0.2 A to 6.6 A	0.882		−0.500	1.000
	6.6 A to 0.2 A		0.933	−0.500	1.000
1.2V MGTAVTT North	0.2 A to 15.6 A	1.116		−0.590	1.300
	15.6 A to 0.2 A		1.263	−0.590	1.300
1.2V MGTAVTT South	0.2 A to 15.6 A	1.156		−0.500	1.300
	15.6 A to 0.2 A		1.267	−0.500	1.300

5.3.2 High-speed lanes

The Serenity-S1 routes 124 of the FPGA's 128 MGTs to FireFly connectors, two to the Kria SoM via AXI Chip2Chip, and two to the ATCA backplane, as Figure 5.16 shows. Signal integrity was validated using the FPGA's integrated bit error rate test (IBERT), with bathtub scans extrapolated using a dual-Dirac fit [222] to the CMS Phase-2 requirement of BER 10^{-12}.

Passive copper loopbacks provided the most stringent test, as signals traversed the complete electrical path without Clock-Data Recovery (CDR). Table 5.5 summarises the results: eye openings extrapolated to BER 10^{-12} ranged from $21.1\,\%$ to $54.8\,\%$, with the smallest margins corresponding to the longest traces ($\sim$270 mm). Optical loopbacks consistently yielded larger eye openings above $45\,\%$, demonstrating the positive impact of CDR in the FireFly modules.

Representative eye diagrams and their corresponding bathtub curves for different trace lengths are shown in Figure 5.17 to Figure 5.20. These scans show one lane (Lane 0) of the optical modules on the four southern FireFly connectors. These results confirm that Serenity-S1 still has comfortably large eye openings at BER = 10^{-12}, validating the chosen PCB stack-up, routing strategy, and transceiver configuration. Extending the dual-Dirac fit for the intended runtime of 12 years [201] (10^{-20}) shows that the only the longest traces (S3) are expecting any bit errors at all.

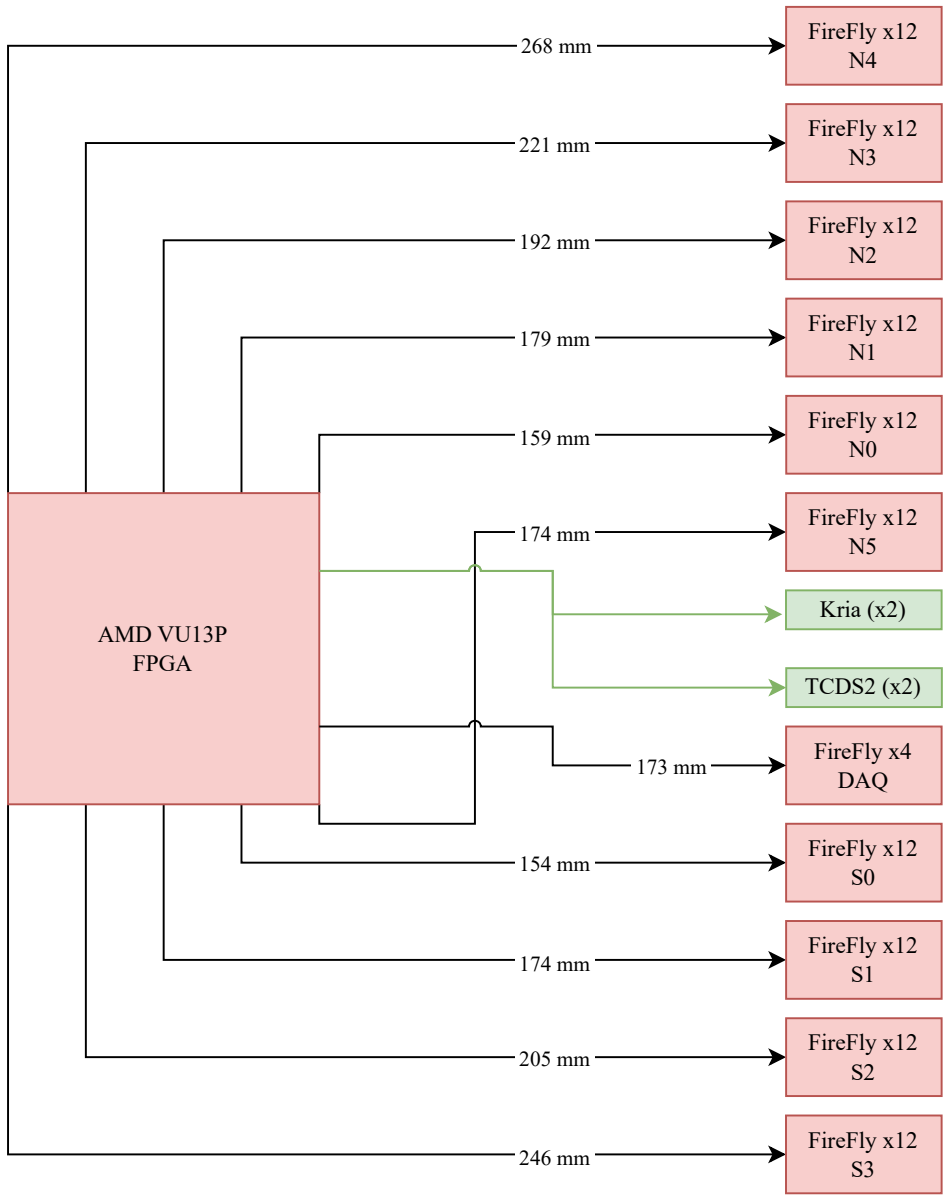


Figure 5.16: Positions of the FireFly connectors and the additional Kria and TCDS2 links relative to the FPGA. Arrow origins indicate whether a connection maps to the north or south MGT banks, and the numbers denote the maximum lane length within each FireFly module.

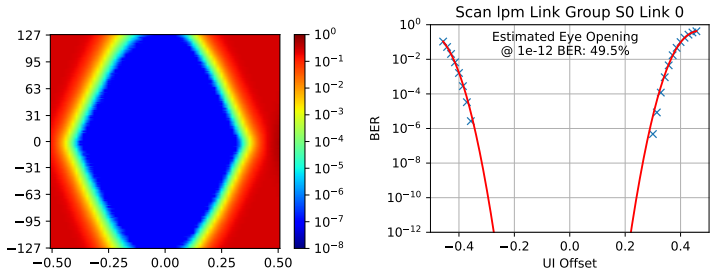


Figure 5.17: Eye diagram and bathtub curve of S0 Lane 0 (trace length: 135.5 mm).

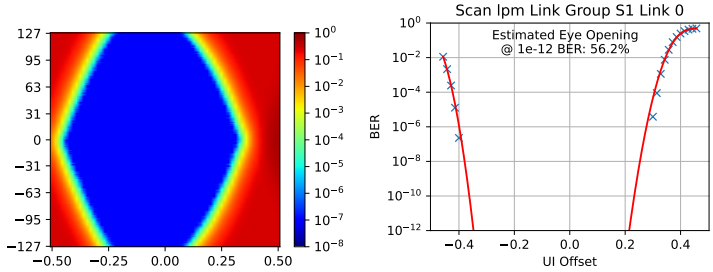


Figure 5.18: Eye diagram and bathtub curve of S1 Lane 0 (trace length: 155.5 mm).

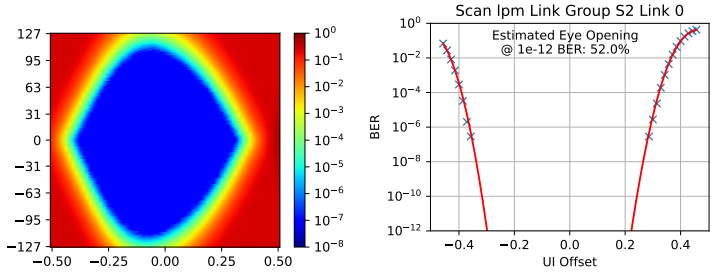


Figure 5.19: Eye diagram and bathtub curve of S2 Lane 0 (trace length: 176.9 mm).

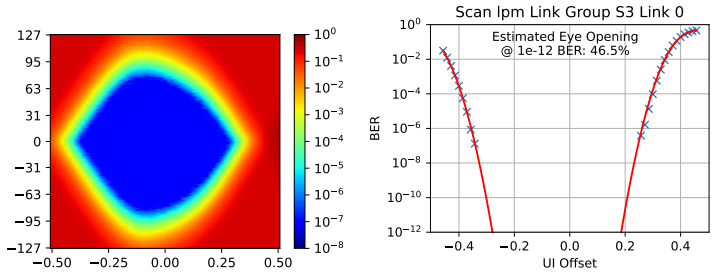


Figure 5.20: Eye diagram and bathtub curve of S3 Lane 0 (trace length: 202.1 mm).

Table 5.5: Interpolated eye-opening margins at BER = 10^{-12} from bathtub scans. Table entries reflect physical link positions on the PCB; the FPGA row indicates device location.

FireFly	Trace Length [mm]	Eye opening (passive) [%]			Eye opening (optical) [%]		
	(longest RX)	Min	Avg	Max	Min	Avg	Max
N4	267.7	32.0	40.6	48.7	45.0	49.4	53.7
N3	221.0	33.7	42.0	53.3	45.9	51.8	57.3
N2	192.4	36.0	42.7	54.2	47.3	51.0	55.4
N1	178.5	38.2	44.4	48.7	45.5	50.8	54.4
N0	159.3	36.5	43.1	49.4	45.6	52.9	57.9
N5	174.3	31.2	40.1	51.9	51.1	53.1	55.4
			FPGA				
DAQ	172.8	34.2	40.9	46.2	63.6	65.9	66.7
S0	153.5	30.5	43.3	52.9	49.5	51.7	54.1
S1	174.0	37.5	45.4	54.8	48.9	52.0	56.3
S2	205.3	28.0	38.4	46.6	47.4	52.0	55.5
S3	245.6	21.1	36.6	50.4	46.5	49.8	53.8

5.3.3 Production tests

In preparation for large-scale deployment, a factory acceptance test (FAT) was developed to verify functionality across all produced boards [223]. The FAT is executed at the PCB assembler and commissions a board in under ten minutes. Following automated optical inspection, the board undergoes an electrical short test of all power rails using a multi-channel digital multimeter. The measured impedances are compared against a reference set of known-good values and subsequently stored in the database.

Afterwards, an OpenIPMC, a Kria SoM, and copper FireFly loopback cables are installed, and the automated self-test sequence is initiated. Each step must complete successfully before the next is executed; after the sequence concludes – whether due to a failure or after successful completion – the board is returned to a safe, powered-down state, after which the FAT report and all associated logs are generated and uploaded to the database.

1. **IPMC-based tests**

    a) Read all sensor and verify results.

2. **Kria-based tests using SMASH**

    a) Program the BoardID into the on-board EEPROM.

b) Probe and configure the power supplies and power sequencer via I^2C.

c) Enable the power rails and verify their output voltages.

d) Probe and configure all clock-generation devices via I^2C.

e) Probe additional I^2C-connected devices.

f) Probe the FPGA via JTAG and load a test bitstream.

3. **FPGA-based tests using EMP-toolbox and serenitybutler**

a) Verify the AXI Chip2Chip link between the Kria and the FPGA.

b) Measure the clock signals supplied to the FPGA.

c) Perform a high-speed loopback test, enabling only sixteen channels at a time to avoid thermal stress on the FPGA, which is tested without a heatsink.

Once delivered to CERN, the boards are assembled with their final OpenIPMC, Kria, optical FireFly modules, and heat sinks, and undergo the User Acceptance Test (UAT). The UAT mirrors the FAT but – enabled by mounting the heat sinks – includes extended high-speed link testing to ensure full compliance with CMS requirements.

Finally, burn-in tests are planned to identify early-life failures in accordance with the bathtub reliability model [224]. Boards will be operated under nominal conditions for extended periods, ensuring that infant mortality failures are detected before deployment. This strategy supports the expected operational lifetime of 12 years for CMS Phase-2 [201].

The test campaign confirms that Serenity-S1 meets the functional and performance requirements for CMS Phase-2 operation. The power-distribution network delivers low-noise, specification-compliant rails with stable behaviour under dynamic loading, validated through ripple and transient measurements following targeted regulator replacements. High-speed link characterisation shows that all 124 routed MGT lanes achieve eye openings exceeding $40\,\%$ at BER = 10^{-12}. Reproducible commissioning and quality assurance are ensured through production-scale FAT and UAT procedures, covering electrical integrity, slow-control functionality, and high-speed interfaces. Prototype and pre-production boards have already been deployed in subsystem tests for HGCAL [225], BRIL [226], MTD [227], and the Global Trigger [228], demonstrating readiness for system-level integration within CMS. With delivery of the pre-series cards in early 2026, full vertical subsystem tests will become possible.

5.4 Potential Revision

With the Serenity-S1 design now validated across power integrity, high-speed links, and production testing, it is timely to consider how the platform might evolve in future hardware generations. The experience gained during component qualification, prototyping, and large-scale commissioning reveals both enduring strengths of the current architecture and structural limitations that may constrain its long-term applicability. These insights motivate a forward-looking assessment of which design principles should be preserved and which aspects may benefit from architectural refinement or a transition to an alternative form factor.

Several characteristics of Serenity-S1 have proven central to its success and should be retained in any successor design. First, the board maintains conceptual simplicity: it is a monolithic FPGA platform that exposes its full I/O bandwidth through optical interfaces, avoiding interchangeable modules and thereby limiting system complexity. Second, the Kria-based management architecture has demonstrated robust performance across prototypes and in multi-board subsystem environments, and the associated software infrastructure already scales from small setups (i.e. TRISTAN, BRIL) to installations comprising hundreds of boards (i.e. Outer Tracker, HGCAL). Reusing this management approach would therefore provide continuity and reduce integration overhead in future iterations.

Together, these features form a stable foundation – a high-bandwidth FPGA core paired with a reliable and scalable management subsystem – from which a next-generation Serenity-S2 could be developed.

Upgraded processing and memory resources At the time of design, the Serenity-S1 employed the largest available FPGA by bandwidth, the AMD Virtex UltraScale+ VU13P. Since then, AMD has released second-generation Versal devices, which offer significantly more logic resources, advanced I/O capabilities, and integrated AI/ML accelerators. A Serenity-S2 should therefore adopt a high-end AMD Versal SoC to extend processing capacity and bandwidth.

Another limitation of Serenity-S1 lies in its minimal local memory resources. Because the DAQ and Timing Hub (DTH) and the detector front end store the bulk of the data in the CMS detector, the board is optimised for streaming at high throughput and lacks capacity for intermediate storage. Adding DDR memory would broaden its use cases, enabling features such as online histogramming, long-latency trigger buffering, and more complex algorithm deployment.

Thus, upgrading both the FPGA and the memory subsystem represent a natural evolution, aligning Serenity with the computational demands expected for experiments beyond CMS Phase-2.

Future platform options While the ATCA standard provided a suitable foundation for Serenity-S1, its adoption has declined in recent years. Components such as backplane connectors and crates are increasingly difficult to procure, and reconstructing the entire infrastructure – including crates, shelf managers, and backplanes – would be impractical. For a potential Serenity-S2, alternative form factors must therefore be considered.

Besides retaining the ATCA standard, three principal options emerge for a successor platform:

- **PCIe form factor**: Implementing Serenity as a PCIe accelerator would enable direct integration into commodity servers, but it would impose hard constraints on board area, front-panel I/O density, and cooling. Achieving $O(100)$ optical links, deterministic timing distribution, and several hundred watts of dissipation within a PCIe envelope would likely require external I/O breakouts and specialised chassis infrastructure, offsetting much of the operational simplicity that motivates PCIe in the first place.
- **VPX standard**: As the successor of VME and widely used in aerospace and defence, VPX offers long-term availability and backplane-based high-speed serial connectivity. Its limitations are twofold: lane rates are restricted to $25\,\mathrm{Gbit\,s^{-1}}$, and the form factor is smaller than ATCA, constraining the system complexity. With ambitions to increase transceiver bandwidth and processing capacity, VPX may therefore impose unacceptable limits.
- **Server-class system**: A more radical option is to abandon the Field Replaceable Unit (FRU)-based paradigm entirely and adopt a 1U server-style system. Projects such as the Open Compute Project or ETH Zürich's Enzian [229] provide reference architectures for high-availability servers. Such a design would permit larger board areas and greater flexibility while maintaining the Serenity principle of a single, powerful processing unit. In this scenario, the Kria could continue to serve as the board management controller.

Retaining ATCA would allow re-use of the existing ecosystem and the Serenity-framework, but it also imposes significant restrictions:

- Limited power dissipation ($400\,\mathrm{W}$)
- Limited cooling performance through forced airflow
- Limited PCB height ($2.4\,\mathrm{mm}$)
- Limited component height ($1.6\,\mathrm{mm}$ on bottom, $21.3\,\mathrm{mm}$ on top [70])
- Limited backplane speed ($12\,\mathrm{Gbit\,s^{-1}}$ [230])

Adding significant memory will necessitate a thicker PCB, and power consumption of next-generation devices is already approaching 600 W (e.g. the AMD VPK180 evaluation board [231]). These constraints would limit the functionality of a Serenity-S2. Hence, a server-style – colloquially called “pizzabox” – system currently appears the most promising option.

System-level consequences A transition away from ATCA would primarily matter in the context of this dissertation because it would change the boundary conditions under which the Modular-Hardware Toolbox is instantiated. The Toolbox is form-factor independent, but the current Serenity framework encodes ATCA-specific assumptions such as shelf-managed power-state negotiation, backplane-centric connectivity, and FRU-style maintenance workflows. Moving to a self-contained node architecture would therefore require a corresponding framework adaptation that preserves the same modular partitioning, interface specifications, and release/verification practices, while mapping them onto server-style operational conventions.

From a hardware-design perspective, the main architectural pressure points are power delivery, cooling, and interconnect density. Next-generation devices and the addition of local memory tend to increase dissipation and routing complexity, which in turn drive PCB layer count and mechanical constraints. These trends are compatible with the Toolbox approach, but they reduce the design margin available within ATCA’s mechanical envelope and its ecosystem constraints.

Consequently, the key methodological requirement for a successor platform is not a specific form factor, but a reference framework that maintains the Toolbox invariants: stable module boundaries, controlled interface evolution, and reproducible design artefacts across generations. A more detailed discussion of candidate form factors and their implications is provided in Appendix A.3.

5.5 Summary

Serenity-S1 is the primary processing platform of the CMS Phase-2 back-end and the first large-scale validation vehicle for the Modular-Hardware Toolbox. The chapter motivated the board architecture from subsystem requirements and described the resulting separation into a reusable service area and a payload optimised for high-density optical I/O and deterministic timing. Measurements on successive prototypes and pre-production boards verified power integrity, high-speed link margin, and production-scale testability via automated acceptance test procedures. Collectively, these results establish Serenity-S1 as a production-ready platform for multi-subdetector deployment and as a concrete reference implementation of the Modular-Hardware Toolbox workflows.

6 Discussion and Impact

The development of the Serenity-S1 ATCA card offered a unique opportunity to evaluate the Modular-Hardware Toolbox under realistic, production-relevant constraints. The Toolbox was developed in response to the methodological gaps identified in Chapter 3, and Serenity-S1 provided a concrete case study to evaluate these concepts in practice. The project combined complex system requirements, multi-institute collaboration, and an extended prototyping cycle shaped by supply-chain fluctuations. These conditions make Serenity-S1 a representative case for understanding both the strengths and limitations of structured, modular design methodologies. This chapter synthesises these findings and discusses the broader implications for hardware development within and beyond high-energy physics. A key inflection point during the transition from prototyping to a production-ready design was the migration from Mentor PADS to Altium Designer. This change enabled the systematic application of hierarchical design structures, rule-based checks, and reusable artefact generators central to evaluating the Modular-Hardware Toolbox under realistic constraints.

6.1 Validation Against Methodological Requirements

This section evaluates the Serenity-S1 development against the methodological requirements derived in chapter 3. Rather than restating the Toolbox architecture, the focus is on how these requirements manifested in practice during the design, integration, and revision of the board.

6.1.1 Consistency (R1)

The Serenity-S1 development provides substantial evidence that standardisation mechanisms – including project templates, naming conventions, validated layer stacks, and deterministic release bundles – contribute to more consistent design outcomes. The migration of the Serenity-S1 design to a unified Altium-based structure enforced a single hierarchy of schematic documents, a defined directory layout, and shared rule sets for ERC and DRC. These measures ensured that

contributions from different institutes remained mutually compatible and could be integrated without extensive adaptation.

Consistent naming conventions were introduced to maintain uniform signal and power-net identifiers across the hierarchy. The resulting general-to-specific naming structure reinforced the abstraction model adopted by the Modular-Hardware Toolbox and reduced ambiguity at subsystem boundaries. To support interoperability, frequently used signal groups such as I^2C and JTAG were formalised as reusable interface harnesses, providing stable integration points across modules. A complete overview of these interface definitions is provided in Table A.2.

The introduction of deterministic release bundles – while common in modern engineering workflows – further improved reproducibility across the S1.1–S1.4 revisions. The generator framework, combined with Git-based version control, ensured that BOMs, pick-and-place files, fabrication data, and schematics remained synchronised across contributors, reducing manual corrections during manufacturing preparation.

Taken together, these measures demonstrate that the consistency requirement can be achieved in a realistic, multi-institute hardware development. Importantly, the achieved level of consistency was not merely beneficial but functioned as an enabling condition for effective multi-institute collaboration: shared conventions, uniform interfaces, and deterministic artefacts allowed subsystem development to proceed in parallel without substantial integration overhead.

6.1.2 Reuse and Modularity (R2)

The Serenity-S1 development resulted in a substantial set of reusable power, clocking, and slow-control modules, several of which were instantiated multiple times within the same design. Examples include three instances of the LTM4676A power module, two LTM4700 core-supply modules, two LTM4680 regulators for high-current MGT supplies, and six ZL30274 jitter-cleaners for timing distribution. This internal repetition demonstrates one important form of reuse: a module validated once can be deployed consistently in multiple locations with predictable behaviour and without re-engineering effort. Beyond Serenity-S1, several of these modules have been reused in unrelated PCB projects, where reuse provides the greatest benefit by avoiding repeated schematic translation and layout effort.

A core expectation of the Modular-Hardware Toolbox is that modularisation reduces duplicated design effort and lowers integration risk. The Serenity-S1 experience supports this expectation, particularly for power-conversion modules and several slow-control functions, where module reuse accelerated development. Across revisions S1.1–S1.3, several device substitutions were carried out with limited design disruption. Notably, the TPSM8D6B24 regulator was replaced by an LTM4680, and the LTM4657 supply was supplemented by an LTM4703 to reduce noise on

VCCO. These substitutions were implemented without modifying the global system architecture, and only minor adjustments were required at the module interface. Power-conversion modules demonstrated high interchangeability, owing to their low port density and predominantly planar power interfaces.

Serenity-S1 also clarifies a practical distinction between *logical* and *physical* modularity. For complex, high-pin-count functions, schematic-level compatibility can often be preserved through stable interface abstractions (e.g. consistent net grouping, harness contracts, and port naming), allowing alternative devices to remain functionally interchangeable in the logical design. In layout, however, package-specific pin locations, escape routing, and impedance constraints dominate, so late substitutions typically require extensive re-routing and may violate placement or SI constraints. The case study therefore supports R2 where modularity is technically effective, while making explicit that portability can diverge between schematic and layout levels.

This limitation does not undermine the concept of modularity; instead, it describes the conditions under which modular reuse is effective. Modules with predominantly planar interfaces – most notably power-conversion stages – exhibit high reuse potential and can be substituted with comparatively little integration effort. In contrast, modules whose external connectivity is governed by SI constraints or mechanically fixed pin topologies must be defined early in the design process, since any later substitution demands substantial re-routing and may compromise performance. The Serenity-S1 case study therefore validates requirement R2 within the domain where modularity is technically feasible and, at the same time, shows the practical boundaries.

6.1.3 Automation and Error Reduction (R3)

The Serenity-S1 development provides clear evidence that automation improves design reliability and reduces manual overhead. Several established industry practices were applied systematically, including IPC-7351-compliant footprint generation [232], rule-based ERC/DRC enforcement, and automated generation of fabrication and assembly artefacts. Although these tools are well established in commercial workflows, their disciplined use within a distributed academic design environment provided tangible benefits. Generated footprints matched manufacturer specifications without requiring post-hoc correction, and the fabrication generators enforced consistent directory structures and naming conventions across all revisions. This standardisation mitigated a form of error recurrence that had affected earlier prototype boards, where inconsistent artefact formats complicated manufacturing preparation.

Automated rule checks further strengthened error prevention during schematic entry and layout. ERC and DRC served as a common enforcement layer across the distributed design team, ensuring that changes introduced in one subsystem could not silently violate design conventions elsewhere.

In a multi-institute environment with asynchronous contributions, this automated validation reduced the integration burden and increased the transparency of design modifications.

Despite these gains, automation remained incomplete in areas directly relevant to the Modular-Hardware Toolbox. In particular, extracting reusable modules from existing Altium projects proved labour-intensive when the original design had not explicitly declared modules as such. The process required manual removal of design-specific constraints, restructuring of annotations, and reconstruction of interface contracts. At present, no automated tooling exists within Altium Designer to detect module boundaries, convert design fragments into reusable entities, or validate module integrity according to Toolbox rules. As a result, the scalability of module lifecycle management remains limited, and retrofitting modularity into established projects is still a bottleneck.

In summary, Serenity-S1 demonstrates that requirement R3 is partially fulfilled: automation was highly effective for artefact generation, footprint creation, and rule-based verification, but remains insufficient for module extraction and reuse workflows. Enhancing automation in these areas would significantly improve the realisation of the full methodological benefits envisioned by the Modular-Hardware Toolbox.

6.1.4 Traceability (R4)

Traceability requirements motivated the introduction of a unified, version-controlled design environment that encompassed not only project documentation and source files but also the component libraries used throughout the Serenity-S1 development. Within the local development workflow, this structure provided a clear historical record of schematic and layout modifications, while structured naming conventions ensured reliable correlation between module variants, regulator configurations, and design revisions. The combination of repository-based version control and well-defined identifiers enabled contributors to follow the evolution of design artefacts with substantially greater transparency than in earlier projects.

The distributed nature of Serenity-S1 development, however, revealed important limitations of the library infrastructure. In practice, three separate library sources were used: the CERN Altium library (read-only for external collaborators), a local project-specific library, and an Altium 365-based KIT library. This fragmentation introduced inconsistencies that could not be detected automatically. A notable example occurred when a designer adapted a footprint from the CERN library locally; because the CERN library was read-only, this modification did not propagate to other collaborators. The resulting mismatch contributed to a placement error in the first revision of Serenity-S1, illustrating the fragility of traceability when library infrastructures are not fully shared.

This experience highlights that R4 is technically achievable but highly dependent on organisational and infrastructural alignment. While the methodological provisions of the Modular-Hardware Toolbox support systematic traceability, their effectiveness is limited in environments where contributors cannot access or modify a common, full-featured component library. A unified, shared library with consistent access rights would have prevented the divergence observed in Serenity-S1 and represents a necessary precondition for robust traceability in future multi-institute developments.

6.1.5 Scalability and Collaboration (R5)

Serenity-S1 provides a representative evaluation of requirement R5 due to its multi-institute development structure, extended design cycle, and substantial subsystem complexity. The modular partitioning strategy enabled different teams to work in parallel on the power-entry subsystem, Kria control path, FPGA support infrastructure, clock tree, FireFly integration, and IPMC logic. This distributed workflow was feasible because the preceding requirements – consistent project structures (R1), well-defined reusable modules (R2), and reliable traceability mechanisms (R4) – ensured that subsystem boundaries were unambiguous and that design artefacts remained coherent across contributors.

The routing phase presented a particularly stringent test of collaborative scalability. As Altium Designer does not support concurrent editing of a single PCB document, contributors adopted a shift-based workflow in which layout ownership rotated between institutes. This approach enabled continuous progress despite the tool limitation, but it also highlighted the fragility of collaboration when EDA infrastructures do not support true multi-user operation. In this context, modular design played a supportive role: although concurrent editing of the same layout region was not possible, the clear separation of functional regions reduced the likelihood of conflicting modifications and made hand-offs between teams more predictable. Together with the earlier migration to Altium for the production design, this illustrates how tool capabilities and tool constraints jointly shape the collaboration model that a modular methodology must accommodate.

Overall, the Serenity-S1 development demonstrates that the modular methodology supports effective distributed collaboration even under restrictive tool conditions. At the same time, the experience shows that achieving higher scalability will require improvements in tool-agnostic collaboration infrastructures, particularly in shared library access and multi-user layout capabilities. The project therefore substantiates R5 in practice while also indicating the infrastructural enhancements necessary for broader, large-scale adoption.

6.2 Mutual Influence Between Toolbox Development and Serenity-S1

The relationship between the Modular-Hardware Toolbox and the Serenity-S1 development was intrinsically bidirectional. Rather than applying a predefined methodology to an isolated design task, both the Toolbox and the Serenity-S1 hardware evolved in parallel. The processing card served as the first large-scale environment in which emerging methodological concepts were exercised under production-relevant constraints, while the Toolbox provided increasing architectural structure that shaped subsequent design decisions. This section analyses this co-evolution, distinguishing (i) design practices that originated during Serenity-S1 and were later formalised in the Toolbox, (ii) methodological shortcomings revealed by Serenity-S1 that necessitated refinements of the Toolbox, and (iii) reinforcement loops through which practical constraints iteratively sharpened methodological rules.

6.2.1 Emergent Toolbox Concepts Identified During Serenity-S1

Several principles that would later become foundational elements of the Modular-Hardware Toolbox first emerged organically during Serenity-S1. The most influential of these was the structural separation between the service area and the payload area. Initially introduced to accommodate potential variants of the Serenity family, this partition provided a clear architectural boundary that guided schematic hierarchy, defined subsystem responsibilities, and ultimately informed the architectural framework codified in Chapter 4. Subsequent refinements during Serenity-S1 – including the treatment of Zone 3 connectivity and front-panel infrastructure – revealed the need for a more precise definition of framework templates, which the Toolbox later systematised.

Early reusable modules developed during Serenity-S1 likewise served as prototypes for the module-encapsulation principles formalised in the Toolbox. Power-supply blocks and the ZL30274 timing module were the first subsystem fragments designed with repeatability in mind. Their development demonstrated the need for explicit interface contracts, clear parametrisation, and encapsulated design intent. For example, an early issue in the first Serenity-S1 revision – a missing pull-up resistor on the *STATUS* pin of an LTM4676A regulator – illustrated that modules must include all auxiliary components required for correct operation, even if certain pins are intended to be connected outside the module. This incident reinforced the principle that reusable modules must never rely implicitly on project-level circuitry.

Related methodological ideas also originated during Serenity-S1. Naming conventions were introduced to ensure consistent logical grouping and general-to-specific hierarchy (e.g. I2C-FireFly-North, 3V3-MGTAVTT-South). Likewise, the use of Altium harnesses to abstract protocol-level interfaces provided an early manifestation of what later became the Toolbox's interface-contract philosophy. These practices, although informally established at the outset, formed the conceptual basis for formal methodological constructs introduced in Chapter 4.

6.2.2 Methodological Gaps Revealed by Serenity-S1

While Serenity-S1 benefited from emerging Toolbox principles, the design programme also exposed methodological and infrastructural limitations that shaped subsequent Toolbox refinements. Several of these issues arose from toolchain constraints. Altium Designer restricts reusable modules combining schematic and layout to the Altium 365 cloud infrastructure, which in turn limited effective module sharing across institutes. Modules could only be exchanged by embedding them in example projects, making collaborative review and distributed development impractical. This was one of the limitations that triggered a conceptual shift in the Toolbox – from a repository of ready-to-use modules to a methodology emphasising the construction and validation of modules across heterogeneous EDA tools.

The collaboration likewise exposed weaknesses in library governance. Maintaining three parallel library sources – the CERN read-only library, a local project-specific library, and a KIT-specific Altium 365 workspace – introduced inconsistencies. A prominent example was the mismatch in FireFly connector footprints described before. The resulting mechanical misalignment in the first prototype highlighted the necessity of centralised library management, strict metadata governance, and structured review templates. These requirements were later incorporated explicitly into the Toolbox.

Serenity-S1 also revealed the difficulty of retroactively extracting reusable modules from a mature design. This observation reinforced a fundamental methodological principle: modularity must be designed in the earliest architectural stages, and reusable artefacts must be constructed with abstraction and portability in mind. The Toolbox adopted this insight through its emphasis on early encapsulation.

6.2.3 Reinforcement Loops Between Method and System

Several iterative refinement cycles demonstrate how Serenity-S1 and the Toolbox shaped one another. Initial module parametrisation strategies, although aligned with good design practice, conflicted with Altium's handling of power nets, which restricted the ability to propagate

parametrised net names. This prompted a methodological shift toward explicit port-based connectivity for power interfaces, later codified as a design rule in the Toolbox.

Early Serenity-S1 layout work experimented with irregular module geometries that aligned with local placement constraints but hindered reuse and spatial integration across the board. These experiences motivated a transition to rectangular module boundaries, which now form part of the Toolbox's layout guidelines.

Similarly, divergences in passive-component selection across modules led to unnecessary BOM proliferation and procurement overhead. This observation fed back into the Toolbox through harmonised passive-selection policies and batch-replacement strategies based on industry-standard footprints.

These reinforcement loops illustrate that key methodological rules did not originate from abstract reasoning alone but from the practical constraints and lessons encountered during Serenity-S1. They clarify which aspects of modularity, parametrisation, interface definition, and library governance scale reliably in a collaborative HEP electronics environment, and which depend on external infrastructure beyond the designer's control.

6.3 Process Capability Assessment

This section examines the development workflow through a capability-maturity lens. The analysis uses the CMMI-inspired evaluation scheme introduced in section 3.4 to determine whether the structured, modular approach adopted in Serenity-S1 translated into observable improvements in engineering practice relative to earlier Phase-2 projects.

6.3.1 Methodological Limitations of the CMMI Assessment

The capability assessment provides the primary quantitative indication of methodological improvement, but several limitations constrain its interpretability. First, the three-level scoring scale captures broad behavioural differences but cannot resolve incremental improvements within each level. Serenity-S1 therefore exhibits qualitative advances that are not fully reflected numerically. Second, only a single project was developed under the full influence of the Modular-Hardware Toolbox, while the baseline comprises few heterogeneous designs; broader statistical validation requires more Toolbox-based projects and, consequently, more time than available in this thesis. Third, several categories inherently rely on expert judgement, particularly documentation discipline and review practice, introducing a degree of subjectivity despite the use of structured artefacts

and version control. Finally, the capability levels are not independent from each other. Improvements in one category often enable or constrain progress in others; for example, schematic-entry discipline affects review quality, which in turn influences verification maturity and documentation reliability. Such interdependencies mean that capability scores can reinforce one another or mask weaknesses, limiting the extent to which individual categories can be interpreted in isolation.

Despite these limitations, the assessment provides a directionally reliable indication of process improvements. The observed capability gains are consistent with the methodological enhancements introduced by the Modular-Hardware Toolbox and reinforce the findings from section 6.1 and section 6.2. As a result, the capability model should be understood as a coarse but meaningful indicator of process evolution rather than a set of strictly separable metrics. The capability results should therefore be seen as a conservative but still substantive quantification of the Toolbox's impact.

6.3.2 Quantitative Summary of the Capability Assessment

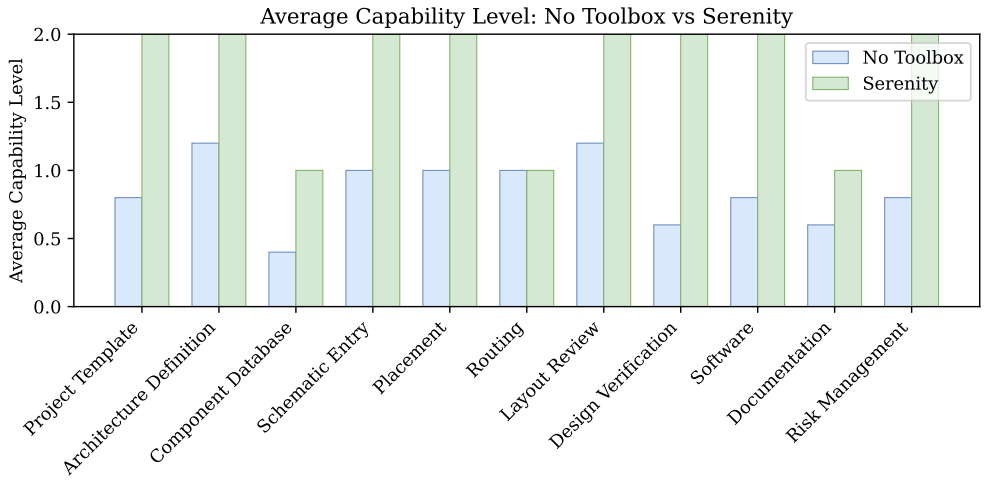


Figure 6.1: CMMI-based capability levels for Serenity-S1 compared with baseline projects.

The capability model evaluates thirteen process categories on a three-level scale (0–2), covering project infrastructure, component-database use, schematic discipline, placement and routing practice, verification, documentation, and cross-team integration. Figure 6.1 summarises the results for Serenity-S1 compared with the baseline established in section 3.4. The baseline exhibits mean capability levels around 1.2, whereas Serenity-S1 reaches an average of 1.7, with

improvements visible in most categories. Detailed values are shown in the figure; the following discussion focuses on interpreting these differences.

6.3.3 Interpretation of Capability Strengths and Limitations

The Serenity-S1 development demonstrates markedly higher process capability than the baseline projects, achieving the "Managed" level in all categories except component-database management, routing, and documentation. The following interpretation discusses the principal strengths and remaining limitations in the context of the Modular-Hardware Toolbox.

Structured project initiation and architectural definition Serenity-S1 attained the highest capability level in project-template usage and architectural definition. The systematic use of interface contracts, hierarchical composition, and the service/payload partition produced a significantly more structured initiation phase than observed in the baseline projects. These conventions supported alignment across institutes and provided a stable foundation for later integration. The architectural review involved a broad collaboration, including firmware and software developers, ensuring early compatibility with subsystem requirements. These practices correspond to Level 2 in the project-template, architectural definition, and software categories of the capability model. By contrast, component-library governance remained limited by the parallel use of three independent libraries (CERN, project-local, and Altium 365). This fragmentation increased maintenance effort and contributed to a footprint mismatch in the first Serenity-S1 revision. While a single, fully shared library would have avoided this divergence, such an arrangement is not always feasible in multi-institute collaborations. In practice, traceability can be preserved through strict version pinning of reference libraries, or by confining project-specific components to the version-controlled project repository.

Schematic-entry discipline Schematic-entry capability reached Level 2 through consistent application of naming conventions, hierarchical sheet structuring, and systematic use of Electrical Rules Check (ERC). Version-controlled review artefacts – such as annotated schematics and deterministic release bundles – further strengthened design discipline. These practices represent a substantial improvement over the baseline projects, where schematic conventions and design-rule enforcement were applied more heterogeneously.

Placement and layout Module placement was planned and reviewed prior to routing, leading to Level 2 in the placement category. Experience from predecessor boards informed spatial allocation for power planes and high-speed channels, reducing the risk of later congestion. Routing

capability, however, remained at Level 1. Despite systematic Design Rules Check (DRC) use, DC-drop analysis, and partial PI simulations, the absence of a comprehensive SI-verification workflow prevented the board from achieving a fully managed routing process. Given the density and variety of high-speed nets, toolchain limitations rather than methodological deficiencies constituted the primary constraint.

Review and verification Serenity-S1 exhibited more systematic review and verification behaviour than earlier designs. Review tasks were assigned to experts not directly involved in subsystem development, providing independent assessment and early defect detection. Module-level analyses – including power-stage simulation, clock-distribution evaluation, and early control-plane firmware bring-up – contributed to a predictable integration and commissioning phase. Although verification was not fully institutionalised, the use of structured checklists and automated tests (see subsection 5.3.3) significantly improved process maturity.

Documentation and risk management Documentation improved relative to prior projects but remained fragmented across CERNBox directories, private records, and local repositories, limiting maturity to Level 1. Nevertheless, all significant issues identified during prototyping and commissioning were systematically recorded and fed into subsequent revisions. The design was also shaped by supply-chain volatility during the semiconductor crisis, which motivated explicit mitigation strategies for component obsolescence and availability risks. These measures strengthened risk management, which achieved capability level 2.

6.4 Broader Implications and Future Opportunities

The preceding sections demonstrated that the Modular-Hardware Toolbox improves design consistency, supports structured collaboration, and enhances process capability in the context of the Serenity-S1 development. At the same time, the analysis revealed several methodological and toolchain-related constraints that limit the realisation of the Toolbox's full potential. Building on these findings, this section examines the broader implications for future workflow design and discusses opportunities for applying, extending, or operationalising the Toolbox beyond the Serenity-S1 case study.

6.4.1 Methodological Implications for Future Workflow Design

The Serenity-S1 experience demonstrates that modular and interface-driven workflows can be applied effectively in large, distributed hardware projects. Clear subsystem boundaries, deterministic artefact generation, and reusable design elements enabled a more structured and predictable development process than observed in the baseline projects. These improvements do not only enhance reproducibility and integration efficiency but also increase the organisational headroom for managing more complex designs, as anticipated in section 3.3. In this sense, the Toolbox provides a methodological foundation that scales with the growing architectural and organisational complexity of modern scientific instrumentation.

Beyond internal design considerations, the supply-chain constraints encountered during Serenity-S1 further highlighted the value of module-level abstraction: the ability to instantiate critical blocks as stand-alone evaluation platforms reduced integration risk when late component substitutions became unavoidable.

At the same time, several limitations identified in section 6.1 and section 6.3 motivate further refinement of the methodology. Although the Toolbox itself is conceptually independent of any specific EDA environment, the design artefacts it produces inevitably rely on proprietary tool formats. Given the limited interoperability between commercial EDA tools, this constraint cannot be fully eliminated. Nevertheless, abstraction layers based on netlists, metadata, and rule descriptions may allow partial cross-tool validation and automated consistency checks, even when full migration of schematic–layout modules is not feasible. Developing automation workflows that operate on tool-agnostic artefacts wherever possible would therefore increase portability and robustness.

A further implication concerns empirical validation. Broader adoption of the Toolbox across additional projects and institutes will both enlarge the evidence base for quantifying reuse effectiveness, error reduction, and long-term maintainability, and provide new insights that may refine the methodology itself. As the Toolbox is applied to a wider range of architectures and system scales, its processes, templates, and module-governance rules will evolve correspondingly. Continued use is therefore not only a means of evaluation but an integral driver of the Toolbox's future evolution.

6.4.2 Applicability Beyond ATCA and High-Energy Physics

Although developed for CMS Phase-2 back-end electronics, the underlying principles of the Modular-Hardware Toolbox extend naturally to other scientific instrumentation domains. Laboratory control systems, detector readout boards, cryogenic electronics, embedded monitoring devices, and data acquisition platforms share common requirements for reproducible workflows, structured project initiation, and validated component libraries. Even in systems where full modularity is constrained by analogue performance, mechanical geometry, or thermal design, partial adoption of Toolbox conventions – such as naming rules, hierarchical project templates, automated artefact generation, and CI-integrated rule checks – offers immediate benefit.

Furthermore, the modular decomposition of hardware into reusable elements facilitates collaboration in distributed research environments. As scientific instrumentation increasingly incorporates heterogeneous processing, high-bandwidth interconnects, and complex control infrastructure, parallel development supported by clear interface contracts will become essential for maintaining project timelines and integration coherence. The Serenity-S1 case study provides concrete evidence that such practices can be deployed successfully at scale.

6.4.3 Abstraction Boundaries in Highly Integrated Processing Devices

The Modular-Hardware Toolbox is based on the assumption that reusable design artefacts can be defined at stable abstraction boundaries, separating processing, conversion, power delivery, clocking, and interfacing functions at PCB level. This assumption holds for the majority of today's high-performance DAQ architectures and underpins the module-selection criteria introduced in Section 4.3.1. However, a small class of highly integrated devices challenges this separation by collapsing multiple functional domains into a single component.

A boundary case is provided by Radio-Frequency System-on-Chip (RFSoC) devices, which integrate programmable logic, embedded processing, and high-performance ADCs/DACs blocks within a single package. From a PCB-design perspective, this collapses the conventional boundary between processing and conversion: analogue performance, clocking, power sequencing, and converter configuration become inseparably coupled to the main device. In such architectures, fine-grained reuse at the level of discrete converter stages becomes less meaningful, and modularity shifts toward framework-level templates that encapsulate the integrated device's power, clock, configuration, and thermal constraints while exposing stable system interfaces. RFSoCs therefore do not contradict the modular approach; they primarily require selecting a coarser granularity that matches the integration density.

6.4.4 Relevance for Industrial Design Workflows

The methodological concepts formalised through the Toolbox align closely with trends observed in industrial electronics development, where design reuse, verification, and automation are central priorities. Component vendors routinely distribute reference schematics and evaluation boards, but these artefacts seldom take the form of reusable schematic–layout modules. The Toolbox demonstrates how such modules – enriched with metadata, integration assumptions, and parameter ranges – could provide a more rigorous foundation for cross-project reuse.

Organisations lacking mature internal reuse frameworks could adopt the Modular-Hardware Toolbox as a tool-independent conceptual starting point for building consistent design libraries and validation workflows. As industrial teams become increasingly distributed, either by outsourcing or mobile working, methodologies that support traceability, design governance, and automated validation will become more valuable.

Modular design also opens opportunities for new forms of hardware intellectual property. Verified, versioned hardware modules could be supplied as reusable IP, analogous to digital IP cores or firmware components. Semiconductor manufacturers could distribute pre-validated modules for common subsystems such as power regulation, clocking, and high-speed interfaces, reducing integration risk for customers and shortening development cycles. Challenges such as liability, certification, and BOM consistency would require governance mechanisms, but the potential for an ecosystem of reusable hardware design blocks is clear.

6.4.5 Long-Term Outlook for Modular Hardware Methodologies

As scientific and industrial electronics evolve toward heterogeneous architectures – combining FPGAs, CPUs, AI accelerators, and high-bandwidth interconnects – modular methodologies will become increasingly important. Framework templates may evolve to encode not only electrical and mechanical constraints but also protocol abstractions, firmware interfaces, and system-level deployment assumptions. Version-controlled component databases and reproducible generation pipelines will play a central role in maintaining long-term maintainability and interoperability across technology generations.

The Serenity-S1 development demonstrates that such structured workflows can be deployed effectively in a realistic, multi-year engineering environment. Although no direct comparison of development duration is available, the reduced need for late redesigns and the ability to validate subsystems independently suggest that modular workflows can improve development efficiency by lowering integration risk rather than by accelerating individual design steps. While

the methodology is not yet fully mature, it establishes a foundation on which future modular-hardware practices can build. The broader implications therefore extend beyond the immediate context of CMS Phase-2 and point toward a future in which reusable, validated hardware modules and structured design processes form the backbone of complex scientific and industrial electronics.

7 Conclusion and Outlook

Modern high-energy physics experiments, such as the Compact Muon Solenoid (CMS) detector at the High-Luminosity Large Hadron Collider (HL-LHC), rely on complex Data Acquisition (DAQ) systems that must process aggregate data rates of $2.7\,\mathrm{Pbit\,s^{-1}}$, operate reliably over decades, and evolve under changing physics and technology constraints. Designing the underlying DAQ electronics is not only a technical challenge but also a methodological one: projects involve many institutes, extended design cycles, and increasing complexity driven by rising detector bandwidths and the growing functional density of modern Field-Programmable Gate Arrays (FPGAs). Against this backdrop, this dissertation examined how modular design principles can improve development methodologies for such systems. Two complementary contributions were presented: the *Modular-Hardware Toolbox*, which provides a structured and reusable methodology for electronic design, and the *Serenity-S1* processing card, which demonstrates this methodology within the CMS Phase-2 upgrade project. Together, these contributions illustrate how modularity, reproducibility, and structured workflows can support hardware development under long design cycles, distributed collaborations, and evolving performance demands.

A requirements analysis formed the conceptual basis of the Modular-Hardware Toolbox. It synthesised recurring technical and organisational challenges – including heterogeneous project structures, fragmented component libraries, limited documentation baselines, and insufficient reuse of validated artefacts – into five overarching requirements for modern design workflows: consistency (R1), reuse and modularity (R2), automation and error reduction (R3), traceability (R4), and scalability and collaboration (R5). A Capability Maturity Model Integration (CMMI)-based assessment revealed systematic weaknesses in verification practices and configuration management, reinforcing the need for methodological improvement. Broader particle-physics community studies, such as the ECFA Detector R&D Roadmap and Snowmass reports, further emphasise the growing importance of heterogeneous processing, embedded Artificial Intelligence (AI)/Machine Learning (ML), and flexible, high-bandwidth back-end architectures, underscoring the relevance of modular and interface-driven workflows.

The Modular-Hardware Toolbox translates these requirements into a coherent methodological framework composed of six subsystems: project infrastructure, reusable design modules, framework templates, automated generators, a component database, and verification processes. Each

subsystem addresses specific challenges identified in the requirements analysis, collectively supporting consistency, reducing duplicated manual work, and enabling systematic reuse across the hardware lifecycle. Because fully controlled process comparisons are not feasible for PCB development, the evaluation relies on structured, semi-quantitative evidence. A CMMI-based capability comparison between the Serenity-S1 development and a representative baseline indicates stronger configuration management, more disciplined verification workflows, and improved reproducibility of design artefacts. In addition, the library of reusable modules established during the Serenity-S1 project demonstrates practical uptake of the Toolbox and provides a foundation for future designs across institutes.

The Serenity-S1 processing card is a central hardware platform of the CMS Phase-2 back-end electronics. It was designed to provide up to $3.1\,\mathrm{Tbit\,s^{-1}}$ of optical input bandwidth, deterministic low-latency processing, and robust board management within the constraints of the ATCA form factor. Its architecture, combining a high-bandwidth VU13P FPGA with precise timing distribution and an SoC-based management subsystem, reflects the stringent performance and reliability requirements of the CMS back-end at the HL-LHC. Extensive electrical, thermal, and high-speed link validation, together with a production-scale factory acceptance test (FAT) procedure, confirmed compliance with the power-integrity, signal-integrity, and timing constraints of the Phase-2 environment.

With more than 700 boards planned for production and approximately 50 prototype units already deployed across multiple CMS subsystems for firmware development and system integration, Serenity-S1 is the most widely deployed back-end processing card of the CMS Phase-2 upgrade. At this scale, it forms a cornerstone of the CMS Phase-2 back-end architecture and serves as the primary processing platform for large parts of the upgraded data acquisition chain. The project therefore represents both a major engineering outcome of this doctoral work and a long-term operational component of the CMS Phase-2 upgrade. A central contribution was the coordination and technical leadership of the Serenity-S1 hardware design process across its full development lifecycle, with a strong emphasis on modular design practices. This included authoring and reviewing the majority of the schematics and defining reusable hardware modules, contributing directly to the placement and local routing of these modules at board level, and establishing stable interfaces that enabled subsequent global routing to be carried out in a distributed manner. Early stages of production and commissioning were carried out hands-on, while subsequent validation and testing were structured and continued through supervised student and staff contributions.

Beyond its technical role, Serenity-S1 also shaped the methodological contribution of this dissertation. The scale, complexity, and distributed nature of its development created the conditions under which the need for consistent project structures, reusable design elements, systematic verification artefacts, and robust configuration management became evident. These experiences directly

informed the formulation of requirements R1–R5 and guided the architecture of the Modular-Hardware Toolbox. Conversely, as the Toolbox matured, its structured templates, automated artefact generation, and reusable modules provided tangible support during Serenity-S1 design iterations and verification workflows. The board therefore not only benefited from methodological improvements but also served as the primary real-world environment in which these improvements were tested, validated, and refined.

Synthesising these results, two overarching contributions emerge. The Modular-Hardware Toolbox establishes a requirement-driven methodology for reproducible and maintainable hardware design, while the Serenity-S1 project demonstrates its feasibility and practical value in a large-scale, production-relevant DAQ system.

Several limitations point to opportunities for future work. Although conceptually tool-agnostic, the current implementation depends on specific EDA environments, limiting portability. Broader validation across additional boards, both within and beyond high-energy physics, would expand the module library and provide a more diverse empirical basis for evaluating process improvements. As design cycles repeat, quantitative indicators such as module reuse rates or reductions in duplicated artefacts could complement the capability-based assessment used here and enable more statistically grounded conclusions.

In conclusion, this dissertation demonstrates that modular design principles, long established in software and systems engineering, can be effectively transferred to the hardware development of high-performance DAQ electronics. The Modular-Hardware Toolbox provides a practical and structured methodology that strengthens process capability and supports reproducible, maintainable design workflows. Serenity-S1 exemplifies this methodological approach in a particularly demanding experimental context, providing evidence that structured modular practices can improve both development quality and long-term system integration. Future DAQ architectures will increasingly incorporate heterogeneous processing resources, AI-driven algorithms, and flexible back-end topologies. In this context, methodologies centred on modularity, reusability, and reproducibility will become essential. The framework developed here offers a foundation on which future generations of modular hardware design can be built.

A Appendix

A.1 Practical Usage of the Modular-Hardware Toolbox

This appendix complements Chapter 4 by documenting practical conventions, workflows, and implementation details of the Modular Hardware Toolbox. While the main chapter focuses on architectural rationale and methodological assessment, the following sections provide operational guidance intended for direct application in Altium projects.

A.1.1 Project-Level Infrastructure

This section documents the project-level infrastructure as implemented in the reference deployment of the Modular-Hardware Toolbox at IPE. The described infrastructure represents a complete, production-used instantiation of the toolbox architecture introduced in Chapter 4. It is intentionally specific and serves as empirical evidence that the toolbox concept can be applied to complex, multi-board hardware developments under realistic constraints.

Project Directory Structure All hardware projects at IPE that follow the Modular-Hardware Toolbox methodology are initialised from a common project template. The template fixes the logical structure of design artefacts and reduces structural freedom in favour of consistency and traceability across projects.

The project template contains the following assets:

Project file The project file stores global parameters that are reused across schematics and layout, including project number, revision identifier, and licence information. These parameters are referenced automatically in title blocks and PCB graphics, eliminating manual duplication and reducing the risk of inconsistencies between documents.

Top-level schematic file A dedicated schematic sheet serves as the hierarchical root of the design. All functional schematics are instantiated below this level. The sheet includes

three fiducials on the top side and three on the bottom side, as well as a standardised graphical element displaying project number and revision. This ensures visual and semantic consistency across all designs.

Layout file The PCB layout document is preconfigured with a validated eight-layer stack suitable for PCB pooling services commonly used at IPE. Fiducials and project graphics are already placed to ensure consistency between schematic and layout-level artefacts from the first design iteration onward.

Bill of materials A bill-of-materials document is included and configured for automatic updates based on schematic content. The BOM is treated as a generated artefact and is only manually edited to allow for production enumeration of parts.

Assembly drawing The assembly drawing is preconfigured to display populated top and bottom views of the PCB using a fixed visual style. It serves as an artefact for assembly review and manufacturing hand-off and is regenerated as part of the release process.

Harness definition file Recurring electrical interfaces such as I2C, SPI, clock signals, and high-speed serial links are defined in a central harness file. These harness definitions enforce consistent signal naming, directionality, and grouping across all modules and projects and are referenced throughout the Serenity-S1 design.

Output generator configuration A generator configuration is included to produce schematic PDFs, manufacturing data, and assembly files in a repeatable manner. All generated outputs are tagged with project revision and date, ensuring coherent release bundles and traceable manufacturing artefacts.

Project Templates and Initialisation New projects are created by selecting the project template and adapting a small, predefined set of metadata parameters (described above). At minimum, this includes assignment of a project identifier, revision state, and short textual description.

From the first commit onward, each project is maintained under version control. Design sources and generated artefacts are clearly separated to ensure that all outputs can be regenerated deterministically at any later revision. This workflow has been applied throughout the development of Serenity-S1 and related boards and proved robust under iterative design updates and parallel collaboration.

Layer Stacks and Design-Rule Packs The reference implementation employs a restricted set of validated layer stacks and associated design-rule packs to ensure manufacturability and layout reuse across projects.

For low- to medium-complexity designs, PCB pooling services are used with the following MultiCB layer stacks [233]:

- 4L-01: Four-layer stack with up to two impedance-controlled layers
- 6L-01: Six-layer stack with up to three impedance-controlled layers
- 8L-01: Eight-layer stack with up to four impedance-controlled layers

For high-complexity designs, project-approved layer stacks are employed. In particular, the Serenity-S1 board uses a twenty-layer stack implemented as two sequential laminations, providing four impedance-controlled signal layers and four dedicated power layers. This stack was validated through prototyping and used for large-scale production.

To further improve coherence across reusable layout modules, a reduced via library is enforced. Limiting the number of via geometries reduces manufacturing complexity and avoids incompatible assumptions between independently developed modules. The recommended via sizes used in the reference implementation are summarised in Table A.1.

Table A.1: Recommended via sizes for reusable layout design.

Via	Pad Size	Finished Hole Size	Recommended Use
STD400	400 µm	200 µm	For very dense areas (No IPC Class 3)
STD500	500 µm	250 µm	For standard signal routing
STD750	750 µm	350 µm	For power nets (use multiple)
STD1000	1000 µm	500 µm	For very high currents (use multiple)

Documentation and Review Conventions Each project repository contains a `README` file that serves as the primary entry point for collaborators and reviewers. The file includes a brief project description, a high-level block diagram or interface overview, and references to key schematic sheets or modules.

Design reviews are supported through consistent schematic formatting, embedded design notes, and clearly defined hierarchical boundaries. While no formal review workflow is mandated at template level, the provided infrastructure ensures that all artefacts required for structured reviews and manufacturing hand-off are available in a uniform form.

A.1.2 Reusable Design Modules

This section documents how reusable schematic and layout modules were realised in the reference implementation of the Modular-Hardware Toolbox during the Serenity-S1 project. The focus is on practical conventions, interface definitions, and constraints that enabled repeated use of validated circuitry under production-relevant conditions.

Modules were created selectively where repeated occurrence, non-trivial integration effort, or hidden design constraints justified encapsulation. The intent is not to provide an exhaustive module catalogue, but to demonstrate how modularisation was applied systematically and at scale.

Module File Structure and Naming Each reusable module is maintained as a self-contained design unit consisting of a schematic sheet, an optional layout template, and – if necessary – associated documentation. Module identifiers follow the primary component. Versioning is handled through the Altium 365 cloud repository rather than embedded into the module name.

Interface Definition Contracts and Signal Naming Module interfaces are defined explicitly through named ports and standardised harnesses. In addition to defining signal groups and electrical roles, the interface contract also imposes minimal but strict rules on signal naming to ensure unambiguous interpretation across schematics, layout, and documentation.

Signal names are required to encode function rather than physical implementation. Naming rules are intentionally conservative and limited to aspects that affect module interoperability.

- **Hierarchical signal naming**: Signal and harness names refine meaning from left to right, progressing from generic interface class to increasingly specific context. Names follow the pattern `<Interface>[_<Subsystem>][_<Instance>]`, for example `I2C_FIREFLY_North` or `UART_KRIA0`. Once assigned, names are treated as stable identifiers and are not modified at module boundaries until reaching its highest hierarchical level.
- **Directionality**: Signal direction is encoded semantically rather than by schematic orientation. For example, `TX` and `RX` refer to the perspective of the module that owns the interface. When connecting to each other (e.g. UART), an obvious RX-TX crossover should be placed.
- **Polarity and inversion**: Active-low, single ended signals are suffixed with `n` (e.g. `RESETn`, `ALERTn`). No other polarity encoding is used.

- **Differential pairs**: Differential signals are named using `_P` and `_N` suffixes and should always be instantiated within a harness.
- **Buses and vectors**: Multi-bit signals are represented using bus notation (e.g. `DATA[3..0]`). Bit ordering follows descending significance unless explicitly documented otherwise.
- **Harness ownership**: Signals that are part of a harness must not be renamed at module boundaries. Adapters, if required, are implemented explicitly at system level.

These conventions enable deterministic signal grouping, simplify automated checks, and reduce ambiguity during integration and review. They have been used consistently across all modules in the Serenity-S1 reference implementation.

All harnesses used in the reference implementation are derived from a central definition file and are treated as locked interfaces. Individual modules may instantiate one or more harnesses but must not modify their internal signal composition.

Table A.2 summarises the harnesses used in the Serenity-S1 implementation.

Table A.2: Standardised harness definitions used in the Modular-Hardware Toolbox reference implementation.

Harness	Signals	Typical Use
PMBUS	SCL, SDA, ALERTn	Power-module configuration and telemetry
I2C	SCL, SDA, INTn, RSTn	Low-speed control and monitoring
JTAG	TCK, TDI, TDO, TMS	Device configuration and debug access
UART	TX, RX	Console and diagnostic interfaces
DIFFCLK	CLK_P, CLK_N	Differential reference clocks
DIFF	D_P, D_N	Generic differential signalling
GTX	TX_P, TX_N, RX_P, RX_N	High-speed serial transceiver links
ETH	P0_P/N – P3_P/N, LED0/1	Ethernet PHY serial interface
ETH_RGMII	TX*, RX*	RGMII Ethernet MAC–PHY interface
MDIO	MDC, MDIO, RSTn	Ethernet PHY management interface
USB2	VBUS, DP, DN	USB 2.0 device interface
SDIO	DAT[3..0], CMD, CLK, CD	SD card interfaces
FireFly_x4	4× GTX	Quad-lane FireFly optical modules
FireFly_x12	12× GTX	Twelve-lane FireFly optical modules

The use of interface definition contracts serves two purposes: it reduces interface ambiguity during integration, and it allows modules implementing similar functions to be interchanged without renaming or regrouping signals at system level.

Parametrisation and Allowed Configuration Space Modules are designed to support controlled parametrisation, primarily through resistive networks, configuration pins, or optional population variants. Parametrisation is limited to dimensions that do not alter the fundamental topology or violate validated operating conditions.

Each parametrised element must document:

- its intended adjustment range,
- relevant electrical constraints (ratings, stability margins),
- dependencies on external circuitry or sequencing.

Configurations that fall outside the validated parameter space are intentionally not supported, even if technically feasible, in order to preserve module robustness and reuse reliability.

Module Documentation Requirements Every reusable module includes embedded documentation on the schematic sheet. This documentation captures design intent, integration assumptions, and known limitations that are not directly visible from connectivity alone.

At minimum, module documentation includes:

- a brief functional description,
- explicit interface assumptions,
- references to data sheets or application notes.

This information is required to prevent hidden dependencies and to enable safe reuse by designers not involved in the original module development.

Module Inventory in the Serenity-S1 Reference Implementation Table A.3 summarises the reusable modules developed during the Serenity-S1 project. The table illustrates the breadth of module types and the degree of reuse within the design. It is not intended as a complete catalogue of the Modular-Hardware Toolbox.

A.1.3 Framework Templates

Framework templates extend the concept of reusable design modules to subsystems whose functional scope, interconnect density, or standard-defined constraints cannot be captured adequately

Table A.3: Reusable modules developed during the Serenity-S1 project.

Module/Device	Uses in Serenity-S1	Primary Function
ADS1115	0	Slow auxiliary ADC for power monitoring
LMK61E2	3	Local programmable reference clock
ZL30274	6	Low-jitter clock distribution for MGTs
88E1512	1	Ethernet PHY
LTM4676A	3	Dual-output PMBus-controlled SMPS
LTM4680	2	Dual-output PMBus-controlled SMPS
LTM4700	2	Single-output PMBus-controlled SMPS (bridged)
LTM4703	1	Low-noise SMPS for sensitive rails
LTM8074	1	Low-power SMPS
ADP7156	11	Low-noise LDO (multiple fixed voltages)
TCA9548A	3	8x I^2C bus expansion
PCAL6408A	14	8-bit GPIO expansion via I^2C

by self-contained modules. In the reference implementation of the Modular-Hardware Toolbox, frameworks are used deliberately and sparingly.

Two framework templates were derived directly from the Serenity-S1 design: a device-centric framework for the AMD Virtex UltraScale+ VU13P FPGA and a form-factor-centric framework for the ATCA service area. Both represent validated architectural extractions from a production-grade board.

Device-Centric Framework: VU13P FPGA The device-centric framework captures the complete integration context of the Virtex UltraScale+ VU13P FPGA as used on Serenity-S1. It instantiates all infrastructure required for reliable operation of the device and defines the boundary between fixed integration logic and project-specific payload connections.

The framework includes the following elements (see Figure 4.2):

- the VU13P FPGA device itself,
- all FPGA power supplies, comprising two LTM4700 modules for the core supply, two LTM4680 modules for the MGT power domains, and one LTM4703 module for the I/O,
- an I^2C I/O expander (PCAL6408A) used for FPGA configuration and control signals,

- a JTAG interface with automatic plug detection, allowing the interface to be switched between an external debug header and the board-management system,
- five ZL30274 jitter-attenuating PLLs for transceiver clock generation, including their associated low-noise regulation (ten ADP7156 LDOs) and a dedicated LTM4676A supply generating the intermediate voltage for the LDO stages,
- one LMK61E2 programmable local oscillator, and
- an I^2C multiplexer (TCA9548A) used to segment and manage the clock-control buses.

Rather than abstracting this functionality behind a large module interface, the framework instantiates the complete device environment directly. This reflects the fact that FPGA integration imposes a substantial fixed design overhead that is largely independent of the application logic implemented in the fabric. Application-specific transceiver usage, payload interfaces, and logic connectivity are realised outside the framework boundary. If any of the power supplies or clock distribution ICs need replacing, this is possible, as they are all modularised.

Form-Factor Framework: ATCA Service Area The form-factor framework captures the complete service area of the Serenity-S1 board and encodes the mechanical, electrical, and management requirements of the ATCA standard. It provides a compliant baseline from which payload-specific designs can be developed without re-implementing standard-defined infrastructure.

The ATCA framework includes (see Figure 4.3):

- the ATCA mechanical form factor, including backplane connectors for Zone 1 and Zone 2, ESD stripe, and handle switch with blue LED indication,
- the PIM and DC/DC modules, both implemented as quarter-brick converters,
- the board-management subsystem, including support for OpenIPMC and compatible implementations (e.g. CERN IPMC or Zynq-based IPMC), connected via a SoDIMM connector,
- a Kria K26 System-on-Module used for board management and control-plane functionality,
- an Ethernet switch (KSZ9897R) and a dedicated Ethernet PHY (88E1512) for network connectivity of the Kria,
- power supplies for the management network and control logic, including an LTM4691 supplying the Ethernet switch from the 3.3 V standby rail and an LTM4644 supplying the Kria from the 12 V DC/DC output.

All elements required for ATCA compliance and board management are fixed by the framework. The payload area is intentionally excluded and left unstructured, allowing project-specific processing and interface logic to be developed independently of the service infrastructure.

Customising Framework Templates Framework templates follow the same fundamental design principles as reusable modules with respect to documentation, clarity, and traceability. However, they are intentionally treated as architectural scaffolding rather than immutable artefacts.

Designers are free to adapt, extend, or remove elements of a framework as required by the target application, provided that the underlying architectural intent and standard compliance are preserved. Unlike reusable modules, frameworks are not expected to remain identical across projects.

A.1.4 Generator Framework

The generator framework in the Modular-Hardware Toolbox reference implementation addresses repetitive and error-prone tasks in footprint creation and manufacturing data generation. Its scope is intentionally limited to artefacts where automation provides clear reliability and consistency benefits. No attempt was made to introduce comprehensive documentation or report generators beyond these areas.

Footprint Generation All regularly shaped component footprints used in Serenity-S1 were created using the IPC-compliant footprint wizard provided by Altium Designer [234]. Using IPC-derived footprints ensured consistent pad sizing, courtyard definition, and assembly clearances across the design and reduced manual interpretation of component data sheets. For complex or mechanically constrained packages, the wizard output served as a starting point that was subsequently refined to meet device-specific or layout-specific requirements. Fully manual footprint creation was avoided wherever possible.

Fabrication and Assembly Outputs Fabrication and assembly artefacts were generated using Altium Designer's *OutJob* mechanism. A predefined output configuration was included in each project to ensure that manufacturing data could be regenerated deterministically at any revision.

The generated outputs include:

- Gerber X2 files,

- ODB++ manufacturing data,
- pick-and-place files for top and bottom side (exported as Excel files),
- bill of materials (exported as Excel file),
- assembly drawings (PDF),
- schematic documentation (PDF).

The structure and naming of these artefacts are defined at project-template level and were described in Section A.1.1. All generated files are tagged with project revision and generation date to ensure coherent release bundles and traceable manufacturing hand-offs.

Documentation and Report Generation No general-purpose documentation or design-report generators were implemented beyond the manufacturing-critical artefacts described above. In particular, automated extraction of documentation from schematic and layout data is currently not performed.

In contrast, automated board commissioning already exists in the reference implementation. A set of scripted tests (FAT, see subsection 5.3.3) is used during bring-up to exercise design-specific interfaces and to record measurement results in a designated test report for each board. This extends determinism beyond design-file generation and provides a traceable link between released hardware revisions and the associated commissioning outcomes.

A plausible future extension of the generator framework would be the automated derivation of connectivity mappings directly from the design netlist, for example mapping FireFly lanes to FPGA MGT channels and mapping PLL outputs to FPGA clock inputs. However, implementing this reliably requires additional structured device metadata (e.g. a machine-readable representation of the FPGA clocking and transceiver resources) that is not currently available in the required form.

Overall, the generator framework combines deterministic manufacturing output generation with automated board-level commissioning reports, while leaving documentation extraction out of scope. This reflects a pragmatic prioritisation of automation where it is already well supported, and measurably reduces integration and production risk.

A.1.5 Component Database

This section documents the evolution, structure, and intended usage of the centralised component database within the Modular-Hardware Toolbox. The database addresses limitations observed

during the Serenity-S1 development and provides a foundation for consistent, tool-agnostic component management in future projects.

Motivation and Context As discussed in Section 6.3.3, the split library approach used during the Serenity-S1 development introduced unnecessary complexity and limited consistency across schematic symbols, footprints, and metadata. While functional, this approach proved suboptimal for long-term reuse and collaborative development.

At the same time, the transition of IPE to Altium Designer necessitated the establishment of a new component library infrastructure. In parallel, a Helmholtz-wide initiative was launched to develop a shared, institute-spanning component database, motivated by the same requirements of consistency, traceability, and reuse identified in this dissertation.

Component Entry Structure Each component in the database is uniquely identified by its manufacturer part number, which serves as the primary key across all supported EDA environments. A component entry comprises, at minimum:

- schematic symbol definition,
- footprint definition,
- manufacturer and ordering information,
- essential electrical and packaging parameters,
- references to data sheets and qualification notes.

This structure establishes a single source of truth for component-related information and prevents divergence between project-local libraries.

Library Generation for EDA Tools A central design goal of the component database is tool interoperability. To this end, the database is structured such that symbols and footprints can be used from both Altium Designer and KiCad simultaneously.

This cross-compatibility elevates the toolbox architecture beyond tool-specific workflows and preserves portability even as institutes adopt different EDA environments. A proof-of-concept implementation of this approach exists and has demonstrated successful round-trip use between Altium and KiCad. Full deployment of the shared library is planned for early 2026.

Version Control and Change Management The component database is intended to be maintained in a Git-based version control environment. All changes to symbols, footprints, or metadata are tracked, reviewable, and attributable to specific revisions.

This approach enables structured review processes, supports traceability of design decisions, and allows controlled evolution of the library without breaking existing designs. It also aligns component management with the same version-controlled workflows used for schematics, layouts, and generated artefacts throughout the Modular-Hardware Toolbox.

A.1.6 Verification and Release Infrastructure

This section documents the verification and release mechanisms applied in the Modular-Hardware Toolbox reference implementation to ensure electrical correctness, manufacturability, and traceability prior to fabrication. The infrastructure combines automated rule checks with structured human review and deterministic release artefact generation.

ERC and DRC Rule Sets Both electrical rule checks (ERC) and design rule checks (DRC) are performed as mandatory steps before design release. Together, they provide complementary coverage of logical correctness at schematic level and physical correctness at layout level.

The ERC is applied during schematic entry and prior to schematic freeze. It is used to detect logical and connectivity-related issues such as:

- conflicting signal directions (e.g. multiple outputs driving a net),
- unconnected or floating signal inputs,
- unconnected or floating power inputs,
- inconsistent pin-type usage across hierarchical boundaries.

ERC violations are resolved either by correcting the schematic or, where the behaviour is intentional, by explicitly waiving or documenting the exception. Undocumented ERC violations are not permitted at release stage. This practice ensures that remaining warnings represent conscious design decisions rather than oversights.

The DRC is applied to the PCB layout and enforces a comprehensive set of physical and electrical constraints derived from manufacturing requirements, applicable standards, and signal integrity considerations. In the Serenity-S1 reference implementation, the DRC covers the following classes of rules:

- **Manufacturing constraints**, including minimum feature sizes, minimum drill diameters, minimum drill-to-drill and drill-to-copper spacing, and minimum distances to the board edge.
- **Form-factor-specific constraints**, such as maximum allowable component heights on top and bottom side for ATCA boards and keep-out regions required for rail-guided insertion and extraction.
- **Signal-specific constraints**, including controlled impedances (e.g. $100\,\Omega$ differential, $50\,\Omega$ single-ended), length matching within differential pairs, and skew constraints within multi-lane buses.
- **Electrical safety constraints**, such as creepage and clearance distances for high-voltage nets.

By combining these rule classes into a unified DRC configuration, layout-level errors are detected early and consistently across projects.

Review Checklists Automated checks are complemented by structured review checklists applied during schematic review, layout review, and release preparation. These checklists focus on aspects that are difficult to verify automatically, such as architectural consistency, integration assumptions between modules, and compliance with project-specific requirements.

Release Bundles and Artefact Integrity Design release is performed by generating deterministic release bundles that contain all artefacts required for manufacturing, assembly, and documentation. Release bundles include the outputs described in Section A.1.1 and are generated using predefined output configurations.

A.2 FPGA and SoC Devices

This appendix provides an exhaustive overview of FPGA and SoC devices that have been deployed in IPE projects from 2019 to 2025. The table complements the discussion in subsection 4.3.1, where a limited number of processing devices is recommended to simlify firmware and software development and to benefit from economies of scale. Table A.4 shows, that there is already some consolidation happening (e.g. AMD Zynq UltraScale+ XCZU19EG).

Table A.4: Overview of FPGA devices used across different projects [59, 235, 236, 237].

Device	CLB LUTs	MGTs	Special	Used in
Spartan	–	–	–	–
Artix	–	–	–	–
Kintex UltraScale XCKU115	663 k	64 GTH[1]	–	QSolid
Kintex UltraScale+ XCKU3P	163 k	16 GTY[2]	–	TRISTAN Remote ADC
Virtex UltraScale+ XCVU9P	1.18 M	120 GTY[2]	–	TRISTAN
Virtex UltraScale+ XCVU13P	1.73 M	128 GTY[2]	–	Serenity-S1
Zynq UltraScale+ XCK26	117 k	4 GTR[1], 4 GTH[1]	PS	Kria K26 SoM for Serenity-S1
Zynq UltraScale+ XCZU4EG	88 k	16 GTH[1]	PS	ZynqMP Mezzanine FMC+
Zynq UltraScale+ XCZU19EG	523 k	32 GTH[1], 16 GTY[2]	PS	HiFlex2; DTS-100G; PANDA AMC
Zynq UltraScale+ RFSoC XCZU49DR	425 k	16 GTY[2]	PS, ADC, DAC	ZCU216 for CryoDAQ and QSolid
Versal AI Edge XCVE2802	521 k	32 GTYP[2]	PS, AI/ML	VEK280 for KINGFISHER
Versal AI Core XCVC1902	900 k	44 GTY[2]	PS, AI/ML	VCK190 for Evaluation for CryoDAQ

[1] GTR, GTH: MGTs with up to 16.3 Gbit s^{-1}
[2] GTY, GTYP: MGTs with up to 32.75 Gbit s^{-1}

A.3 Candidate Platform Options for a Potential Serenity-S2

As a forward-looking illustration, a potential successor to Serenity-S1 (denoted Serenity-S2, see Figure A.1) could adopt a server-class form factor rather than ATCA. This consideration is primarily driven by scaling trends observed during the Serenity-S1 development: next-generation processing devices and the addition of local memory resources are expected to push power dissipation beyond the 400 W envelope of ATCA shelves, while simultaneously increasing routing density and required PCB layer count. Comparable systems employing first-generation Versal devices, such as the ATLAS Global Common Module, already require layer counts of the order of 26 layers [120], complicating PCB construction within ATCA thickness constraints.

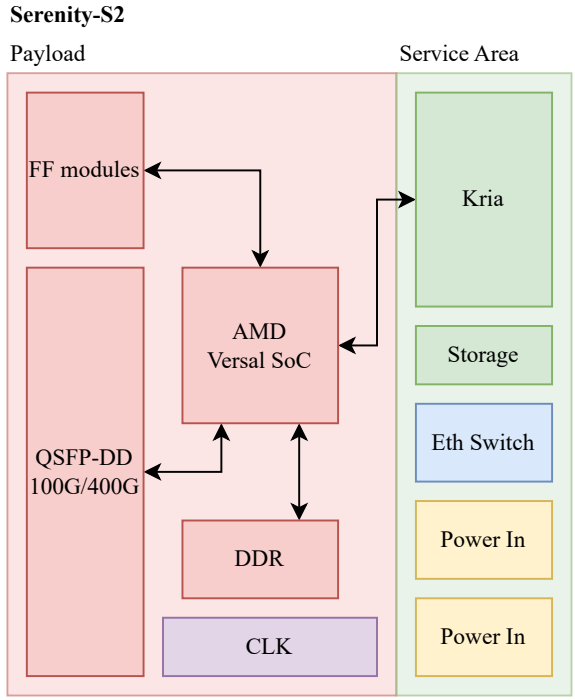


Figure A.1: High-level block diagram of the Serenity-S2 architecture.

A server-style architecture would relax several of these limitations by removing constraints on total board area, component height, and cooling topology. In particular, the use of direct, component-level water-cooling – commonly employed in high-performance server systems – would offer significantly improved thermal margins compared to crate-level forced airflow, albeit at the cost of increased system complexity. Such approaches are compatible with existing water-cooled infrastructure in CMS computing areas and would permit sustained operation of devices with power consumptions far exceeding 400 W (e.g. the AMD VPK180 [231]).

From a design-architecture perspective, the availability of additional board area and relaxed mechanical constraints would also enable the integration of substantial local memory and more flexible signal partitioning. Retaining a two-sub-stack PCB architecture while separating high-speed transceiver routing from memory and slow-control circuitry would improve signal integrity at increased data rates. Furthermore, abandoning the ATCA backplane and adopting more server-grade high-speed optical interfaces such as QSFP-DD [238] or OSFP [239], would increase the bandwidth capabilities of the system.

Crucially, this Serenity-S2 candidate is not proposed as a concrete hardware design. Its relevance within this dissertation lies in illustrating how the Modular-Hardware Toolbox could be

re-instantiated under different system-level boundary conditions. While the Toolbox itself is form-factor independent, the current Serenity framework encodes ATCA-specific assumptions, including shelf-managed power negotiation, FRU-based lifecycle management, and backplane-centric connectivity. A successor platform would therefore require a new reference framework that preserves the same modular partitioning, interface control, and release workflows, while mapping them onto a self-contained server-node architecture. In this sense, Serenity-S2 serves as an example of how the methodological principles validated with Serenity-S1 could be maintained beyond the ATCA ecosystem.

B Acronyms

ADC	Analog-to-Digital Converter
AI	Artificial Intelligence
ALLEGRO	A Lepton-Lepton collider Experiment with Granular Read-Out
APD	Avalanche Photodiode
ASIC	Application-Specific Integrated Circuit
ATCA	Advanced Telecommunications Computing Architecture
ATLAS	A Toroidal LHC ApparatuS
BE	Back-End
BOM	Bill of Materials
BRAM	Block Random-Access Memory
BRIL	Beam Radiation, Instrumentation, and Luminosity
CBM	Compressed Baryonic Matter
CDR	Clock-Data Recovery
CEPC	Circular Electron Positron Collider
CL	Capability Level
CLB	Configurable Logic Block
CLD	Compact LInear Collider (CLIC)-Like Detector
CMMI	Capability Maturity Model Integration
CMS	Compact Muon Solenoid
COTS	Commercial Off-The-Shelf
CPU	Central Processing Unit
CSC	Cathode Strip Chamber
CSP	CMS Standard Protocol
D2S	Data-to-Surface
DAC	Digital-to-Analog Converter
DAQ	Data Acquisition
DRC	Design Rules Check
DRD	Detector Research and Development
DSP	Digital Signal Processing
DT	Drift Tube
DTC	DAQ and Timing Card

DTH	DAQ and Timing Hub
ECAL	Electromagnetic Calorimeter
ECFA	European Committee for Future Accelerators
EDA	Electronic Design Automation
ERC	Electrical Rules Check
FCC	Future Circular Collider
FE	Front-End
FPGA	Field-Programmable Gate Array
FRU	Field Replaceable Unit
GEM	Gas Electron Multiplier
gFEX	global Feature EXtractor
GPU	Graphics Processing Unit
HBM	High-Bandwidth Memory
HCAL	Hadronic Calorimeter
HDL	Hardware Description Language
HEP	High-Energy Physics
HGCAL	High-Granularity Calorimeter
HL-LHC	High-Luminosity Large Hadron Collider
HLT	High-Level Trigger
HPC	High Performance Computing
IDEA	Innovative Detector for Electron-positron Accelerator
ILC	International Linear Collider
ILD	International Large Detector
IPMC	intelligent platform management controller
L1T	Level-1 Trigger
LEP	Large Electron-Positron Collider
LHC	Large Hadron Collider
LHCb	Large Hadron Collider beauty
lpGBT	Low-power Gigabit Transceiver
LUT	Look-Up Table
MGT	Multi-Gigabit Transceivers
ML	Machine Learning
MTD	Minimum Ionizing Particle Timing Detector
NoC	Network-on-Chip
OT	Outer Tracker
PCB	Printed Circuit Board
PI	Power Integrity
PLL	Phase-Locked Loop
PS	Processing System

QSFP	Quad Small Form Factor Pluggable
RAM	Random-Access Memory
RPC	Resistive Plate Chamber
RTM	Rear Transition Module
SI	Signal Integrity
SiPM	Silicon Photomultiplier
SoC	System-on-Chip
SoM	System-on-Module
TCDS2	Phase-2 Timing and Control Distribution System
VLSI	Very-Large-Scale Integration
µTCA	Micro Telecommunications Computing Architecture

C List of Figures

D List of Tables

E List of Publications

[1] T. Mehner et al. “ZynqMP-based board-management mezzanines for Serenity ATCA-blades”. In: *JINST* 17.03 (2022), p. C03009. DOI: 10.1088/1748-0221/17/03/C03009. arXiv: 2404.03447 [physics.ins-det].

[2] T. Mehner et al. “Lessons learned while developing the Serenity-S1 ATCA card”. In: *JINST* 19.02 (Feb. 2024), p. C02018. DOI: 10.1088/1748-0221/19/02/C02018.

[3] T. Mehner et al. “Serenity-S1 – A Versatile ATCA Processing Card for the CMS Phase-2 Upgrade”. In: (Oct. 2025). DOI: 10.36227/techrxiv.176162104.42530921/v1. URL: https://doi.org/10.36227/techrxiv.176162104.42530921/v1.

[4] M. Fuchs et al. “Split Boot—True Network-Based Booting on Heterogeneous MPSoCs”. In: *IEEE Trans. Nucl. Sci.* 70.6 (2023), pp. 1157–1163. DOI: 10.1109/TNS.2023.3237968. arXiv: 2301.05642 [physics.ins-det].

[5] W. Adam et al. “Evaluation of 3D pixel silicon sensors for the CMS Phase-2 Inner Tracker”. In: (Oct. 2025). arXiv: 2510.13304 [physics.ins-det].

[6] A. Hayrapetyan et al. “Operation and performance of the CMS silicon strip tracker with proton-proton collisions at the CERN LHC”. In: *JINST* 20.08 (2025), P08027. DOI: 10.1088/1748-0221/20/08/P08027. arXiv: 2506.17195 [physics.ins-det].

[7] W. Adam et al. “Measurement of the fractional radiation length of a pixel module for the CMS Phase-2 upgrade via the multiple scattering of positrons”. In: *JINST* 19.10 (2024), P10023. DOI: 10.1088/1748-0221/19/10/P10023. arXiv: 2407.13721 [physics.ins-det].

[8] W. Adam et al. “Beam test performance studies of CMS Phase-2 Outer Tracker module prototypes”. In: *JINST* 19.10 (2024), P10032. DOI: 10.1088/1748-0221/19/10/P10032. arXiv: 2404.08794 [physics.ins-det].

[9] W. Adam et al. “Evaluation of planar silicon pixel sensors with the RD53A readout chip for the Phase-2 Upgrade of the CMS Inner Tracker”. In: *JINST* 18.11 (2023), P11015. DOI: 10.1088/1748-0221/18/11/P11015. arXiv: 2307.01580 [physics.ins-det].

[10] W. Adam et al. “Evaluation of HPK n^+-p planar pixel sensors for the CMS Phase-2 upgrade”. In: *Nucl. Instrum. Meth. A* 1053 (2023), p. 168326. DOI: 10.1016/j.nima.2023.168326. arXiv: 2212.04793 [physics.ins-det].

[11] W. Adam et al. “Beam test performance of a prototype module with Short Strip ASICs for the CMS HL-LHC tracker upgrade”. In: *JINST* 17.06 (2022), P06039. DOI: 10.1088/1748-0221/17/06/P06039.

[12] W. Adam et al. “Test beam performance of a CBC3-based mini-module for the Phase-2 CMS Outer Tracker before and after neutron irradiation”. In: *JINST* 18.04 (2023), P04001. DOI: 10.1088/1748-0221/18/04/P04001. arXiv: 2205.00961 [physics.ins-det].

[13] W. Adam et al. “Selection of the silicon sensor thickness for the Phase-2 upgrade of the CMS Outer Tracker”. In: *JINST* 16.11 (2021), P11028. DOI: 10.1088/1748-0221/16/11/P11028.

[14] W. Adam et al. “Comparative evaluation of analogue front-end designs for the CMS Inner Tracker at the High Luminosity LHC”. In: *JINST* 16.12 (2021), P12014. DOI: 10.1088/1748-0221/16/12/P12014. arXiv: 2105.00070 [physics.ins-det].

F Bibliography

[15] G. Apollinari et al., eds. *High-Luminosity Large Hadron Collider (HL-LHC)*. CERN-2015-005. CERN, 2015. URL: https://cds.cern.ch/record/2116337.

[16] T. R. F. P. Tomei. *CMS upgrades for the High-Luminosity LHC era*. 2025. arXiv: 2501.03412 [hep-ex]. URL: https://arxiv.org/abs/2501.03412.

[17] CMS Offline Software and Computing. *CMS Phase-2 Computing Model: Update Document*. Tech. rep. Geneva: CERN, 2022. URL: https://cds.cern.ch/record/2815292.

[18] CMS Collaboration. *Technical proposal for the Phase-2 upgrade of the CMS detector*. Tech. rep. CERN-LHCC-2015-010, LHCC-P-008, CMS-TDR-15-02. CERN, 2015. DOI: 10.17181/CERN.VU8I.D59J. URL: https://cds.cern.ch/record/2020886.

[19] ATLAS Collaboration. *Highlights of the HL-LHC physics projections by ATLAS and CMS*. Tech. rep. Geneva: CERN, 2025. URL: https://cds.cern.ch/record/2928907.

[20] CMS Collaboration. *The Phase-2 upgrade of the CMS tracker*. Tech. rep. CERN-LHCC-2017-009, CMS-TDR-014. CERN, 2017. DOI: 10.17181/CERN.QZ28.FLHW. URL: https://cds.cern.ch/record/2272264.

[21] CMS Collaboration. *The Phase-2 upgrade of the CMS endcap calorimeter*. Tech. rep. CERN-LHCC-2017-023, CMS-TDR-019. CERN, 2017. URL: https://cds.cern.ch/record/2293646.

[22] CMS Collaboration. *The Phase-2 upgrade of the CMS muon detectors*. Tech. rep. CERN-LHCC-2017-012, CMS-TDR-016. CERN, 2017. URL: https://cds.cern.ch/record/2283189.

[23] CMS Collaboration. *A MIP timing detector for the CMS Phase-2 upgrade*. Tech. rep. CERN-LHCC-2019-003, CMS-TDR-020. CERN, 2019. URL: https://cds.cern.ch/record/2667167.

[24] C. Foudas et al. "The CMS tracker readout front-end driver". In: *IEEE Trans. Nucl. Sci.* 52.6 (2005), pp. 2836–2840. DOI: 10.1109/TNS.2005.860173.

[25] G. Haefeli et al. "TELL 1". In: *CERN Report* (2003). DOI: 10.5170/CERN-2003-006.129. URL: https://cds.cern.ch/record/712059.

[26] I. Bestintzanos, K. Adamidis, I. Euaggelou, et al. “An ATCA processor for Level-1 trigger primitive generation and readout of the CMS barrel muon detectors”. In: *JINST* 18.02 (Feb. 2023), p. C02039. DOI: 10.1088/1748-0221/18/02/C02039. URL: https://dx.doi.org/10.1088/1748-0221/18/02/C02039.

[27] P. Das, S. Dasu, P. Kumar, et al. “The APx board for the CMS Phase 2 L1 calorimeter trigger: Testing and performance”. In: *Proc. TWEPP 2024 Workshop*. Accessed: 2025-09-05. Oct. 2024. URL: https://indico.cern.ch/event/1381495/contributions/5988643/attachments/2938295/5161929/Ojalvo_TWEPP_2024.pdf.

[28] S. Goadhouse and N. Loukas. *The CMS barrel calorimeter processor (BCP V2) evaluation*. Oral presentation at TWEPP 2025, Rethymno, Crete. Oct. 2025. URL: https://indico.cern.ch/event/1502285/contributions/6554518/attachments/3149350/5593210/BCPV2_Board_Eval_TWEPP2025_Oct2025_sdg.pdf.

[29] A. Akpinar, A. Blaizot, S. Cholak, et al. “Design, construction, and testing of the APOLLO ATCA blades for use at the HL-LHC”. In: *JINST* 20.04 (Apr. 2025), p. C04001. DOI: 10.1088/1748-0221/20/04/C04001. URL: https://dx.doi.org/10.1088/1748-0221/20/04/C04001.

[30] A. Greshilov, D. Acosta, M. Bachtis, et al. “X2O ATCA IPMC and control solution”. In: *CMS Indico Workshop*. Presentation given on 24 May 2023. May 2023. URL: https://indico.cern.ch/event/1277467/contributions/5418239/attachments/2652900/4593740/2023-05-24_X2O_ATCA_IPMC_and_control_solution_AG.pdf.

[31] A. Paramonov. “FELIX: The detector interface for the ATLAS experiment at CERN”. In: *EPJ Web Conf.* 251 (2021), p. 04006. DOI: 10.1051/epjconf/202125104006. URL: https://doi.org/10.1051/epjconf/202125104006.

[32] J. Osterhout. *A philosophy of software design*. 2nd ed. First edition published 2018. Yaknyam Press, 2021. ISBN: 978-1732102217.

[33] R. C. Martin. “Design principles and design patterns”. In: *Object Mentor* 1.34 (2000), p. 597.

[34] A. Hunt and D. Thomas. *The pragmatic programmer: From journeyman to master*. Harlow, England: Addison-Wesley, 1999. ISBN: 978-0-201-61622-4.

[35] C. Szyperski, D. Gruntz, and S. Murer. *Component software: Beyond object-oriented programming*. ACM Press, 2002. ISBN: 9780201745726.

[36] Y. Calleecharan. “Data acquisition and signal processing for smart sensors”. In: *Int. J. Adapt. Control Signal Process.* 22.8 (2008), pp. 811–812. DOI: 10.1002/acs.997. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/acs.997.

[37] M. Di Paolo Emilio. *Embedded systems design for high-speed data acquisition and control*. 1st ed. eBook published 1 September 2014. Cham, Switzerland: Springer, 2015. ISBN: 978-3-319-06864-0. DOI: 10.1007/978-3-319-06865-7. URL: https://doi.org/10.1007/978-3-319-06865-7.

[38] J. D. González-Martínez. "H2GCROC: Design and performance of a dedicated very front-end ASIC for SiPM readout of the CMS high granularity calorimeter". In: *Nucl. Instrum. Meth. A* 1047 (2023), p. 167863. DOI: 10.1016/j.nima.2022.167863. URL: https://www.sciencedirect.com/science/article/pii/S016890022201155X.

[39] W. R. Leo. *Techniques for nuclear and particle physics experiments: A how-to approach*. 2nd ed. Berlin, Heidelberg: Springer, 1994. ISBN: 978-3-540-57280-0. DOI: 10.1007/978-3-642-57920-2. URL: https://doi.org/10.1007/978-3-642-57920-2.

[40] K. Uchida et al. *The CBC3 readout ASIC for CMS 2S-modules*. Tech. rep. Geneva: CERN, 2018. URL: https://cds.cern.ch/record/2312215.

[41] N. Ellis. *Trigger and data acquisition*. Lectures at the 2005 CERN–CLAF School of High-Energy Physics. pp. 241–266. Cali, Colombia: CERN, Mar. 2006. DOI: 10.5170/CERN-2006-015.241. URL: https://cds.cern.ch/record/1017829/files/p241.pdf.

[42] CMS Collaboration. *The Phase-2 upgrade of the CMS Level-1 trigger*. Tech. rep. Final version. Geneva: CERN, 2020. URL: https://cds.cern.ch/record/2714892.

[43] CMS Collaboration. *The Phase-2 upgrade of the CMS data acquisition and high-level trigger*. Tech. rep. Final LHCC-approved version. CERN, 2021. URL: https://cds.cern.ch/record/2759072.

[44] CBM Collaboration. *Technical design report for the CBM online systems – Part I: DAQ and FLES entry stage*. Tech. rep. GSI-2023-00739. GSI Helmholtz Centre for Heavy Ion Research, 2023. URL: https://repository.gsi.de/record/340597/files/CBM%20TDR%20Online%20Systems%20-%20Part%20I.pdf.

[45] F. Alessio. "Trigger-less readout architecture for the upgrade of the LHCb experiment at CERN". In: *JINST* 8.12 (Dec. 2013), p. C12019. DOI: 10.1088/1748-0221/8/12/C12019. URL: https://dx.doi.org/10.1088/1748-0221/8/12/C12019.

[46] P. Hazell et al. "Phase stability compliancy testing of a White Rabbit based solution for the LHC RF and timing distribution backbone upgrade". In: *JINST* 19.12 (2024), p. C12019. DOI: 10.1088/1748-0221/19/12/C12019.

[47] M. Lipiński et al. "White Rabbit applications and enhancements". In: *Proc. IEEE Int. Symp. Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS)*. 2018, pp. 1–7. DOI: 10.1109/ISPCS.2018.8543072.

[48] B. P. Abbott et al. “Observation of gravitational waves from a binary black hole merger”. In: *Phys. Rev. Lett.* 116.6 (Feb. 2016), p. 061102. DOI: 10.1103/PhysRevLett.116.061102. URL: https://link.aps.org/doi/10.1103/PhysRevLett.116.061102.

[49] F. An et al. “Neutrino physics with JUNO”. In: *J. Phys. G* 43.3 (Feb. 2016), p. 030401. DOI: 10.1088/0954-3899/43/3/030401. URL: https://dx.doi.org/10.1088/0954-3899/43/3/030401.

[50] M. G. Aartsen et al. *Neutrino astronomy with the next generation IceCube Neutrino Observatory*. 2019. arXiv: 1911.02561 [astro-ph.HE]. URL: https://arxiv.org/abs/1911.02561.

[51] R. Sato et al. “AugerPrime implementation in the DAQ systems of the Pierre Auger Observatory”. In: *Proc. 38th Int. Cosmic Ray Conf. — PoS(ICRC2023)*. Vol. 444. 2023, p. 373. DOI: 10.22323/1.444.0373.

[52] QSolid Project Consortium. *QSolid: Quantum computer in the solid state*. 2022. URL: https://www.quantensysteme.info/projektatlas/projekte/q/qsolid.

[53] T. Muscheid et al. “Full-scale readout electronics for the ECHo experiment”. In: *J. Low Temp. Phys.* 217.3–4 (2024). 54.12.02; LK 01, pp. 456–463. ISSN: 0022-2291, 1573-7357. DOI: 10.1007/s10909-024-03213-y.

[54] J. G. Webster. *Medical instrumentation: Application and design*. Wiley, 2009. ISBN: 9780471676003.

[55] S. Stein and M. Wysession. *An introduction to seismology, earthquakes, and Earth structure*. Wiley-Blackwell, 2009.

[56] S. Ahmad Khan, J. Mitra, and T. K. Nayak. “FPGA-based high-speed DAQ systems for HEP experiments: Potential challenges”. In: *Adv. Radiation Detector and Instrumentation in Nuclear and Particle Physics*. Cham: Springer, 2023, pp. 83–89. DOI: 10.1007/978-3-031-19268-5_11.

[57] Pong P. Chu. *FPGA Prototyping by VHDL Examples: Xilinx MicroBlaze MCS SoC, 2nd Edition*. Accessed: 2025-12-08. Hoboken, NJ, USA: John Wiley & Sons, 2017. ISBN: 978-1-119-28274-7. URL: https://www.lehmanns.de/shop/mathematik-informatik/40218075-9781119282754-fpga-prototyping-by-vhdl-examples.

[58] L. Shannon. “Reconfigurable computing architectures”. In: *Reconfigurable Computing*. Ed. by S. Hauck and A. Dehon. Systems on Silicon. Burlington: Morgan Kaufmann, 2008, pp. 29–46. DOI: 10.1016/B978-012370522-8.50006-6. URL: https://www.sciencedirect.com/science/article/pii/B9780123705228500066.

[59] AMD Xilinx. *UltraScale+ FPGA product selection guide*. Document XMP103 (v2.4.1). July 2024. URL: https://docs.amd.com/v/u/en-US/ultrascale-plus-fpga-product-selection-guide.

[60] A. Li et al. “Evaluating modern GPU interconnect: PCIe, NVLink, NV-SLI, NVSwitch, and GPUDirect”. In: *IEEE Trans. Parallel Distrib. Syst.* 31.1 (2020), pp. 94–110. DOI: 10.1109/TPDS.2019.2928289.

[61] R. Aaij et al. “A comparison of CPU and GPU implementations for the LHCb experiment Run 3 trigger”. In: *Comput. Softw. Big Sci.* 6.1 (Dec. 2021), p. 1. DOI: 10.1007/s41781-021-00070-2. URL: https://doi.org/10.1007/s41781-021-00070-2.

[62] Advanced Micro Devices (AMD). *UltraScale architecture configurable logic block user guide (UG574)*. Revision 1.6. Jan. 2025. URL: https://docs.amd.com/r/en-US/ug574-ultrascale-clb/CLB-Overview.

[63] X. Zang et al. “An open-source fast parallel routing approach for commercial FPGAs”. In: (2024). arXiv: 2407.00009 [cs.DC]. URL: https://arxiv.org/abs/2407.00009.

[64] AMD. *UltraScale Package and Pinout Guide (UG575)*. Accessed: 2025-12-08. 2025. URL: https://docs.amd.com/v/u/en-US/ug575-ultrascale-pkg-pinout.

[65] Magmio. *Top FPGA manufacturers: Leading companies in the FPGA industry*. Accessed 13 June 2025. June 2025. URL: https://www.magmio.com/news/139-fpga-companies.

[66] B. Gaide, D. Gaitonde, C. Ravishankar, et al. “Xilinx adaptive compute acceleration platform: Versal architecture”. In: *Proc. ACM/SIGDA Int. Symp. Field-Programmable Gate Arrays*. 2019, pp. 84–93. DOI: 10.1145/3289602.3293906.

[67] P. Buncic, M. Krzewicki, and P. Vande Vyvre. *Technical Design Report for the Upgrade of the Online-Offline Computing System*. Tech. rep. CERN, 2015. URL: https://cds.cern.ch/record/2011297.

[68] ATLAS Collaboration. *Technical Design Report for the Phase-II Upgrade of the ATLAS TDAQ System*. Tech. rep. Geneva: CERN, 2017. DOI: 10.17181/CERN.2LBB.4IAL. URL: https://cds.cern.ch/record/2285584.

[69] LHCb Collaboration. *LHCb Data Acquisition Enhancement TDR*. Tech. rep. Geneva: CERN, 2024. DOI: 10.17181/CERN.L9E7.N06X. URL: https://cds.cern.ch/record/2886764.

[70] PCI Industrial Computer Manufacturers Group (PICMG). *AdvancedTCA base specification*. Tech. rep. PICMG 3.0, Rev. 3.0. PICMG, Mar. 2008.

[71] VITA Standards Organization. *ANSI/VITA 65.0-2025 – OpenVPX System Standard*. Tech. rep. OpenVPX system standard for VPX interoperability. VITA / ANSI, 2025. URL: https://www.vita.com/Sys/Store/Products/270931.

[72] PCI-SIG. *PCI Express card electromechanical specification, Revision 4.0, Version 1.0*. Tech. rep. Companion to PCIe Base Specification 4.0. Beaverton, OR, USA: PCI-SIG, Sept. 2019.

[73] C. Y. Baldwin and K. B. Clark. *Design Rules, Volume 1: The Power of Modularity*. Cambridge, MA, USA: MIT Press, 2000. DOI: 10.7551/mitpress/2366.001.0001. URL: https://doi.org/10.7551/mitpress/2366.001.0001.

[74] G. Pahl et al. *Engineering Design: A Systematic Approach*. London, UK: Springer, 2007. ISBN: 978-1-84628-318-5. DOI: 10.1007/978-1-84628-319-2. URL: https://link.springer.com/book/10.1007/978-1-84628-319-2.

[75] H. W. Johnson and M. Graham. *High-speed signal propagation: Advanced black magic*. Prentice Hall PTR, 2003.

[76] J. Walrand and P. P. Varaiya. *High Performance Communication Networks*. Morgan Kaufmann Series in Networking. Morgan Kaufmann, 2000. ISBN: 978-1-55860-574-9. URL: https://books.google.de/books?id=FEQCKJFRpYsC.

[77] M. Torngren, D. Chen, and I. Crnkovic. "Component-based vs. model-based development: A comparison in vehicular embedded systems". In: *Proc. EUROMICRO Conf. Software Engineering and Advanced Applications*. 2005, pp. 432–440. DOI: 10.1109/EUROMICRO.2005.18.

[78] INCOSE. *INCOSE systems engineering handbook*. Wiley, 2023. ISBN: 9781119814290.

[79] *ISO/IEC/IEEE 15288:2023 – Systems and software engineering – system life cycle processes*. 15288:2023. ISO/IEC/IEEE. 2023.

[80] E. D. Sozzo, D. Conficconi, A. Zeni, et al. "Pushing the level of abstraction of digital system design: A survey on programming FPGAs". In: *ACM Comput. Surv.* 55.5 (Dec. 2022). DOI: 10.1145/3532989. URL: https://doi.org/10.1145/3532989.

[81] D. Fritchman. "An integrated circuit design framework for human, computer, and ML designers". PhD thesis. University of California, Berkeley, Dec. 2023. URL: http://www2.eecs.berkeley.edu/Pubs/TechRpts/2023/EECS-2023-275.html.

[82] CERN. *Layout of CMS*. Image; accessed 2025-12-15. Courtesy of CERN Audiovisual Media. 1998. URL: https://cds.cern.ch/images/CERN-DI-9803027-1.

[83] G. L. Bayatian et al. *CMS physics: Technical design report volume 1: Detector performance and software*. Technical Design Report. CMS. Geneva: CERN, 2006. URL: https://cds.cern.ch/record/922757.

[84] T. Schörner-Sadenius, ed. *The Large Hadron Collider: A marvel of technology*. Berlin, Heidelberg: Springer, 2015. ISBN: 978-3-642-04573-3. DOI: 10.1007/978-3-642-04574-0.

[85] L. Evans and P. Bryant. "The LHC machine". In: *JINST* 3.08 (Aug. 2008), S08001. DOI: 10.1088/1748-0221/3/08/S08001. URL: https://dx.doi.org/10.1088/1748-0221/3/08/S08001.

[86] P. Langacker. *The standard model and beyond*. 2nd ed. Series in High Energy Physics, Cosmology and Gravitation. Boca Raton, FL: CRC Press, 2017. ISBN: 978-1-4987-6321-9. DOI: 10.1201/b22175. URL: https://library.oapen.org/handle/20.500.12657/50890.

[87] S. Chatrchyan et al. "Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC". In: *Phys. Lett. B* 716 (2012), pp. 30–61. DOI: 10.1016/j.physletb.2012.08.021. arXiv: 1207.7235 [hep-ex].

[88] ATLAS Collaboration. "Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC". In: *Phys. Lett. B* 716.1 (2012), pp. 1–29. DOI: 10.1016/j.physletb.2012.08.020.

[89] N. Craig. "The state of supersymmetry after Run I of the LHC". In: *arXiv:1309.0528* (2014). Lectures delivered at the GGI workshop "Beyond the Standard Model after the first run of the LHC". DOI: 10.48550/arXiv.1309.0528. URL: https://arxiv.org/abs/1309.0528.

[90] J. R. Ellis. "Beyond the Standard Model with the LHC". In: *Nature* 448 (2007), pp. 297–301. DOI: 10.1038/nature06079.

[91] CMS Collaboration. *A transverse slice of the CMS detector and the particles detected by each subdetector*. Online; accessed 23 March 2025. URL: https://cms-opendata-workshop.github.io/workshop2024-lesson-cms-detector/instructor/01-introduction.html.

[92] T. O. James. *A hardware track-trigger for CMS at the High-Luminosity LHC*. Springer Theses. Springer International Publishing, 2019. ISBN: 9783030319342. DOI: 10.1007/978-3-030-31934-2.

[93] A. Albert et al. "The Apollo ATCA design for the CMS track finder and the pixel readout at the HL-LHC". In: *JINST* 17.04 (Apr. 2022), p. C04033. DOI: 10.1088/1748-0221/17/04/C04033. URL: https://doi.org/10.1088/1748-0221/17/04/C04033.

[94] CMS Collaboration. *The Phase-2 Upgrade of the CMS Barrel Calorimeters*. Tech. rep. Final version approved by the LHCC. Geneva: CERN, 2017. URL: https://cds.cern.ch/record/2283187.

[95] P. Sundararajan, K. Ntekas, et al. "New developments in the MDT trigger processor for the ATLAS Level-0 muon trigger at the High-Luminosity LHC". In: *JINST* 18.02 (Feb. 2023), p. C02030. DOI: 10.1088/1748-0221/18/02/C02030.

[96] A. Greshilov. *The X2O ATCA IPMC and control solution: Update*. 4th CERN SoC Workshop, Geneva. Oct. 2025. URL: https://indico.cern.ch/event/1518977/contributions/6644617/attachments/3162600/5619666/2025-10-28_X2O_ATCA_IPMC_and_control_solution_AG.pdf.

[97] J. Alawieh, M. Durasov, U. Behrens, et al. “Phase-2 CMS DAQ: Growing from prototype boards to demonstrator systems”. In: *JINST* 20.02 (Feb. 2025), p. C02005. DOI: 10.1088/1748-0221/20/02/C02005. URL: https://dx.doi.org/10.1088/1748-0221/20/02/C02005.

[98] A. Polyneikis and T. Tzanis. *CMS Phase-2 DAQ: Architecture and progress*. TWEPP 2025, Crete. Oct. 2025. URL: https://indico.cern.ch/event/1502285/contributions/6554634/attachments/3149329/5592157/CMS_Phase2_DAQ_Polyneikis_Tzanis_TWEPP_2025.pdf.

[99] P. Tzanis et al. *Integration of the Upgraded CMS Data Readout System for LHC Phase-2*. Presentation at TWEPP 2025, Rethymno, Crete. Accessed: 2026-01-10. Oct. 2025. URL: https://indico.cern.ch/event/1502285/contributions/6554634/.

[100] V. Sola. “Precision timing with the CMS MTD endcap timing layer”. In: *Proc. 29th Int. Workshop on Vertex Detectors (VERTEX2020)*. Tsukuba, Ibaraki, Japan: Physical Society of Japan, Oct. 2021, pp. 010013.1–010013.10. DOI: 10.7566/JPSCP.34.010013. URL: https://cds.cern.ch/record/2789997/files/jpscp.34.010013.pdf.

[101] S. Baron et al. “Jitter impact on clock distribution in LHC experiments”. In: *JINST* 7.12 (Dec. 2012), p. C12023. DOI: 10.1088/1748-0221/7/12/C12023. URL: https://dx.doi.org/10.1088/1748-0221/7/12/C12023.

[102] P. Klabbers et al. “Performance of the CMS regional calorimeter trigger”. In: (2009). DOI: 10.5170/CERN-2009-006.191. URL: https://cds.cern.ch/record/1234892.

[103] ALEPH Collaboration. *ALEPH technical report*. Geneva: CERN, 1983. URL: https://cds.cern.ch/record/300680.

[104] DELPHI Collaboration. “The DELPHI detector at LEP”. In: *Nucl. Instrum. Methods Phys. Res. A* 303.2 (1991), pp. 233–276. DOI: 10.1016/0168-9002(91)90793-P. URL: https://www.sciencedirect.com/science/article/pii/016890029190793P.

[105] O. Callot, M. Cattaneo, M. Frank, et al. *Experience with the ALEPH online system*. Tech. rep. Geneva: CERN, 2001. URL: https://cds.cern.ch/record/529545.

[106] K. Compton et al. “The MP7 and CTP-6: Multi-hundred Gbps processing boards for calorimeter trigger upgrades at CMS”. In: *JINST* 7.12 (Dec. 2012), p. C12024. DOI: 10.1088/1748-0221/7/12/C12024. URL: https://dx.doi.org/10.1088/1748-0221/7/12/C12024.

[107] A. Whitbeck and J. Hirschauer. “The CMS central hadron calorimeter DAQ system upgrade”. In: *JINST* 10.05 (May 2015), p. C05019. DOI: 10.1088/1748-0221/10/05/C05019. URL: https://dx.doi.org/10.1088/1748-0221/10/05/C05019.

[108] A. Triossi. *The CMS Barrel Muon Trigger upgrade*. Tech. rep. Geneva: CERN, 2017. DOI: 10.1088/1748-0221/12/01/C01095. URL: https://cds.cern.ch/record/2252301.

[109] D. Acosta et al. "The CMS modular track finder boards, MTF6 and MTF7". In: *JINST* 8.12 (Dec. 2013), p. C12034. DOI: 10.1088/1748-0221/8/12/C12034. URL: https://dx.doi.org/10.1088/1748-0221/8/12/C12034.

[110] CMS Tracker Collaboration. "The CMS Phase-1 pixel detector upgrade". In: *JINST* 16.02 (Feb. 2021), P02027. DOI: 10.1088/1748-0221/16/02/P02027. URL: https://doi.org/10.1088/1748-0221/16/02/P02027.

[111] A. Kugel et al. *The final design of the ATLAS Trigger/DAQ readout-buffer input (ROBIN) device*. Tech. rep. Geneva: CERN, 2004. DOI: 10.5170/CERN-2004-010.258. URL: https://cds.cern.ch/record/802272.

[112] R. E. Blair et al. *A prototype ROI builder for the second level trigger of ATLAS implemented in FPGAs*. Tech. rep. Geneva: CERN, 1999. URL: https://cds.cern.ch/record/683975.

[113] F. Costa et al. "The ALICE C-RORC GBT card: A prototype readout solution for the ALICE upgrade". In: *Proc. IEEE-NPSS Real Time Conf. (RT)*. 2016, pp. 1–5. DOI: 10.1109/RTC.2016.7543109.

[114] A. Borga et al. "The C-RORC PCIe card and its application in the ALICE and ATLAS experiments". In: *JINST* 10.02 (Feb. 2015), p. C02022. DOI: 10.1088/1748-0221/10/02/C02022. URL: https://dx.doi.org/10.1088/1748-0221/10/02/C02022.

[115] F. Pastore. *ATLAS TDAQ upgrades for Phase-2*. Tech. rep. Geneva: CERN, 2024. URL: https://cds.cern.ch/record/2907275.

[116] R. Spiwoks et al. *The ATLAS Level-1 central trigger processor (CTP)*. Tech. rep. CERN, 2005. DOI: 10.5170/CERN-2005-011.265. URL: https://cds.cern.ch/record/889547.

[117] G. Aad et al. *Technical design report for the Phase-I upgrade of the ATLAS TDAQ system*. Tech. rep. Final version presented to the December 2013 LHCC. CERN, 2013. URL: https://cds.cern.ch/record/1602235.

[118] S. Tang et al. "gFEX: The ATLAS calorimeter Level-1 real-time processor". In: *Proc. IEEE Nucl. Sci. Symp. Med. Imaging Conf. (NSS/MIC)*. 2015, pp. 1–5. DOI: 10.1109/NSSMIC.2015.7581865.

[119] S. Tang et al. "Implementation of the Global Feature Extractor (gFEX) for the ATLAS calorimeter Level-1 trigger system for LHC Run 3 upgrade". In: *JINST* 13.07 (July 2018), P07010. DOI: 10.1088/1748-0221/13/07/P07010. URL: https://dx.doi.org/10.1088/1748-0221/13/07/P07010.

[120] W. Qian et al. “A full-function Global Common Module prototype for the ATLAS Phase-II upgrade”. In: *JINST* 19.02 (Feb. 2024), p. C02049. DOI: 10.1088/1748-0221/19/02/C02049.

[121] J. Buytaert. “The frontend electronics for the LHCb upgrade”. In: *JINST* 5.12 (Dec. 2010), p. C12035. DOI: 10.1088/1748-0221/5/12/C12035. URL: https://dx.doi.org/10.1088/1748-0221/5/12/C12035.

[122] J. P. Cachemiche et al. “The PCIe-based readout system for the LHCb experiment”. In: *JINST* 11.02 (Feb. 2016), P02013. DOI: 10.1088/1748-0221/11/02/P02013. URL: https://dx.doi.org/10.1088/1748-0221/11/02/P02013.

[123] M. Bellato et al. “A PCIe Gen3-based readout for the LHCb upgrade”. In: *J. Phys. Conf. Ser.* 513.1 (June 2014), p. 012023. DOI: 10.1088/1742-6596/513/1/012023. URL: https://dx.doi.org/10.1088/1742-6596/513/1/012023.

[124] J. J. Langouet. *Technical challenges designing a prototype common readout board for LHCb future upgrades*. 2024. arXiv: 2411.01648 [physics.ins-det]. URL: https://arxiv.org/abs/2411.01648.

[125] F. Carena et al. “The ALICE data-acquisition read-out receiver card”. In: *10th Workshop on Electronics for LHC and Future Experiments*. 2004. DOI: 10.5170/CERN-2004-010.273. URL: https://cds.cern.ch/record/814255.

[126] O. Bourrion et al. “Versatile firmware for the common readout unit (CRU) of the ALICE experiment at the LHC”. In: *JINST* 16.05 (May 2021), P05019. DOI: 10.1088/1748-0221/16/05/P05019. URL: https://dx.doi.org/10.1088/1748-0221/16/05/P05019.

[127] S. Yamada et al. “Common readout subsystem for the Belle II experiment and its performance measurement”. In: *IEEE Trans. Nucl. Sci.* 64.6 (2017), pp. 1415–1419. DOI: 10.1109/TNS.2017.2693297.

[128] Y.-T. Lai et al. “Commissioning and operation of the upgraded Belle II DAQ system with PCIe-based high-speed readout”. In: *IEEE Trans. Nucl. Sci.* 70.6 (2023), pp. 890–897. DOI: 10.1109/TNS.2022.3228288.

[129] T. Geßler, W. Kühn, J. S. Lange, et al. “The ONSEN data reduction system for the Belle II pixel detector”. In: *IEEE Trans. Nucl. Sci.* 62.3 (2015), pp. 1149–1154. DOI: 10.1109/TNS.2015.2414713.

[130] AMD Xilinx. *Virtex-II platform FPGAs: Complete data sheet*. Document DS031 (v3.4). Apr. 2014. URL: https://docs.amd.com/v/u/en-US/ds031.

[131] AMD Xilinx. *Virtex-4 family overview data sheet*. Document DS112 (v3.1). Aug. 2010. URL: https://docs.amd.com/v/u/en-US/ds112.

[132] AMD Xilinx. *Virtex-5 family overview data sheet*. Document DS100 (v5.1). Aug. 2015. URL: https://docs.amd.com/v/u/en-US/ds100.

[133] AMD Xilinx. *Virtex-6 family overview data sheet*. Document DS150 (v2.5). Aug. 2015. URL: https://docs.amd.com/v/u/en-US/ds150.

[134] AMD Xilinx. *7-series product selection guide*. Document XMP101 (v1.8). Apr. 2021. URL: https://docs.amd.com/v/u/en-US/7-series-product-selection-guide.

[135] AMD Xilinx. *UltraScale FPGA product selection guide*. Document XMP102 (v1.7). Feb. 2016. URL: https://docs.amd.com/v/u/en-US/ultrascale-fpga-product-selection-guide.

[136] AMD Xilinx. *Versal Premium series product selection guide*. Document XMP463 (v1.15). June 2025. URL: https://docs.amd.com/v/u/en-US/versal-premium-psg.

[137] B. Kreis. "The online calibration, operation, and performance of the CMS pixel detector". In: *Nucl. Instrum. Meth. A* 650.1 (2011), pp. 14–18. DOI: 10.1016/j.nima.2010.11.083. URL: https://www.sciencedirect.com/science/article/pii/S016890021002588X.

[138] V. Veszpremi. "Operation and performance of the CMS tracker". In: *JINST* 9.03 (Mar. 2014), p. C03005. DOI: 10.1088/1748-0221/9/03/C03005. URL: https://dx.doi.org/10.1088/1748-0221/9/03/C03005.

[139] B. Vormwald. "Operational experience of the Phase-1 CMS pixel detector". In: *Nucl. Instrum. Meth. A* 958 (2020). Proceedings of the Vienna Conference on Instrumentation 2019, p. 162018. DOI: 10.1016/j.nima.2019.03.073. URL: https://www.sciencedirect.com/science/article/pii/S0168900219304152.

[140] W. Adam et al. "The CMS Phase-1 pixel detector upgrade". In: *JINST* 16.02 (Feb. 2021), P02027. DOI: 10.1088/1748-0221/16/02/P02027. URL: https://dx.doi.org/10.1088/1748-0221/16/02/P02027.

[141] ECFA Detector R&D Roadmap Process Group. *The 2021 ECFA detector research and development roadmap*. Tech. rep. Geneva, 2020. DOI: 10.17181/CERN.XDPL.W2EX. URL: https://cds.cern.ch/record/2784893.

[142] T. Behnke, J. E. Brau, B. Foster, et al. *International Linear Collider technical design report, Volume 1: Executive summary*. Tech. rep. International Linear Collider, June 2013. URL: https://www.linearcollider.org/files/images/pdf/Executive%20Summary.pdf.

[143] M. Benedikt, F. Zimmermann, B. Auchmann, et al. *Future Circular Collider Feasibility Study Report Volume 1*. Tech. rep. Submitted to European Physics Journal ST. Geneva: CERN, 2025. DOI: 10.17181/CERN.9DKX.TDH9. arXiv: 2505.00272. URL: https://cds.cern.ch/record/2928193.

[144] T. Roser et al. "Snowmass 2022 Accelerator Frontier Implementation Task Force report". In: *Snowmass Community Summer Study*. Presentation by T. Roser (BNL); contributed July 18, 2022. July 2022. URL: https://indico.fnal.gov/event/22303/contributions/245261/attachments/157255/205819/2022%2007%2018%20ITF%20Snowmass%20talk.pdf (visited on 09/07/2025).

[145] A. Subba et al. "The Linear Collider Facility (LCF) at CERN". In: (June 2025). arXiv: 2503.24049 [hep-ex].

[146] M. Benedikt et al. *Future Circular Collider feasibility study report volume 2: Accelerators, technical infrastructure and safety*. 2025. DOI: 10.17181/CERN.EBAY.7W4X. URL: http://cds.cern.ch/record/2928793.

[147] D. Dannheim, L. Linssen, and A. Sailer. "Progress on detector R&D for CLIC 2019–2025". In: (2025). URL: https://cds.cern.ch/record/2927639.

[148] M. Benedikt, F. Zimmermann, B. Auchmann, et al. *FCC integrated programme stage 2: The FCC-hh*. Tech. rep. Geneva: CERN, 2025. DOI: 10.17181/CERN.M6OU.VG88. URL: https://cds.cern.ch/record/2928941.

[149] CEPC Study Group. *CEPC conceptual design report: Volume 2 — physics and detector*. 2018. arXiv: 1811.10545 [hep-ex]. URL: https://arxiv.org/abs/1811.10545.

[150] J. Baudot et al. *DRD 7: Proposal for an R&D collaboration on electronics and on-detector processing*. Tech. rep. CERN, 2024. URL: https://cds.cern.ch/record/2901965.

[151] D. Acosta et al. *Snowmass topical group summary report: IF04 – Trigger and data acquisition systems*. 2022. arXiv: 2209.03794 [physics.ins-det]. URL: https://arxiv.org/abs/2209.03794.

[152] P. S. Barbeau et al. *Report of the instrumentation frontier working group for Snowmass 2021*. 2022. arXiv: 2209.14111 [hep-ex]. URL: https://arxiv.org/abs/2209.14111.

[153] A. V. Kotwal. "A fast method for particle tracking and triggering using small-radius silicon detectors". In: *Nucl. Instrum. Methods Phys. Res. A* 957 (Mar. 2020), p. 163427. DOI: 10.1016/j.nima.2020.163427. URL: http://dx.doi.org/10.1016/j.nima.2020.163427.

[154] L. Boggia et al. *Review of machine learning for real-time analysis at the Large Hadron Collider experiments ALICE, ATLAS, CMS and LHCb*. 2025. arXiv: 2506.14578 [hep-ex]. URL: https://arxiv.org/abs/2506.14578.

[155] E. Wittenhagen, U. Hecht, H. Cirit, et al. “A 224 Gbit/s transceiver front-end design for next-generation data centers”. In: *Proc. IEEE Int. Conf. Electron., Circuits and Systems (ICECS)*. 2022, pp. 1–4. DOI: 10.1109/ICECS202256217.2022.9970895.

[156] R. Gebauer et al. “QiCells: A modular RFSoC-based approach to interface superconducting quantum bits”. In: *ACM Trans. Reconfigurable Technol. Syst.* 16.2 (May 2023). DOI: 10.1145/3571820. URL: https://doi.org/10.1145/3571820.

[157] R. Gartmann et al. “Mixerless RFSoC microwave signal generation for superconducting circuit applications”. In: *Proc. IEEE Int. Conf. Quantum Computing and Engineering (QCE)*. Vol. 02. 2024, pp. 565–566. DOI: 10.1109/QCE60285.2024.10407.

[158] A. K. Sinclair et al. *CCAT-prime: RFSoC-based readout for frequency multiplexed kinetic inductance detectors*. 2022. arXiv: 2208.07465 [astro-ph.IM]. URL: https://arxiv.org/abs/2208.07465.

[159] CMMI Institute. *Capability maturity model integration (CMMI) institute*. Accessed: 2025-11-05. URL: https://cmmiinstitute.com/.

[160] 6 Sigma Global Institute. *Capability maturity model integration (CMMI) – process improvement*. Accessed: 2025-11-05. URL: https://www.6sigma.us/process-improvement/capability-maturity-model-integration-cmmi/.

[161] M. Chaudhary and A. Chopra. *CMMI for development*. SpringerLink. Apress, 2017. URL: http://dx.doi.org/10.1007/978-1-4842-2529-5.

[162] CMMI Institute. *CMMI levels of capability and performance*. Accessed: 2025-09-24. URL: https://cmmiinstitute.com/learning/appraisals/levels.

[163] CMMI Product Team. *CMMI for development (CMMI-DEV), version 1.3*. Tech. rep. CMU/SEI-2010-TR-033. Software Engineering Institute, Carnegie Mellon University, 2010. DOI: 10.1184/R1/6572342.

[164] CMMI Product Team. *CMMI for services (CMMI-SVC), version 1.3*. Tech. rep. CMU/SEI-2010-TR-034. Software Engineering Institute, Carnegie Mellon University, 2010. DOI: 10.1184/R1/6572375.

[165] CMMI Product Team. *CMMI for acquisition (CMMI-ACQ), version 1.2*. Tech. rep. CMU/SEI-2010-TR-032. Software Engineering Institute, Carnegie Mellon University, 2025. DOI: 10.1184/R1/6572291.

[166] IPC. *IPC-2221B: Generic standard on printed board design*. IPC standard. 2012.

[167] K. Mitzner et al. *Complete PCB design using OrCAD Capture and PCB Editor*. 2nd ed. Academic Press / Elsevier, 2019. ISBN: 978-0-12-817684-9.

[168] A. Pulli et al. *HEP verification maturity model*. Presented by A. Pulli on behalf of the CERN EP-ESE-ME Verification Group. CERN, Geneva, Oct. 2023. URL: https://indico.cern.ch/event/1255624/contributions/5445241/ (visited on 09/09/2025).

[169] O. Manzhura, M. Caselle, L. E. Ardila-Perez, et al. "The data acquisition system for the PANDA micro-vertex detector". In: *JINST* 19.03 (Mar. 2024), p. C03036. DOI: 10.1088/1748-0221/19/03/C03036.

[170] Altium LLC. *Altium Designer*. Accessed: 2025-11-05. 2025. URL: https://www.altium.com/altium-designer/.

[171] Cadence Design Systems. *Allegro PCB Designer*. Accessed: 2025-11-05. 2025. URL: https://www.cadence.com/en_US/home/tools/pcb-design-and-analysis/allegro-system-design.html.

[172] Siemens EDA. *PADS PCB design*. Accessed: 2025-11-05. 2025. URL: https://eda.sw.siemens.com/en-US/pcb/pads/.

[173] Siemens EDA. *Xpedition PCB design*. Accessed: 2025-11-05. 2025. URL: https://eda.sw.siemens.com/en-US/pcb/xpedition/.

[174] KiCad Project. *KiCad EDA suite*. Accessed: 2025-11-05. 2025. URL: https://www.kicad.org/.

[175] J. Bachrach, D. Biancolin, A. Buchan, et al. "JITPCB". In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*. 2016, pp. 2230–2236. DOI: 10.1109/IROS.2016.7759349.

[176] JitX. *JitX*. Accessed: 2025-11-05. URL: https://www.jitx.com/.

[177] R. Lin, R. Ramesh, C. Chi, et al. "Polymorphic blocks: Unifying high-level specification and low-level control for circuit board design". In: *Proc. ACM Symp. User Interface Software and Technology (UIST '20)*. 2020, pp. 529–540. DOI: 10.1145/3379337.3415860. URL: https://doi.org/10.1145/3379337.3415860.

[178] A. Mashtizadeh. "PHDL: A Python hardware design framework". Accessed: 2025-11-05. PhD thesis. Massachusetts Institute of Technology, 2007. URL: https://dspace.mit.edu/handle/1721.1/41543.

[179] Atopile. *Atopile – Real-time collaboration platform*. Accessed: 2025-11-05. URL: https://atopile.io/.

[180] KiCad Project. *kicad-library-utils: Library management and symbol/footprint generation scripts*. Accessed: 2025-11-05. 2025. URL: https://gitlab.com/kicad/libraries/kicad-library-utils.

[181] Analog Devices, Inc. *LTM4676A Rev. B data sheet: Dual 13 A or single 26 A μModule regulator with digital power system management*. Tech. rep. Rev. B. Analog Devices, Inc., Jan. 2024. URL: https://www.analog.com/media/en/technical-documentation/data-sheets/4676afa.pdf.

[182] M. Henderson, S. Shurville, K. Fernstrom, et al. "Extraneous information and graph comprehension: Implications for effective design choices". In: *Campus-Wide Information Systems* 26.3 (June 2009), pp. 191–200. DOI: 10.1108/10650740910967375. URL: https://doi.org/10.1108/10650740910967375.

[183] Q. Wu, Z. Tan, and J. Liu. "Experimental study on the relationship between harmony and cognitive load of business intelligence dashboard color combinations". In: *Color Res. Appl.* 47.4 (2022), pp. 920–941. DOI: 10.1002/col.22768. URL: https://onlinelibrary.wiley.com/doi/10.1002/col.22768.

[184] M. Harris. *Easy schematics creation for elegance and readability*. Apr. 2021. URL: https://resources.altium.com/p/creating-elegant-and-readable-schematics.

[185] PICMG. *MicroTCA.0 specification R1.0: Micro telecommunications computing architecture base specification*. Tech. rep. Version R1.0. PCI Industrial Computer Manufacturers Group, July 2006. URL: https://www.picmg.org/openstandards/microtca/.

[186] Multi-Circuit-Boards. *Leiterplatten – Basic Design Rules*. Accessed: 2025-12-08. 2025. URL: https://www.multi-circuit-boards.eu/fileadmin/user_upload/downloads/leiterplatten_design-hilfe/Multi-CB-Leiterplatten_Basic-Design-Rules.pdf.

[187] P. Hazell. *New PLL evaluations for high precision timing*. Tech. rep. Accessed: 2025-11-05. CERN, 2025. URL: https://indico.cern.ch/event/1283971/#35-new-pll-evaluations-for-hig.

[188] R. Aggleton, L. E. Ardila-Perez, F. Ball, et al. "An FPGA-based track finder for the L1 trigger of the CMS experiment at the High-Luminosity LHC". In: *JINST* 12.12 (Dec. 2017), P12019. DOI: 10.1088/1748-0221/12/12/P12019.

[189] S. Mallios et al. "Overview of the CMS HGCAL backend electronics and vertical integration system tests". In: *Proc. CALOR2024*. Accessed: 2025-09-25. Tsukuba, Japan, May 2024. URL: https://indico.cern.ch/event/1339557/contributions/5917852/attachments/2863087/5009761/An%20overview%20of%20the%20CMS%20HGCAL%20backend%20electronics%20and%20vertical%20integration%20system%20tests%20v2.pdf.

[190] P. Moreira, S. Kulis, S. Baron, et al. "lpGBT: Low-power radiation-hard multipurpose high-speed transceiver ASIC for high-energy physics experiments". In: *IEEE Trans. Nucl. Sci.* 72.1 (2025), pp. 24–37. DOI: 10.1109/TNS.2024.3506753.

[191] T. Gorski. *CMS standard protocol*. Tech. rep. Accessed: 2025-11-05. CERN / CMS Collaboration, Apr. 2022. URL: https://indico.cern.ch/event/1137504/contributions/4832473/attachments/2434223/4168920/CMS_Standard_Protocol_29Apr2022.pdf.

[192] D. Gigi, J. Hegeman, and C. Schwick. *Phase-2 CMS SLinkRocket DAQ link: Protocol and format description*. Tech. rep. EDMS 2502737. Accessed: 2025-11-05. CERN, 2020. URL: https://edms.cern.ch/document/2502737/.

[193] CMS Collaboration. *The Phase-2 Upgrade of the CMS L1 Trigger Interim Technical Design Report*. Tech. rep. CMS Interim TDR for the Phase-2 L1 trigger upgrade, approved by the LHCC. Geneva: CERN, 2017. DOI: 10.17181/CERN.UUWZ.FMIS. URL: https://cds.cern.ch/record/2283192.

[194] B. Acar, G. Adamov, C. Adloff, et al. "Timing performance of the CMS High Granularity Calorimeter prototype". In: *JINST* 19.04 (Apr. 2024), P04015. DOI: 10.1088/1748-0221/19/04/P04015.

[195] AMD. *Virtex UltraScale+ FPGA data sheet: DC and AC switching characteristics*. Tech. rep. DS923, v1.20. Accessed: 2025-05-12. AMD, Aug. 2024. URL: https://docs.amd.com/r/en-US/ds923-virtex-ultrascale-plus.

[196] G. Fedi, S. Fiorendi, M. Holmberg, et al. "Lessons learnt from the first vertical slice of the CMS Outer Tracker". In: *JINST* 18.01 (2023), p. C01041. DOI: 10.1088/1748-0221/18/01/C01041.

[197] Serenity Consortium. *Serenity project hardware overview*. Accessed: 2025-11-05. 2025. URL: https://serenity.web.cern.ch/serenity/hardware/.

[198] L. E. Ardila Pérez. "Low-latency track triggering in high-energy physics". Electronics, programmable logic, FPGAs, CMS trigger systems. PhD thesis. Karlsruher Institut für Technologie (KIT), 2022. DOI: 10.5445/IR/1000145595.

[199] A. Rose et al. "Serenity: An ATCA prototyping platform for CMS Phase 2". In: *PoS* TWEPP2018 (2019), p. 115. DOI: 10.22323/1.343.0115.

[200] Samtec Inc. *Z-Ray interposer design guide*. Tech. rep. Samtec Inc., 2023. URL: https://elcon-electronic.de/fileadmin/user_upload/samtec-zray-interposer-design-guide.pdf.

[201] CERN. *LHC long-term schedule*. Accessed: 2025-05-24. Geneva, Switzerland, Sept. 2024. URL: https://lhc-commissioning.web.cern.ch/schedule/LHC-long-term.htm.

[202] Omnion Power Electronics. *PIM400 – isolated, non-regulated board-mounted power module*. Accessed: 2025-11-05. 2025. URL: https://www.omnionpower.com/products/board-mounted-power/isolated-non-regulated-board-mounted-power/pim400.

[203] Omnion Power Electronics. *High-power Barracuda – isolated regulated board-mounted power*. Accessed: 2025-11-05. 2025. URL: https://www.omnionpower.com/products/board-mounted-power/isolated-regulated-board-mounted-power/high-power-barracuda.

[204] L. Calligaris, A. Cascadan, L. E. Ardila-Perez, et al. "OpenIPMC: A free and open-source intelligent platform management controller software". In: *IEEE Trans. Nucl. Sci.* 68.8 (2021), pp. 2105–2112. DOI: 10.1109/TNS.2021.3092689. arXiv: 2011.01088 [physics.ins-det].

[205] AMD. *AXI Chip2Chip interface documentation*. Accessed: 2025-11-05. 2025. URL: https://docs.amd.com/r/en-US/pg067-axi-chip2chip.

[206] G. Badaro, U. Behrens, A. Bocci, et al. "Phase-2 upgrade of the CMS data acquisition". In: *EPJ Web Conf.* 251 (2021), p. 04023. DOI: 10.1051/epjconf/202125104023. URL: https://doi.org/10.1051/epjconf/202125104023.

[207] M. Sweney. *Global shortage in computer chips reaches crisis point*. Mar. 2021. URL: https://www.theguardian.com/business/2021/mar/21/global-shortage-in-computer-chips-reaches-crisis-point.

[208] F. Jackson. *Global chip shortage: Everything you need to know*. Sept. 2024. URL: https://www.techrepublic.com/article/global-chip-shortage-cheat-sheet/.

[209] Ltd. Elite Material Co. *EM-890 datasheet*. Accessed: 2025-12-04. 2024. URL: https://www.emctw.com/upload/website/item/EM-890%20Datasheet%2020240716_24081910973.pdf.

[210] Taiwan Union Technology Corporation. *ThunderClad 2A Sp*. Accessed: 2025-12-04. 2025. URL: https://www.tuc.com.tw/_run.php?work=down_member&id=158.

[211] Inc. Ansys. *Ansys SIwave – high-fidelity signal integrity and power integrity analysis tool*. Accessed: 2025-12-04. 2025. URL: https://www.ansys.com/products/electronics/ansys-siwave.

[212] Serenity Consortium. *SMASH functionality documentation*. Accessed: 2025-12-07. 2025. URL: https://serenity.web.cern.ch/serenity/smash/index.html.

[213] C. Larrea et al. "IPbus: A flexible Ethernet-based control system for xTCA hardware". In: *JINST* 10.02 (Feb. 2015), p. C02019. DOI: 10.1088/1748-0221/10/02/C02019. URL: https://doi.org/10.1088/1748-0221/10/02/C02019.

[214] T. Williams. *EMP – A common infrastructure firmware framework for the CMS Phase-2 upgrades*. Oral presentation at TWEPP 2025, Rethymno, Crete. Accessed: 2025-12-08. Oct. 2025. URL: https://indico.cern.ch/event/1502285/contributions/6554465/.

[215] Serenity Consortium. *EMP framework documentation*. Accessed: 2025-12-08. 2025. URL: https://serenity.web.cern.ch/serenity/emp-fwk/index.html.

[216] Serenity Consortium. *Board-Setup – Cheatsheet*. Accessed: 2025-12-08. 2025. URL: https://serenity.web.cern.ch/board-setup-z1/cheatsheet.html.

[217] S. Bologna et al. "SWATCH: Common software for controlling and monitoring the upgraded Level-1 trigger of the CMS experiment". In: *Proc. IEEE NPSS Real Time Conference (RT)*. Padua, Italy, 2016, pp. 1–6. DOI: 10.1109/RTC.2016.7543077.

[218] T. Williams et al. "System design and prototyping of the CMS Level-1 trigger at the High-Luminosity LHC". In: *JINST* 19.03 (Mar. 2024), p. C03016. DOI: 10.1088/1748-0221/19/03/C03016.

[219] Texas Instruments. *TPSM8D6B24 – Power Module Data Sheet*. Accessed: 2025-12-08. 2025. URL: https://www.ti.com/lit/ds/symlink/tpsm8d6b24.pdf.

[220] Analog Devices. *LTM4680 – Power Module Data Sheet*. Accessed: 2025-12-08. 2025. URL: https://www.analog.com/media/en/technical-documentation/data-sheets/ltm4680.pdf.

[221] Analog Devices. *LTM4703 – Power Module Data Sheet*. Accessed: 2025-12-08. 2025. URL: https://www.analog.com/media/en/technical-documentation/data-sheets/ltm4703.pdf.

[222] P. Boning and S. Dubowsky. "Identification of actuation efforts using limited sensory information for space robots". In: *Proc. IEEE Int. Conf. Robotics and Automation*. 2006, pp. 3873–3878. DOI: 10.1109/ROBOT.2006.1642295.

[223] H. Krause et al. *Factory Acceptance Test for the CMS Phase-2 Back-End Card Serenity-S1*. Oral presentation at TWEPP 2025, Rethymno, Crete. Accessed: 2025-12-08. Sept. 2025. URL: https://indico.cern.ch/event/1502285/contributions/6554590/.

[224] M. Pecht. *Product reliability, maintainability and supportability handbook*. Boca Raton, FL, USA: CRC Press, 2009.

[225] Raghunandan Shukla. *Standalone Timing Distribution and Event Capture Firmware for Back-End Electronics Test Stands*. Oral presentation at TWEPP 2025, Rethymno, Crete. Accessed: 2025-12-08. Oct. 2025. URL: https://indico.cern.ch/event/1502285/contributions/6554671/.

[226] E. Leutgeb. *The Phase 2 Global Trigger / BRIL ATCA Rear Transition Module*. Poster presentation at TWEPP 2025, Rethymno, Crete. Accessed: 2025-12-08. Oct. 2025. URL: https://indico.cern.ch/event/1502285/contributions/6554683/.

[227] M. O. Sahin and T. Orimoto. *A unified DAQ system for the CMS MIP Timing Detector (MTD)*. Oral presentation at TWEPP 2025, Rethymno, Crete. Accessed: 2025-12-08. Oct. 2025. URL: https://indico.cern.ch/event/1502285/contributions/6554632/.

[228] H. Bin Salim et al. *Implementation of ML based conditions for the Phase-2 CMS Global Trigger upgrade*. Poster presentation at TWEPP 2025, Rethymno, Crete. Accessed: 2025-12-08. Oct. 2025. URL: https://indico.cern.ch/event/1502285/contributions/6554680/.

[229] D. Cock, A. Ramdas, D. Schwyn, et al. "Enzian: An open, general CPU/FPGA platform for systems software research". In: *Proc. 27th ACM Int. Conf. Architectural Support Programming Languages Operating Systems (ASPLOS '22)*. 2022, pp. 434–451. DOI: 10.1145/3503222.3507742. URL: https://doi.org/10.1145/3503222.3507742.

[230] TE Connectivity, Inc. *6469001-1: Z-PACK high-speed backplane connector, 80-position, 8-row × 10-column*. Accessed: 2025-09-27. URL: https://www.te.com/de/product-6469001-1.html.

[231] AMD. *VPK180 evaluation board user guide (UG1582)*. Tech. rep. UG1582, Rev. 1.3. AMD, June 2025. URL: https://docs.amd.com/r/en-US/ug1582-vpk180-eval-bd.

[232] Altium Resources. *PCB land pattern design and the IPC-7351 standard*. Accessed: 2025-11-05. URL: https://resources.altium.com/de/p/pcb-land-pattern-design-ipc-7351-standard.

[233] Multi-Circuit-Boards. *Definierter Lagenaufbau und Impedanzen*. Accessed: 2025-12-08. 2025. URL: multi-circuit-boards.eu/fileadmin/pdf/leiterplatten_lagenaufbau/Multi-CB_Definierter_Lagenaufbau_Impedanzen_en.pdf.

[234] Altium. *PCB Footprint Library*. Accessed: 2025-12-18. 2025. URL: https://resources.altium.com/p/pcb-footprint-library.

[235] AMD. *Zynq UltraScale+ Product Selection Guide (XMP104)*. Accessed: 2025-12-08. 2025. URL: https://docs.amd.com/v/u/en-US/zynq-ultrascale-plus-product-selection-guide.

[236] AMD. *Zynq UltraScale+ RFSoC Product Selection Guide (XMP105)*. Accessed: 2025-12-08. 2025. URL: https://docs.amd.com/v/u/en-US/zynq-usp-rfsoc-product-selection-guide.

[237] AMD. *Versal AI Core Product Selection Guide (XMP452)*. Accessed: 2025-12-08. 2025. URL: https://docs.amd.com/v/u/en-US/versal-ai-core-product-selection-guide.

[238] QSFP-DD MSA. *QSFP-DD Hardware Specification – Rev. 7.1*. Accessed: 2025-12-08. 2024. URL: http://www.qsfp-dd.com/wp-content/uploads/2024/07/QSFP-DD-Hardware-Rev7.1.pdf.

[239] OSFP MSA. *OSFP Module Specification – Rev. 5.22*. Accessed: 2025-12-08. 2025. URL: https://osfpmsa.org/assets/pdf/OSFP_Module_Specification_Rev5_22.pdf.