Republic of Türkiye

ANKARA UNIVERSITY

INSTITUTE OF ACCELERATOR TECHNOLOGIES

MASTER'S THESIS

EVOLUTION OF THE MALTA2 READOUT FIRMWARE TO ULTRASCALE AND EVALUATION FOR ITK REPLACEMENT IN ATLAS

Fuat Kerem IŞIK

DEPARTMENT OF ACCELERATOR AND DETECTOR TECHNOLOGIES

ANKARA

2026

All rights reserved.

ÖZET

Yüksek Lisans Tezi

MALTA2 Okuma Aygıt Yazılımının UltraScale'e Evrimi ve ATLAS'ta ITk'nin Yerini Alması Açısından Değerlendirilmesi

Fuat Kerem IŞIK

Ankara Üniversitesi
Hızlandırıcı Teknolojileri Enstitüsü
Hızlandırıcı Teknolojileri Anabilim Dalı

Danışman: Prof. Dr. İlkay TÜRK ÇAKIR
Eş Danışman: Dr. Carlos A. SOLANS SANCHEZ

Bu tezde, ALINX AXKU040 geliştirme kartı kullanılarak MALTA2 monolitik aktif piksel sensörü okuma aygıt yazılımının Kintex UltraScale mimarisine evrimi sunulmaktadır. Çalışma, mevcut DAQ yazılım arayüzüyle uyumluluğunu korurken MALTA2 laboratuvar ölçümleri, demet testi uygulamaları ve ATLAS ITk referanslı yüksek oran çalışmaları için sürdürülebilir uzun vadeli bir okuma platformu ihtiyacından doğmuştur. Çalışma yalnızca doğrudan bir kart geçişiyle sınırlı kalmamış; harici okuma zincirinin çalışmasını iyileştirmek amacıyla aygıt yazılımı yeniden düzenlenmiştir. Temel geliştirmeler arasında nanosaniye ölçekli sensör çıkışlarını yakalamak için optimize edilmiş asenkron aşırı örnekleme veri yolu, güncellenmiş gecikme ve paralelleştirme kaynakları, hedef karta uyarlanmış Ethernet arayüzü, sadeleştirilmiş slow-control yolu ve donanım destekli injection pulse üretim mekanizması yer almaktadır. FPGA uygulama sonuçları ve önceki KC705 tabanlı sistemle yapılan karşılaştırmalı ölçümler, yükseltilmiş aygıt yazılımını gerekli okuma davranışını korurken örnekleme kararlılığını, tam matris yanıt uniformluğunu ve tarama süresini iyileştirdiğini doğrulamaktadır. Temsili analog tarama ölçümlerinde, yükseltilmiş okuma sistemi tam piksel matrisi boyunca okuma kaynaklı uzamsal düzensizlikleri güçlü biçimde azaltmış; beklenen yanıtın yüzde 5'i içinde ölçülen piksel oranı yaklaşık yüzde 77'den yüzde 100'e yakın bir değere yükselmiştir. İyileştirilmiş enjeksiyon kontrol mekanizması ayrıca tarama sürelerini yaklaşık iki kata kadar azaltmaktadır; bu durum özellikle uzun eşik tarama ölçümleri için önemlidir. Ek olarak, ATLAS ITk referanslı yüksek oran simülasyon çalışması MALTA2 yanıtını gerçekçi doluluk referans koşullarıyla ilişkilendirmektedir. Genel olarak, yükseltilmiş UltraScale okuma sistemi devam eden MALTA2 çalışmaları ve gelecekteki MALTA okuma geliştirmeleri için daha güvenilir, daha hızlı ve sürdürülebilir bir yapı sağlamaktadır.

Mayıs 2026, 61 sayfa

Anahtar Kelimeler: Parçacık izleme dedektörleri, monolitik aktif piksel sensörleri, dedektör okuma, FPGA aygıt yazılımı, veri toplama, ATLAS ITk referanslı çalışmalar

ABSTRACT

M.Sc. Thesis

EVOLUTION OF THE MALTA2 READOUT FIRMWARE TO ULTRASCALE AND EVALUATION FOR ITK REPLACEMENT IN ATLAS

Fuat Kerem IŞIK

Ankara University
Institute of Accelerator Technologies
Department of Accelerator Technologies

Supervisor: Prof. Dr. İlkay TÜRK ÇAKIR
Co-Supervisor: Dr. Carlos A. SOLANS SANCHEZ

This thesis presents the evolution of the MALTA2 monolithic active pixel sensor readout firmware to a Kintex UltraScale architecture using the ALINX AXKU040 development board. The work was motivated by the need for a sustainable long-term readout platform for MALTA2 laboratory measurements, beam-test applications, and ATLAS ITk-based high-rate studies, while preserving compatibility with the established DAQ software interface. Rather than being a direct board migration, the firmware was revised to improve the external readout chain. The main developments include an optimized asynchronous oversampling data path for nanosecond-scale sensor outputs, updated delay and deserialization resources, an Ethernet interface adapted to the target board, a simplified slow-control path, and a hardware-assisted injection pulse-generation scheme. FPGA implementation results and comparative measurements with the previous KC705-based system confirm that the upgraded firmware preserves the required readout behavior while improving sampling stability, full-matrix response uniformity, and scan execution time. In representative analog-scan measurements, the upgraded readout reduces readout-related spatial non-uniformities across the full pixel matrix, increasing the fraction of pixels measured within 5% of the expected response from about 77% to nearly 100%. The improved injection-control scheme reduces scan execution times by up to approximately a factor of two, which is especially important for extended threshold-scan measurements. An additional ATLAS ITk-based high-rate simulation study relates the MALTA2 response to realistic occupancy benchmark conditions. Overall, the upgraded UltraScale readout establishes a more reliable, faster, and maintainable framework for continued MALTA2 operation and future MALTA readout developments.

May 2026, 61 pages

Keywords: Particle tracking detectors, monolithic active pixel sensors, detector readout, FPGA firmware, data acquisition, ATLAS ITk-based studies

FOREWORD AND ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor, Prof. Dr. İlkay TÜRK ÇAKIR, for her guidance, support, and valuable feedback throughout my thesis study. Her academic guidance and encouragement were important during the development and completion of this work.

I would like to express my deepest gratitude to my co-supervisor, Dr. Carlos A. SOLANS SANCHEZ, for his continuous guidance, patience, technical support, and valuable feedback throughout this thesis work. His careful reviews and constructive suggestions greatly helped me improve the quality of this study.

I am also grateful to Mehmet Burak AYKENAR for his valuable support in teaching me the fundamentals of FPGA design and digital electronics, which helped me build the technical foundation required for the firmware studies presented in this thesis.

I am grateful to the professors and academic staff of the Institute of Accelerator Technologies for their academic guidance, courses, and the institutional resources provided during my graduate education.

I am also grateful to the MALTA team and to all colleagues who contributed to the discussions, measurements, firmware development, and testing activities related to this thesis.

I acknowledge support from the Scientific and Technological Research Council of Türkiye (TÜBİTAK) under the 1001 Scientific and Technological Research Projects Funding Program, project no. 124F268.

Finally, I would like to thank my family and loved ones for their continuous support, patience, and encouragement throughout my studies.

Fuat Kerem IŞIK
Ankara, May 2026

TABLE OF CONTENTS

ÖZET ..... **i**

ABSTRACT ..... **ii**

FOREWORD AND ACKNOWLEDGEMENTS ..... **iii**

SYMBOLS AND ABBREVIATIONS ..... **vi**

LIST OF FIGURES ..... **vii**

LIST OF TABLES ..... **ix**

1. INTRODUCTION **1**

1.1 Accelerator Technologies and Particle Detection **1**

1.2 CERN, the LHC and the ATLAS Experiment **1**

1.3 Tracking Detectors and the ATLAS Inner Tracker Upgrade **2**

1.4 Pixel Detector Technologies and MALTA Sensors **4**

1.5 Motivation of this Thesis **5**

1.6 Aim and Scope of the Thesis **6**

1.7 Thesis Organization **7**

2. LITERATURE REVIEW **8**

2.1 Particle Tracking in Accelerator Experiments **8**

2.2 Hybrid Pixel Detectors **8**

2.3 Monolithic Active Pixel Sensors **10**

2.4 MALTA Family of MAPS **11**

2.5 FPGA-Based Readout Systems **12**

3. MATERIALS AND METHODS **14**

3.1 Target Hardware Platforms **14**

3.2 Readout Architecture Overview **16**

3.3 Ethernet Communication Through IPbus **17**

3.4 Injection and Pulse Generation **21**

3.5 Signal Conditioning and Alignment **24**

3.6 Asynchronous Oversampling **27**

3.7 Hit Reconstruction **29**

3.8 FIFO Chain **32**

3.9 Slow Control **33**

4. RESULTS **40**

4.1 Implementation Summary **40**

4.2 Tap Calibration **41**

4.3 Analog Scan **42**

4.4 Threshold Scan **45**

5. MALTA2 HIGH-RATE SIMULATION STUDY WITH AN ATLAS ITK-BASED REFERENCE **47**
5.1 Simulation Methodology **47**
5.2 ATLAS ITk-based Occupancy Benchmark **50**
5.3 Simulation Results **51**
5.4 Interpretation of High-rate Effects **53**
6. DISCUSSION AND CONCLUSION **55**
REFERENCES **59**

SYMBOLS AND ABBREVIATIONS

Symbols

%	Percentage
$V_H - V_L$	Injection-pulse amplitude
GHz	Gigahertz
MHz	Megahertz
ns	Nanosecond
ps	Picosecond

Abbreviations

ATLAS	A Toroidal LHC ApparatuS experiment
AXKU040	ALINX Kintex UltraScale development board
CERN	European Organization for Nuclear Research
DAQ	Data acquisition
FIFO	First-in, first-out memory buffer
FMC	FPGA mezzanine card interface
FPGA	Field-programmable gate array
FSM	Finite-state machine
IDELAYE3	UltraScale input-delay primitive
IPbus	UDP/IP-based control and readout protocol
ISERDESE3	UltraScale input-deserializer primitive
ITk	ATLAS Inner Tracker
KC705	Xilinx Kintex-7 evaluation kit
LVDS	Low-voltage differential signaling
MALTA	Monolithic from ALICE To ATLAS
MMCM	Mixed-mode clock manager
RGMII	Reduced Gigabit Media Independent Interface
SERDES	Serializer/deserializer
SGMII	Serial Gigabit Media Independent Interface
TEMAC	Tri-Mode Ethernet MAC
UDP	User Datagram Protocol

LIST OF FIGURES

Figure 1.1 Layout of the ATLAS detector. Source: Ref. [1] 2
Figure 1.2 Schematic view of the ATLAS Inner Tracker upgrade. Source: Ref. [2] 3
Figure 2.1 Schematic representation of the ITkPix-V1 hybrid pixel module concept, where charge produced by an ionizing particle is guided through a bump bond to the ITkPix-V1 readout chip. Source: Ref. [3] 9
Figure 2.2 Representative cross-section of a depleted monolithic active pixel sensor concept with a small n-well collection electrode and an extended depletion region, illustrating the MAPS/DMAPS technology background relevant to MALTA2. Source: Ref. [4] 11
Figure 3.1 Xilinx Kintex-7 KC705 evaluation kit. Source: Ref. [5] 15
Figure 3.2 Kintex UltraScale AXKU040 development board. Source: Ref. [6] . . . 15
Figure 3.3 Readout architecture overview 17
Figure 3.4 Ethernet/IPbus communication between the host PC and FPGA registers 18
Figure 3.5 Comparison of the Ethernet interface adaptation on KC705 and AXKU040 20
Figure 3.6 Comparison of the previous and migrated injection-control schemes . 21
Figure 3.7 Software-side injection function used to prepare the injection command sequence 22
Figure 3.8 Firmware-side injection FSM used in the migrated implementation . . 23
Figure 3.9 Signal conditioning and alignment 25
Figure 3.10 Oversampling architecture 27
Figure 3.11 DDR sampling of the P and N signal paths 29
Figure 3.12 Pin detector logic 31
Figure 3.13 FIFO chain in the MALTA2 readout path 33
Figure 3.14 System-level overview of the MALTA2 slow-control path 34
Figure 3.15 FPGA implementation of the slow-control dataflow for MALTA2 configuration 35
Figure 3.16 Internal organisation of the MALTA2 slow-control configuration 36
Figure 4.1 Tap-calibration maps after timing adjustment with 90 injected hits. The vertical axis shows the readout-channel index, the horizontal axis shows the position within the oversampled hit pattern, and the color scale indicates the number of recorded hits at each sampling position 41
Figure 4.2 Two-dimensional analog-scan response maps obtained with 90 injected pulses per pixel. The horizontal axis corresponds to the pixel column, the vertical axis corresponds to the pixel row in the MALTA2 matrix, and the color scale indicates the measured hit count per pixel . 42

Figure 4.3 Analog-scan execution-time distribution over 1000 repeated runs for the KC705-based and AXKU040-based implementations. The distributions show the scan duration measured for each run and are used to compare both the average execution time and the run-to-run timing stability of the two readout implementations 44

Figure 4.4 Threshold distributions for the same MALTA2 sample under the same threshold setting. The horizontal axis shows the extracted threshold in units of $V_H - V_L$, while the vertical axis shows the number of pixels in each threshold bin..... 45

Figure 5.1 Simulation workflow used to evaluate the MALTA2 response under ATLAS ITk-derived occupancy and beam-rate conditions 48

Figure 5.2 Simulated MALTA2 efficiency as a function of threshold for ATLAS ITk-derived occupancy conditions in different detector regions using a 30 μm epitaxial layer. The ITk region labels define the occupancy inputs used for the replacement-feasibility evaluation and do not indicate that MALTA2 is implemented in the baseline ATLAS ITk detector 52

Figure 5.3 Simulated MALTA2 tracking efficiency as a function of threshold for different beam-rate conditions using 120 GeV protons and a 30 μm epitaxial layer 53

LIST OF TABLES

Table 3.1 Differences between the KC705 and the AXKU040 development boards 16
Table 3.2 IPbus register map of the readout firmware 19
Table 3.3 Comparison of software-side control traffic for a representative injection request 24
Table 3.4 IDELAY implementation comparison 26
Table 3.5 ISERDES implementation comparison 28
Table 3.6 Reference-pattern mapping in the edge detector 30
Table 3.7 MALTA2 data-word field definition 32
Table 3.8 Slow control registers 36
Table 4.1 Post-route timing summary of the implemented AXKU040 firmware . 40
Table 4.2 Post-implementation resource utilization of the migrated AXKU040 firmware 41
Table 4.3 Analog-scan result summary for the same MALTA2 sample with 90 injected pulses 43
Table 5.1 Main inputs and outputs of the MALTA2 high-rate simulation study . 49

1. INTRODUCTION

1.1 Accelerator Technologies and Particle Detection

Accelerator technologies are not limited to the acceleration of charged particles. They also include the experimental systems required to produce, observe, measure, and interpret the products of particle interactions. In accelerator-based research, detector systems are therefore essential components of the experimental chain. After particles are accelerated and brought into collision, or directed onto a target, the physical information of interest is obtained from the signals produced in different detector layers [7, 8].

Tracking detectors are particularly important in collider experiments because they measure the trajectories of charged particles close to the interaction point. From these trajectories, quantities such as direction, momentum, electric charge, and production vertex can be reconstructed. In the presence of a magnetic field, the curvature of a charged-particle track provides the basis for momentum and charge-sign determination [9]. Therefore, the performance of a tracking detector depends not only on the sensor technology itself, but also on the readout electronics, timing alignment, data acquisition system, and firmware infrastructure that convert detector signals into usable experimental data.

1.2 CERN, the LHC and the ATLAS Experiment

CERN hosts the Large Hadron Collider (LHC), a high-energy proton-proton collider designed to search fundamental particle interactions at the energy frontier. One of the major experiments at the LHC is ATLAS, a general-purpose particle detector designed to study a wide range of physics processes, including precision Standard Model measurements and searches for new phenomena [9]. The ATLAS detector is composed of several sub-detector systems arranged in layers around the collision point. These systems are integrated to enable the reconstruction of particles produced in each collision event. The layout of the ATLAS detector shown in **Figure 1.1** was reproduced from Ref. [1].

The innermost region of ATLAS is dedicated to charged-particle tracking. Since this re-

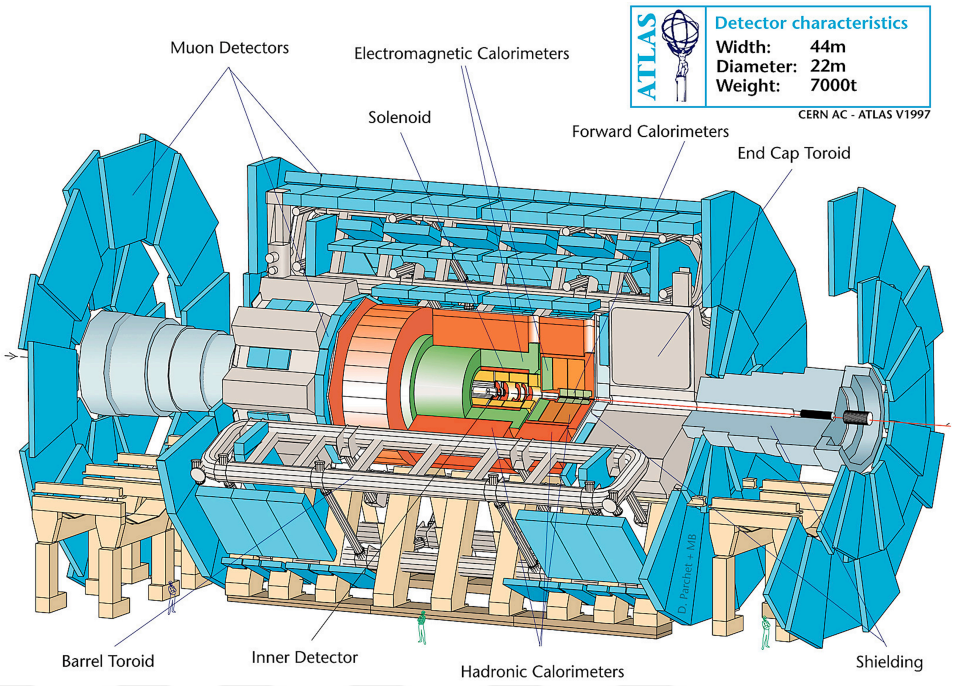


Figure 1.1 Layout of the ATLAS detector. Source: Ref. [1]

gion is closest to the proton-proton interaction point, it is exposed to high particle densities and significant radiation levels. The ATLAS Inner Detector contains pixel detectors, silicon microstrip detectors, and transition radiation tracking elements [9, 10]. The accurate reconstruction of charged-particle tracks is essential for primary-vertex reconstruction, secondary-vertex identification, flavour tagging, and many physics analyses.

1.3 Tracking Detectors and the ATLAS Inner Tracker Upgrade

The operating conditions of the High-Luminosity LHC (HL-LHC) require major upgrades to the tracking detectors of ATLAS. The HL-LHC is designed to increase the instantaneous and integrated luminosity of the LHC, thereby increasing the number of proton-proton interactions and the amount of data delivered to the experiments [11]. In the HL-LHC operating scenario, the ATLAS detector is expected to experience up to 200 simultaneous proton–proton interactions per bunch crossing. The bunch-crossing interval is 25 ns, corresponding to the 40 MHz bunch-crossing frequency of the LHC [12, 13]. These conditions lead to higher particle occupancies, larger radiation doses, increased bandwidth demand, and more demanding readout requirements for the innermost tracking systems.

For this reason, the present ATLAS Inner Detector will be replaced by the Inner Tracker, known as ITk, for the Phase-II upgrade. The ITk is designed as an all-silicon tracking detector, with pixel detectors in the innermost region and silicon strip detectors at larger radii [14, 15]. The ITk detector system is required to cope with increased pile-up, data rates, and radiation levels while maintaining or improving the tracking performance of ATLAS [14]. The schematic view of the ATLAS Inner Tracker upgrade shown in **Figure 1.2** was reproduced from Ref. [2].

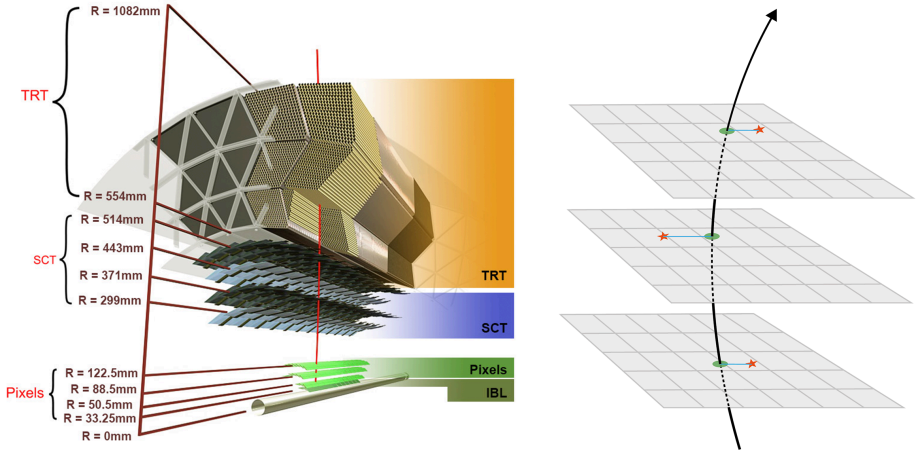


Figure 1.2 Schematic view of the ATLAS Inner Tracker upgrade. Source: Ref. [2]

The ITk upgrade is motivated by the need for higher granularity, improved radiation tolerance, faster readout, and reduced material budget. Pixel detectors are especially critical in the innermost layers, where the track density is highest and the precision requirement is strongest. These requirements have driven extensive research and development on advanced pixel detector technologies, including both hybrid pixel detectors and monolithic active pixel sensors [14, 3, 4].

Although MALTA2 is not part of the baseline ATLAS ITk detector, the ITk environment provides a relevant reference case for evaluating whether a monolithic pixel sensor concept such as MALTA2 could be considered an alternative or replacement candidate under demanding tracking-detector conditions. In this thesis, the term replacement is therefore used in an evaluation sense: the study does not claim that MALTA2 is implemented in the ATLAS ITk detector but investigates whether its response remains compatible with ITk-derived occupancy and timing requirements.

In particular, the local hit density expected in the ITk is used as a benchmark for estimating

how many particles may cross a MALTA2 sensor area within a single 25 ns bunch crossing. This allows the MALTA2 response to be tested against realistic high-rate conditions and provides a basis for identifying whether its sensor and readout architecture would be suitable, limited, or unsuitable for an ITk-replacement scenario.

1.4 Pixel Detector Technologies and MALTA Sensors

Hybrid pixel detectors have been successfully used in high-energy physics experiments because they allow the sensor and the readout electronics to be optimized separately. In this approach, the sensor is connected to a dedicated front-end readout chip, typically through bump bonding. This separation provides high radiation tolerance, high hit-rate capability, and a dedicated optimization of the sensor and readout ASIC. However, the hybrid approach also increases material budget, production complexity, and cost, especially for large-area tracking systems [9, 14, 3].

Monolithic active pixel sensors provide an alternative approach in which the sensing element and at least part of the readout electronics are integrated in the same silicon substrate. This can reduce material, simplify module construction, and improve scalability for large-area tracking detectors [4, 16]. However, monolithic sensors intended for collider environments must also meet strict requirements on radiation tolerance, hit-rate capability, timing performance, low noise, low power consumption, and readout reliability [4, 17].

The MALTA sensor family was developed in this context as a depleted monolithic active pixel sensor technology in TowerJazz/Tower Semiconductor 180 nm CMOS imaging technology. MALTA sensors combine a small collection electrode with a fast, low-power front-end and an asynchronous readout architecture [18, 16]. The small collection electrode reduces input capacitance, which is beneficial for noise and power performance, while process modifications and depletion of the sensitive volume are used to improve charge collection and radiation tolerance [4, 19]. Earlier MALTA developments were studied as an option for the outer pixel layers of the upgraded ATLAS ITk pixel detector [18, 16].

MALTA2 is a later prototype in this sensor family. It incorporates front-end and sensor-design improvements aimed at reducing noise, improving charge collection, and supporting operation at low threshold [17, 19]. MALTA2 has also been characterized in terms

of timing response and charge calibration, both of which are important for interpreting threshold scans, injection-based measurements, and beam-test performance [20, 21]. Therefore, stable and efficient readout operation is required not only for operating the sensor, but also for evaluating its performance consistently in laboratory characterization and test-beam studies.

1.5 Motivation of this Thesis

The work presented in this thesis focuses on the readout firmware required to operate and characterize the MALTA2 sensor. Although the sensor technology is the central element of the detector system, its performance can only be evaluated reliably if the external readout system captures the asynchronous sensor outputs correctly, preserves timing information, and provides a stable interface to the data acquisition software. This is particularly important for the MALTA architecture because hit information is propagated asynchronously from the pixel matrix to the periphery, placing strict requirements on timing alignment, oversampling, buffering, and data transfer in the external FPGA readout chain [16].

The original MALTA2 readout firmware was implemented on a Xilinx Kintex-7 KC705 evaluation board. Since this platform reached end-of-life, a replacement platform became necessary for long-term MALTA2 laboratory measurements and telescope beam-test applications. The ALINX AXKU040 Kintex UltraScale development board was selected as the target platform for the migrated readout implementation.

The motivation for upgrading the MALTA2 readout firmware is therefore not limited to replacing an obsolete FPGA board. In asynchronous monolithic pixel sensors such as MALTA2, the external readout system is part of the measurement chain that determines whether the sensor response can be captured, reconstructed, and characterized reliably. Firmware-level timing alignment, stable oversampling, buffering, and DAQ compatibility directly affect the quality and reproducibility of laboratory characterization measurements. This motivation becomes even more relevant in view of future MALTA-family developments. Published studies on MALTA3 and future MALTA architectures show that the development effort is not limited to sensor-process and front-end improvements, but also includes changes in the digital periphery, synchronization memory, timing architecture,

and high-speed data-transfer concepts [22, 23]. In this context, the present thesis provides a behavior-preserving and more stable FPGA readout platform for MALTA2, while also establishing a basis for future studies of MALTA readout architectures.

The main objective of this thesis is to perform a behavior-preserving migration of the MALTA2 readout firmware from the KC705 platform to the AXKU040 platform. The migration is required to preserve the software-visible behavior of the system, including the sampling rate, timing conventions, register-level operation, and output data format. At the same time, the internal firmware implementation must be adapted to the UltraScale architecture and to the physical interfaces available on the new board.

1.6 Aim and Scope of the Thesis

The aim of this thesis is to develop, implement, and evaluate the migrated MALTA2 readout firmware on the AXKU040 platform. The scope includes the adaptation of the Ethernet communication interface, the implementation of the asynchronous oversampling structure using UltraScale FPGA resources, the preservation of the delay-calibration model, the revision of the injection pulse-generation mechanism, and the simplification of the slow-control path.

This thesis does not focus on the design of the MALTA2 sensor itself. Instead, it focuses on the firmware infrastructure required to read out the sensor reliably and to support characterization measurements. Therefore, the results presented in this thesis should be interpreted as improvements in readout stability, scan reproducibility, matrix-wide response uniformity, and characterization workflow, rather than as a redesign of the intrinsic sensor architecture. The migrated firmware is evaluated through FPGA implementation results and comparative measurements with the previous KC705-based readout system, including tap calibration, analog scans, and threshold scans.

In addition to the firmware migration and laboratory measurements, the thesis evaluates the response of MALTA2 under ATLAS ITk-derived high-rate occupancy conditions in order to assess its limitations with respect to a possible ITk-replacement scenario.

1.7 Thesis Organization

The remainder of this thesis is organized as follows. Chapter 2 gives the background on particle tracking detectors, pixel detector technologies, monolithic active pixel sensors, the MALTA sensor family, FPGA-based readout systems, and readout-related efficiency challenges in future MALTA developments. Chapter 3 describes the materials and methods used in the firmware migration, including the MALTA2 sensor, the FPGA platforms, the readout architecture, Ethernet/IPbus communication, injection control, signal alignment, asynchronous oversampling, hit reconstruction, FIFO-based data transfer, and slow-control implementation. Chapter 4 presents and discusses the implementation and measurement results obtained with the migrated AXKU040 firmware. Chapter 5 presents the ATLAS ITk-based high-rate simulation study used to evaluate the feasibility and limitations of MALTA2 in a possible ITk-replacement scenario. Chapter 6 summarizes the main conclusions of the thesis and outlines the relevance of the work for future MALTA readout developments.

2. LITERATURE REVIEW

2.1 Particle Tracking in Accelerator Experiments

Particle tracking is one of the fundamental measurement tasks in accelerator-based experiments. When charged particles pass through a tracking detector, they leave localized signals in successive detector layers. By combining these spatial measurements, the trajectory of the particle can be reconstructed. In the presence of a magnetic field, the curvature of the reconstructed track provides information about the particle momentum and electric charge [7, 9].

In collider experiments, tracking detectors are placed close to the interaction point in order to measure the initial part of the charged-particle trajectory. This makes them essential for primary-vertex reconstruction, secondary-vertex identification, momentum measurement, and flavour-tagging applications [9, 14]. The performance of a tracking system depends on spatial resolution, hit efficiency, timing performance, material budget, radiation tolerance, and the ability of the readout system to process detector signals reliably.

For the High-Luminosity LHC era, the tracking systems of the major LHC experiments must operate under higher particle occupancies, larger radiation doses, and increased data rates [11, 14]. These requirements have driven the development of advanced silicon tracking detectors, including both hybrid pixel detectors and monolithic active pixel sensors. In this context, the detector sensor, front-end electronics, digital periphery, readout firmware, and data-acquisition system must be considered together as parts of the same measurement chain.

2.2 Hybrid Pixel Detectors

Hybrid pixel detectors are a mature technology for high-energy physics tracking applications. In this approach, the sensor and the readout electronics are implemented as separate silicon components. The sensor layer is optimized for charge collection, while the front-end readout chip is optimized for amplification, digitization, buffering, triggering, and

data transmission. The two components are electrically connected pixel by pixel, typically through bump bonds [9, 3].This separation provides an important advantage: the sensor technology and the readout electronics can be developed and optimized independently. For this reason, hybrid pixel detectors are widely used in environments requiring high radiation tolerance, high hit-rate capability, and precise timing. However, the hybrid approach also introduces additional production complexity, cost, and material, mainly due to the bump-bonding process and the use of separate sensor and readout-chip layers [14, 3]. The ATLAS ITk Pixel detector is an example of a hybrid pixel detector system developed for the demanding conditions of the HL-LHC. In this architecture, the sensor and the readout chip are produced as separate silicon dies and connected during the hybridization process. The basic hybrid concept for the ITkPix-V1/RD53B readout chip shown in **Figure 2.1**, where charge generated in the sensor by an ionizing particle is transferred through a bump bond to the readout chip, was reproduced from Ref. [3].

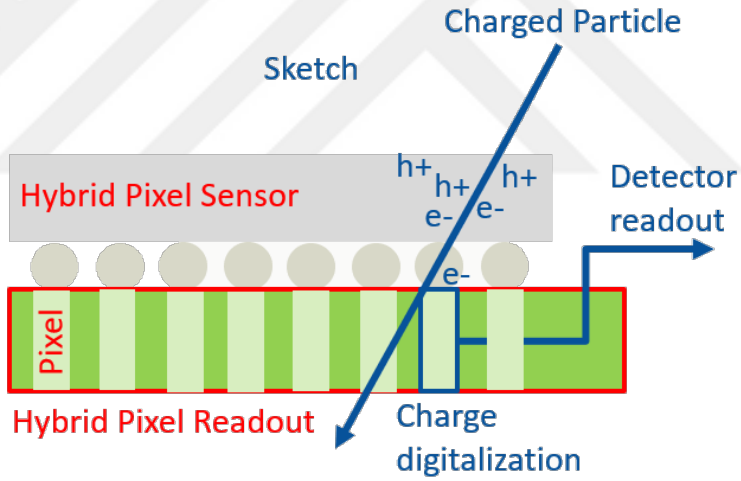


Figure 2.1 Schematic representation of the ITkPix-V1 hybrid pixel module concept, where charge produced by an ionizing particle is guided through a bump bond to the ITkPix-V1 readout chip. Source: Ref. [3]

The hybrid architecture is therefore highly suitable for the innermost regions of collider tracking detectors, where radiation levels and particle rates are most demanding. At the same time, its production complexity and material budget motivate the study of monolithic technologies for future large-area tracking detectors [14, 4].

2.3 Monolithic Active Pixel Sensors

Monolithic active pixel sensors (MAPS) provide an alternative detector concept in which the sensing volume and at least part of the readout electronics are integrated in the same silicon substrate. In this architecture, the need for bump bonding between a separate sensor and a separate readout chip is removed. This can reduce material budget, simplify detector module construction, and make large-area tracking systems more scalable [4, 16].

For particle-physics applications, monolithic sensors must satisfy demanding requirements. They must provide sufficiently fast charge collection, good radiation tolerance, low noise, low power consumption, and reliable readout. One important development direction is the use of depleted monolithic active pixel sensors with small collection electrodes. A small collection electrode can reduce input capacitance and therefore improve noise and performance of power. However, the sensor design must also provide efficient charge collection by drift in order to improve timing performance and radiation tolerance [4, 17].

Snoeys et al. proposed a process modification for CMOS monolithic active pixel sensors to enhance depletion, timing performance, and radiation tolerance. The modification was developed in the context of 180 nm CMOS imaging technology and is directly relevant to the class of depleted monolithic pixel sensors used in high-energy physics. The representative depleted monolithic active pixel sensor concept shown in **Figure 2.2**, based on a small n-well collection electrode and an extended depletion region, was reproduced from Ref. [4]. This type of structure provides the technological background for MAPS/DMAPS developments such as the MALTA sensor family [4, 16].

This type of technology is important for future tracking detectors because it combines the low-material and integration advantages of monolithic sensors with improved charge-collection and radiation-tolerance properties. These developments provide the technological background for depleted monolithic active pixel sensor families such as MALTA [4, 16].

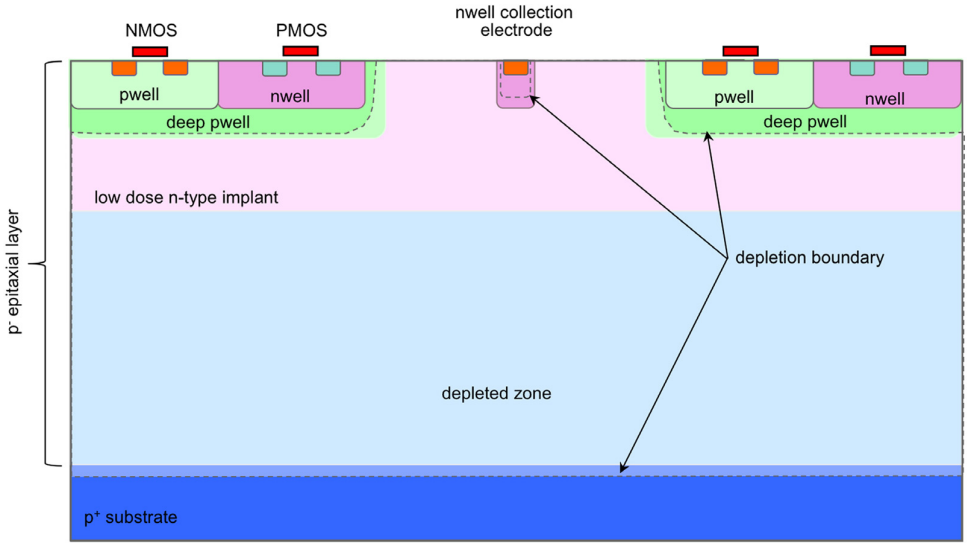


Figure 2.2 Representative cross-section of a depleted monolithic active pixel sensor concept with a small n-well collection electrode and an extended depletion region, illustrating the MAPS/DMAPS technology background relevant to MALTA2. Source: Ref. [4]

2.4 MALTA Family of MAPS

The MALTA sensor family was developed as a depleted monolithic active pixel sensor technology for high-energy physics tracking applications. MALTA sensors are implemented in a modified TowerJazz/Tower Semiconductor 180 nm CMOS imaging process and use a small collection electrode together with a fast front-end and an asynchronous readout architecture [18, 16]. Earlier MALTA developments were studied in the context of the outer pixel layers of the upgraded ATLAS ITk pixel detector [18].

The asynchronous readout architecture is one of the important features of the MALTA family. Instead of distributing a high-speed clock through the full pixel matrix, hit information is propagated from the matrix to the periphery using asynchronous signal transmission [16]. This approach can reduce in-matrix digital activity and power consumption, but it also places strict requirements on the external readout system. The readout firmware must capture fast asynchronous output signals, align their timing, reconstruct hit information, and transfer the resulting data to the data acquisition software.

MALTA2 is a later prototype in the MALTA sensor family. It incorporates front-end and sensor-design improvements intended to reduce noise, improve charge collection, and support operation at low threshold [17, 19]. These developments are important because low

threshold operation, efficient charge collection, and stable timing response are closely connected to hit detection performance in depleted monolithic pixel sensors.

Several studies have focused on the characterization of MALTA and MALTA2 sensors. Depletion-depth studies have investigated the charge-collection properties of MALTA2 as a depleted monolithic active pixel sensor [19]. Timing-characterization studies have examined the response of MALTA and MALTA2 detectors using a micro X-ray source [20]. Charge-calibration studies have provided an experimental basis for interpreting injection-based measurements and threshold scans in terms of deposited or injected charge, while charge-collection parameterization studies support the simulation-oriented interpretation of the MALTA2 charge-response behaviour [21, 24].

The MALTA architecture has also continued to evolve toward future sensor generations. Published studies on MALTA3 and future MALTA developments describe architectural concepts such as improvements in the digital periphery, synchronization memory, timing architecture, and data-transfer scheme [22, 23]. These developments show that the performance of MALTA-type detectors depends not only on the sensing volume and front-end design, but also on the organization of the digital readout path and its ability to transfer hit information efficiently.

Therefore, MALTA2 is not only a sensor prototype, but also an experimental platform that requires a reliable readout and characterization chain. Laboratory measurements such as analog scans, threshold scans, injection studies, and timing studies depend on the stability of both the sensor response and the external readout system. For this reason, the FPGA firmware and DAQ interface are critical parts of the experimental setup. Without a reliable firmware and DAQ interface, the sensor performance cannot be measured or compared consistently.

2.5 FPGA-Based Readout Systems

Field-programmable gate arrays are commonly used in detector readout systems because they provide configurable high-speed digital logic, precise timing control, parallel data processing, and flexible communication interfaces. In pixel detector test systems, an FPGA can receive fast detector outputs, perform deserialization and timing alignment,

buffer event data, and communicate with the host computer through a data acquisition protocol.

For MALTA2, the FPGA readout firmware must handle several tasks simultaneously. It receives the sensor output through differential LVDS lines, aligns the incoming signals using programmable delay elements, oversamples the asynchronous data, reconstructs valid hit patterns, stores the resulting event words in FIFO buffers, and exposes control and readout registers to the software through IPbus. In parallel, the firmware provides a slow-control path for configuring the sensor and an injection-control mechanism for reproducible laboratory measurements.

The use of IPbus provides an Ethernet-based control and readout interface between the host computer and the FPGA register space [25]. This is useful for laboratory characterization systems because control registers, status registers, delay settings, monitoring registers, and data FIFOs can be accessed from the DAQ software through a common communication layer. In the MALTA2 readout system, this interface is used both for configuration and for retrieving the reconstructed MALTA data words.

The migration studied in this thesis is therefore not only a change of FPGA board. It is a firmware migration in which the software-visible behavior of the readout system must be preserved while the internal implementation is adapted to a different FPGA architecture. This includes the transition from Kintex-7 resources to Kintex UltraScale resources, the adaptation of the Ethernet interface, and the redesign or modification of internal blocks such as asynchronous oversampling, injection control, signal alignment, FIFO-based buffering, and slow control. The following chapter describes these materials and methods in detail.

3. MATERIALS AND METHODS

In this section, the materials, firmware architecture, and implementation method used for the migration of the MALTA2 readout firmware from the KC705 platform to the AXKU040 platform are described. The objective of this work was not to redesign the complete MALTA2 data-acquisition chain, but to migrate the firmware to a maintained and cost-effective FPGA platform while preserving the software-visible behavior required by the existing MALTA2 DAQ software.

The work was carried out on the MALTA2 readout system, in which the sensor outputs are connected to the FPGA through the FMC-HPC interface. In the readout data path, 37 LVDS output pairs from the chip are received, signal conditioning and delay alignment are performed, the asynchronous hit pulses are oversampled, the MALTA2 data word is reconstructed, and the resulting event data are transferred to the host computer through the Ethernet/IPbus interface. In parallel, the slow-control path is used to configure the MALTA2 chip by transferring the configuration word from the host software to the chip through the FPGA.

3.1 Target Hardware Platforms

The MALTA2 readout was originally implemented on a Xilinx Kintex-7 KC705 evaluation kit, as shown in **Figure 3.1**. This platform was used because it provided the FMC, Ethernet, and Kintex-7 FPGA resources required by the previous MALTA2 readout system, while also being a generally available and relatively low-cost evaluation platform at the time of the original implementation. A Kintex-7 XC7K325T-2FFG900C FPGA with 203,800 LUTs [26], ISERDESE2 and IDELAYE2 primitives, one HPC FMC connector, one LPC FMC connector, and a Gigabit Ethernet port are provided on this board. Since this platform was no longer suitable as a long-term maintained solution for future MALTA2 measurements, a replacement for the MALTA2 readout became necessary.

The replacement platform considered in this work is the ALINX AXKU040 development board, as shown in **Figure 3.2**. This board was selected because it provides a Kintex UltraScale FPGA, an FMC-HPC connector, Gigabit Ethernet connectivity, and a lower-cost

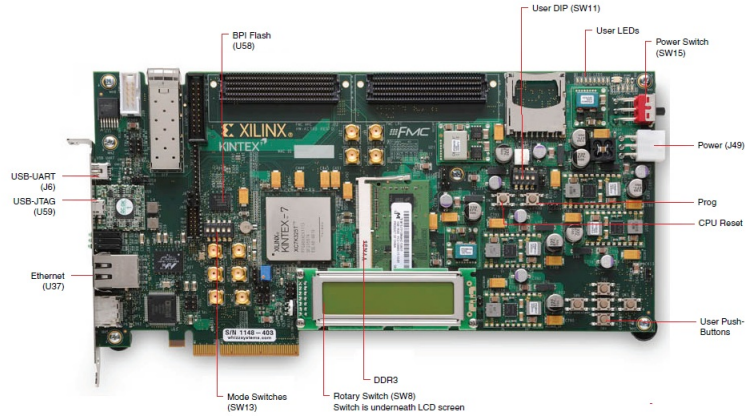


Figure 3.1 Xilinx Kintex-7 KC705 evaluation kit. Source: Ref. [5]

maintained platform suitable for continuing MALTA2 laboratory measurements and telescope beam-test applications. The board is based on a Kintex UltraScale XCKU040-2FFVA1156I FPGA with 242,400 LUTs [27], and ISERDESE3 and IDELAYE3 primitives are used in the migrated implementation. The main objective of this migration was to achieve full functionality on the AXKU040 board without any software-visible change in the readout behavior. In particular, the register map, sampling rate, timing behavior, and output data format were preserved. With this compatibility-first design policy, the existing DAQ software and surrounding firmware blocks could continue to operate without modification.

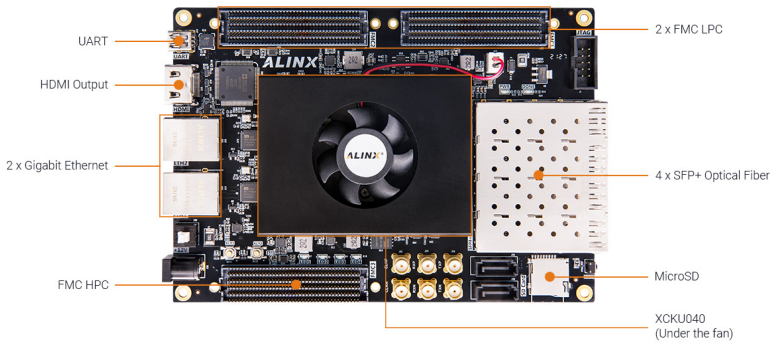


Figure 3.2 Kintex UltraScale AXKU040 development board. Source: Ref. [6]

The main differences between the KC705 and AXKU040 development boards are summarized in **Table 3.1**. The timing characteristics relevant to the SERDES and input-delay resources were taken from the Kintex-7 and Kintex UltraScale data sheets [28, 29].

Table 3.1 Differences between the KC705 and the AXKU040 development boards

Development Board	**KC705**	**AXKU040**
FPGA	XC7K325T-2FFG900C	XCKU040-2FFVA1156I
LUTs	203,800	242,400
ISERDESE version	ISERDESE2	ISERDESE3
ISERDESE speed	1.25 Gb/s	1.30 Gb/s
IDELAYE version	IDELAYE2	IDELAYE3
IDELAYE number of bits	5-bit	9-bit
Number of HPC FMC	1	1
Number of LPC FMC	1	2
Number of Ethernet ports	1	2
Number of user SMAs	4	0
Number of GTX SMAs	6	6

3.2 Readout Architecture Overview

The architecture of the MALTA2 readout firmware after its migration from the KC705 platform to the AXKU040 board is described in this subsection. The readout path starts from the 37 LVDS output lines of the sensor, which enter the FPGA through the FMC-HPC connector. Inside the FPGA, the signals are first conditioned and phase-aligned, then oversampled and processed to identify valid hit patterns. The resulting hit data are buffered and made available to the DAQ system through the Ethernet/IPbus interface. In parallel, a separate slow-control path is used to configure the sensor registers from the host software. The migrated MALTA2 firmware is organized into four main blocks, denoted in **Figure 3.3** as Slow Control, Asynchronous Oversampling, Core, and IPbus. A simplified overview of the overall structure is shown in the figure.

The MALTA2 chip is connected to the FPGA through the FMC interface. In the Slow Control block, configuration data are transferred from the host through IPbus, buffered in FIFOs, and serialized before being sent to the chip. In the migrated implementation described in this thesis, the slow-control path is used as a unidirectional configuration path from the DAQ software to the MALTA2 chip; slow-control readback is reserved in the register map but is not implemented in the current firmware version.

In the readout data path, the sensor outputs arrive at the FPGA as 37 LVDS pairs. Within

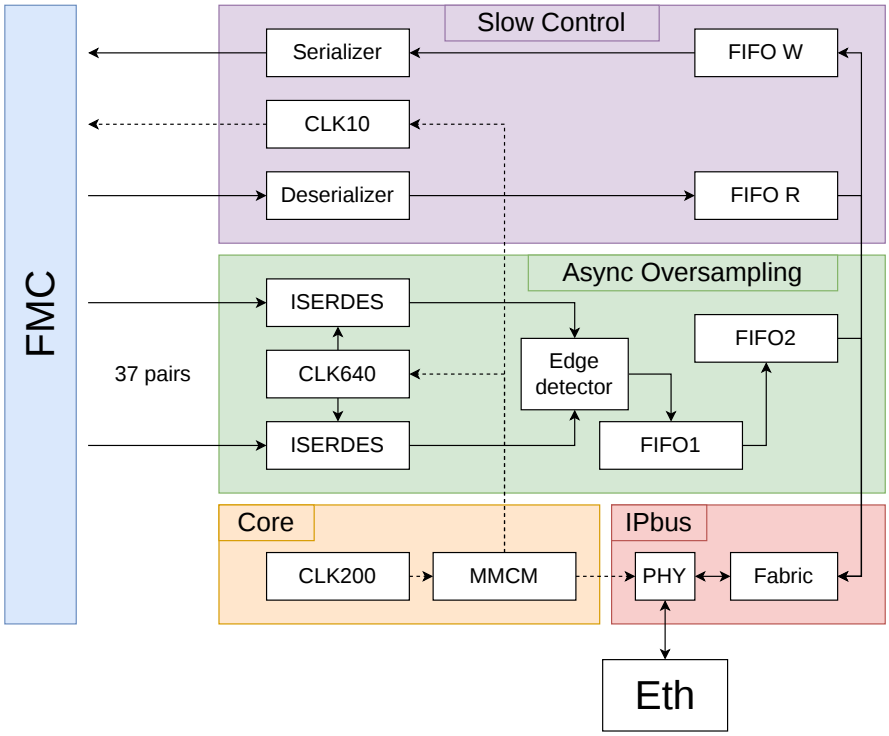


Figure 3.3 Readout architecture overview

the Asynchronous Oversampling block, these high-speed signals are deserialized channel by channel using the FPGA SERDES resources, and the incoming serial data are converted into parallel samples suitable for processing in the FPGA logic. With this parallelization, high-throughput readout is enabled while the subsequent logic is allowed to operate within a operating frequency range inside the FPGA. The processed data are then transferred through the FIFO chain to the IPbus interface, from which they are read out by the host. The internal clocks required by these blocks are generated from a 200 MHz reference clock in the Core block using an MMCM. As indicated in **Figure 3.3**, the generated clocks are distributed to the Asynchronous Oversampling block, the Slow Control block, and the IPbus interface.

3.3 Ethernet Communication Through IPbus

Communication with the readout firmware is performed over Ethernet using IPbus, an Ethernet-based control and readout protocol commonly used for FPGA-based detector electronics [25]. On the host side, UDP transactions are sent by the MALTA2 DAQ sof-

tware over a 1 GbE full-duplex link to the FPGA board. As illustrated in **Figure 3.4**, the Ethernet MAC is implemented in the FPGA using the Xilinx TEMAC IP core [30]. The IPbus payload is decoded in the firmware and mapped to the internal register space, so that control, monitoring, calibration, and data-readout registers can be accessed by the DAQ software through standard read and write operations.

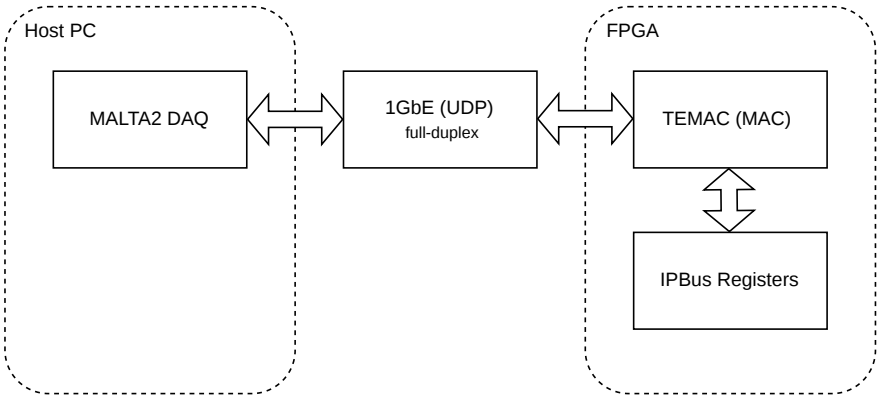


Figure 3.4 Ethernet/IPbus communication between the host PC and FPGA registers

The readout-related IPbus registers used in this work are listed in Table 3.2.
Readout data are retrieved through the FIFO-mapped register `IPBADDR_AOFIFO`, while configuration, monitoring, and calibration are performed through the corresponding control and status registers. In particular, the TAP-calibration interface is implemented through per-line write and readback registers, with one register assigned to each of the 37 input lines. Slow-control readback through `IPBADDR_SC_FIFO_R` is not implemented in the current version; therefore, the slow-control path is unidirectional from the DAQ software to the chip.

Table 3.2 IPbus register map of the readout firmware

Address	Register	R/W	Description
0	IPBADDR_VERSION	RO	Firmware version/build identifier (static).
1	IPBADDR_BUSY	RO	Status register reporting data-path conditions, including FIFO status information (fast FIFO in the 320 MHz domain and the FIFO crossing to the IPbus clock domain) to monitor potential backpressure/overflow during operation.
2	IPBADDR_CONTROL	RW	Control register used for chip-level actions such as injection and reset.
3	IPBADDR_AODELAY	RW	Programmable delay applied to the MALTA word stream before writing into the FIFO.
4	IPBADDR_AOMONI	RO	Monitoring interface for the 37 TAP-calibration lines, allowing per-line access during TAP calibration.
5	IPBADDR_AOSTATUS	RO	Number of words in the IPBADDR_AOFIFO register.
6	IPBADDR_AOFIFO	RO	FIFO-mapped register used to retrieve the final MALTA data words via IPbus reads.
7	IPBADDR_SC_FIFO_W	WO	Slow-control FIFO write port used to stream the slow-control configuration words.
8	IPBADDR_SC_FIFO_R	–	Reserved for slow-control readback (not implemented in the current version).
10	AO_ADDR_TAPW[0]	WO	Per-line TAP-calibration write registers for IDELAYE3 settings. One register is assigned to each of the 37 TAP lines.
11	AO_ADDR_TAPW[1]	WO	
12	AO_ADDR_TAPW[2]	WO	
⋮	⋮	⋮	
46	AO_ADDR_TAPW[36]	WO	
50	AO_ADDR_TAPR[0]	RO	Per-line TAP-calibration read registers for IDELAYE3 settings. One register is assigned to each of the 37 TAP lines.
51	AO_ADDR_TAPR[1]	RO	
52	AO_ADDR_TAPR[2]	RO	
⋮	⋮	⋮	
86	AO_ADDR_TAPR[36]	RO	

3.3.1 Migration from SGMII to RGMII

A board-specific change introduced during the migration concerns the Ethernet physical interface. In the original KC705-based implementation, the Ethernet path was implemented using the TEMAC IP core together with the PCS/PMA IP core, followed by an SGMII-based connection to the on-board Ethernet PHY [30, 31]. In the AXKU040-based implementation, this structure was replaced by a direct RGMII connection from the TEMAC IP core to the on-board Ethernet PHY [30, 32, 33]. This modification was required because the AXKU040 board uses a KSZ9031RNX Gigabit Ethernet PHY, and the board-level connection between the FPGA MAC and the PHY is based on the RGMII interface rather than the SGMII-based structure used in the KC705 implementation [32, 33].

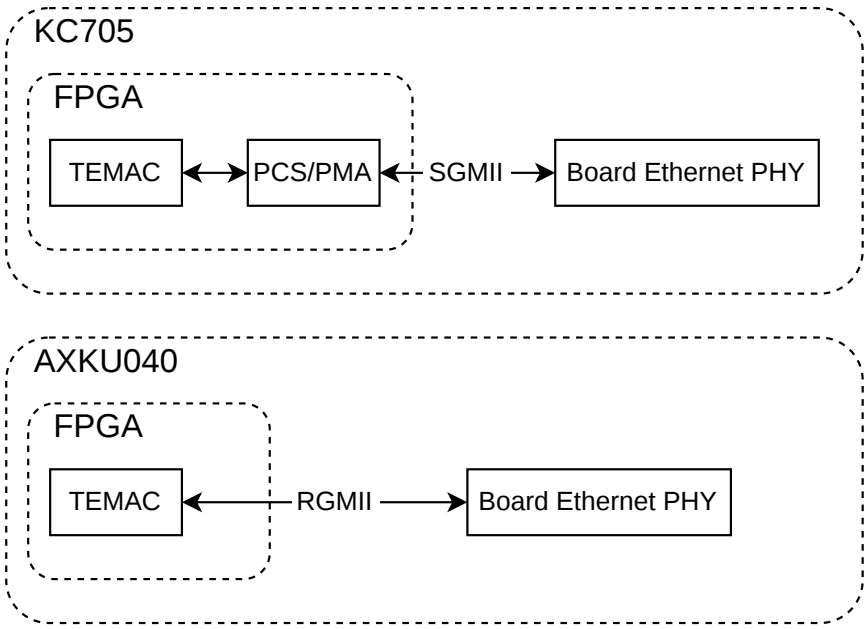


Figure 3.5 Comparison of the Ethernet interface adaptation on KC705 and AXKU040

As shown in **Figure 3.5**, the interface adaptation was implemented successfully, with no effect on the overall system functionality or on the communication model. From the software point of view, the firmware is still accessed by the DAQ software through UDP/IPbus over Gigabit Ethernet, with no software-visible difference in operation.

3.4 Injection and Pulse Generation

In addition to normal hit readout, injection-based testing is supported by the MALTA2 system. In this procedure, a controlled calibration pulse is applied to selected pixels or injection structures in order to stimulate the front-end response under reproducible laboratory conditions. The resulting response is then propagated through the chip and read out through the same output path used during normal operation. Injection-based measurements are commonly used to study threshold behavior and charge response in MALTA2 characterization, and charge-calibration studies provide the basis for relating the injection settings to an equivalent injected charge [21].

In the previous implementation, injection pulses were generated in a software-driven manner through repeated writes to the control register. Because these commands were sent from the host to the FPGA over the UDP/IPbus link, the scan execution time was strongly influenced by communication and protocol latency. In practice, multiple control transactions were required to generate a pulse sequence, which made injection-based scans significantly slower. Therefore, a repetitive timing-sensitive operation was paced by host communication rather than being handled locally inside the firmware.

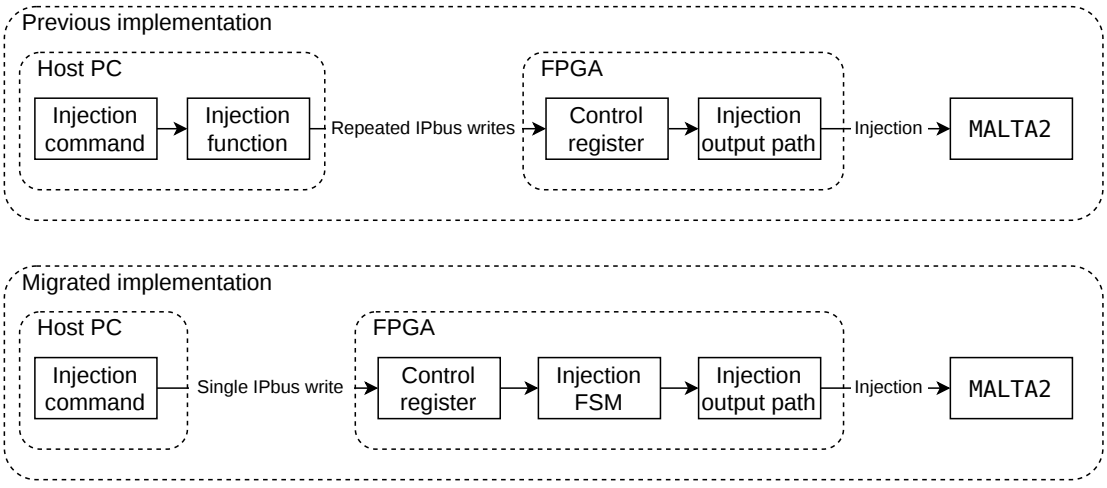


Figure 3.6 Comparison of the previous and migrated injection-control schemes

The previous and migrated injection-control schemes are compared in **Figure 3.6**. In the previous implementation, the pulse sequence was driven directly from the software side through repeated writes to `IPBADDR_CONTROL`. As a result, injection-based scan execution

depended strongly on UDP/IPbus transaction overhead rather than only on the internal operation of the readout system. In the migrated firmware, this control model was replaced by a hardware-assisted scheme in which only the injection request and the pulse count are specified by the software, while the corresponding pulse train is generated locally by a dedicated FSM inside the FPGA. Therefore, the repetitive timing-sensitive part of the injection procedure is shifted from the host software to the firmware, the number of required UDP/IPbus transactions is reduced, and pulse-by-pulse host intervention is removed. As a result, injection-based scan procedures can be executed significantly faster while the same software-level control interface is preserved. This change improves not only scan speed, but also the separation between high-level scan control in software and repetitive pulse-generation logic in firmware.

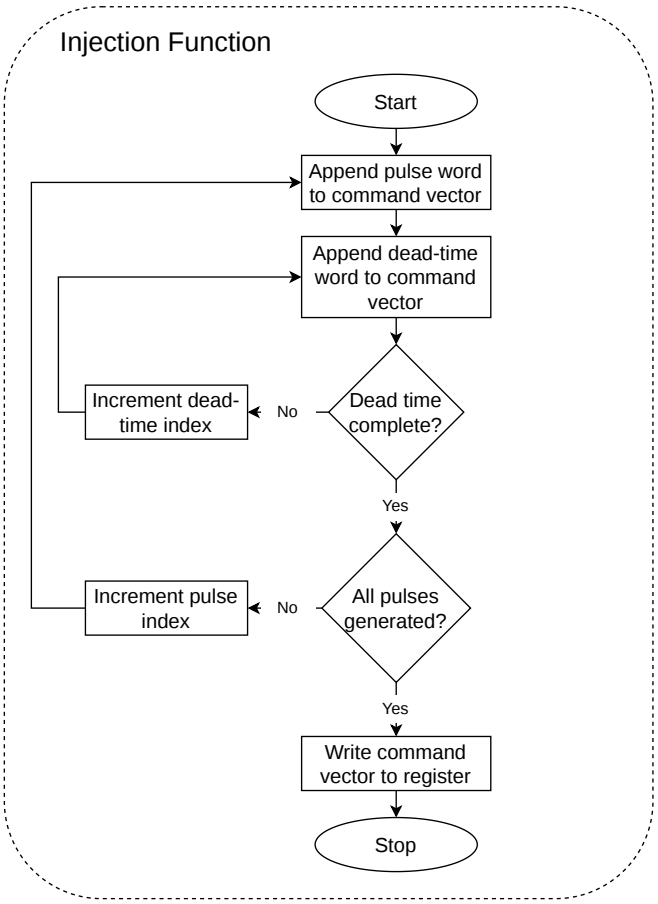


Figure 3.7 Software-side injection function used to prepare the injection command sequence

The software-side implementation of the injection procedure is shown in **Figure 3.7**. In the previous approach, the injection command sequence was constructed explicitly in sof-

tware, including the pulse and dead-time structure, and the resulting command pattern was written to the firmware registers through repeated IPbus transactions.

The firmware-side FSM introduced in the migrated implementation is shown in **Figure 3.8**. In this scheme, the repetitive pulse-generation procedure is handled locally in the FPGA. After the injection request and pulse count are received from the software, the corresponding pulse sequence is generated internally by the FSM. Therefore, the timing-sensitive repetitive task is shifted from software to firmware, and the communication overhead is significantly reduced.

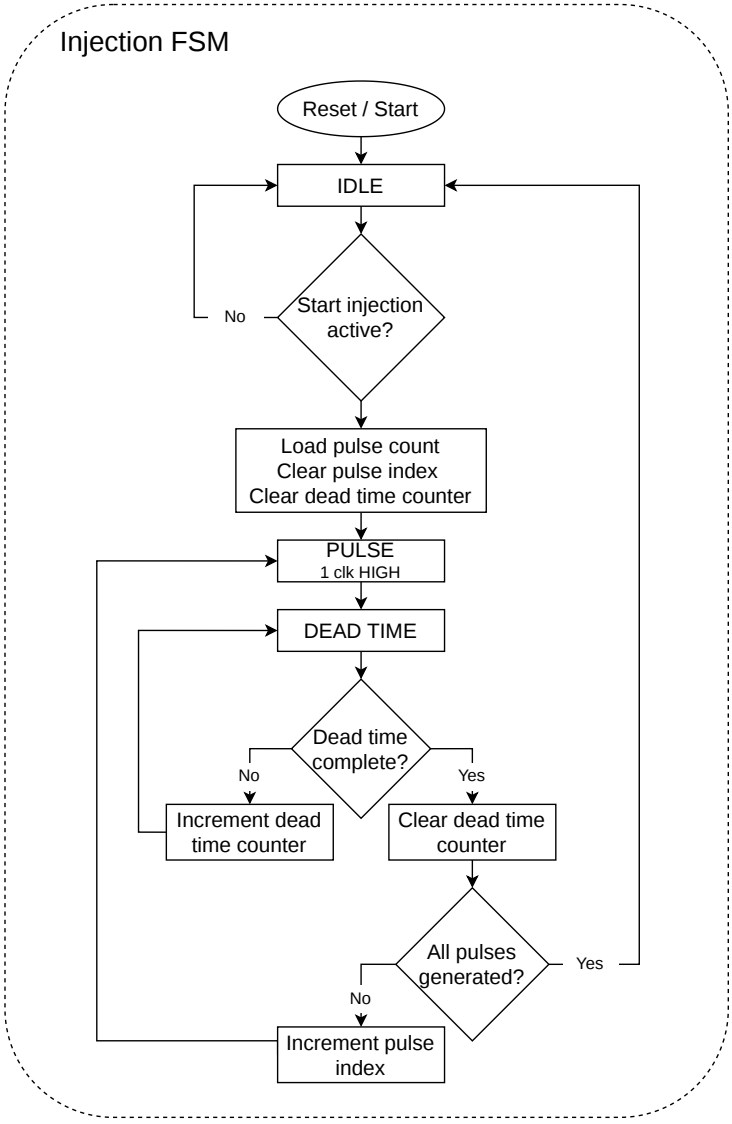


Figure 3.8 Firmware-side injection FSM used in the migrated implementation

The practical difference between the previous and migrated injection-control schemes for

a representative 90-pulse request is quantified in Table 3.3. In the previous implementation, 16 successive writes to `IPBADDR_CONTROL` were required for one pulse, corresponding to 512 bits of control payload per pulse. As a result, 1440 register writes, or 46 080 bits of control payload, were required for 90 pulses. These writes were packetized into six UDP/IPbus transactions when grouped in blocks of 15 pulses. In the migrated implementation, the injection request and pulse count are encoded in a single 32-bit control word, so the same operation requires only one software transaction. Thus, communication overhead is substantially reduced, and the previous restriction to grouped pulse counts is removed, allowing even a single pulse to be injected directly.

Table 3.3 Comparison of software-side control traffic for a representative injection request

Metric	**Previous implementation**	**Migrated implementation**
Control scheme	Software-paced pulse pattern	Firmware-generated pulse train
Writes per pulse	16	N/A
Example request	90 pulses	90 pulses
Writes for 90 pulses	1440	1
Payload for 90 pulses	46 080 bits	32 bits
Packet grouping	15 pulses/packet	Single packet
Packets for 90 pulses	6	1
Supported counts	15-pulse groups	Arbitrary count

3.5 Signal Conditioning and Alignment

The 37 LVDS output pairs from the MALTA2 sensor are routed into the FPGA through the FMC-HPC connector. At the FPGA input, each differential pair is first received by the `IBUFDS_DIFF_OUT` primitive, through which the incoming LVDS signal is converted into two complementary single-ended logic signals [34, 35]. As shown in **Figure 3.9**, the differential input is then split into separate P and N paths, each of which is passed through an independent `IDELAYE3` block before being forwarded to the `ISERDESE3` deserialization stage. In this way, the subsequent delay and deserialization stages are allowed to operate on the two sides of each differential pair separately inside the FPGA logic.

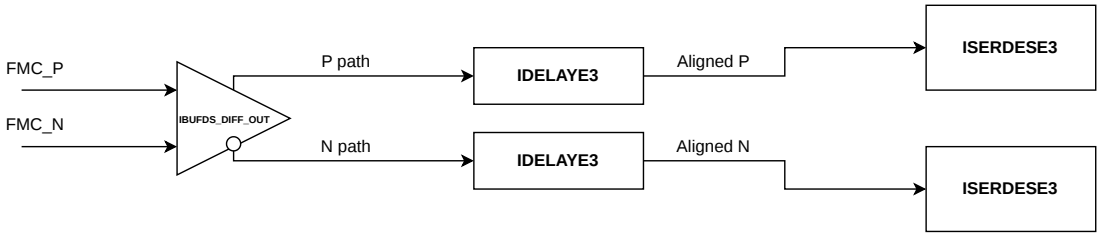


Figure 3.9 Signal conditioning and alignment

After the input buffer stage, both sides of each differential pair are passed through separate IDELAYE3 blocks. The purpose of this stage is to adjust the relative sampling phase of the incoming signals with fine granularity before deserialization. In the UltraScale architecture, IDELAYE3 is provided as a 512-tap programmable delay element with a 9-bit control interface [34, 35]. This provides significantly finer delay granularity than the 31-tap IDELAYE2 resource used in the original KC705 implementation [36].

In the previous KC705-based design, the delay settings exposed to the DAQ software were defined according to the native IDELAYE2 step size. In 7-series devices, the effective tap delay is dependent on the applied reference clock frequency; for example, the average tap delay is given as 78 ps at 200 MHz, 52 ps at 300 MHz, and 39 ps at 400 MHz in the Kintex-7 data sheet [28]. In the present UltraScale implementation, IDELAYE3 is operated in COUNT mode. For UltraScale devices, the IDELAY/ODELAY chain resolution is specified as a calibrated range rather than as a single fixed value [29]. Therefore, the effective tap delay used in this work was determined for the target device and operating point from simulation and was cross-checked with calibration results, yielding an approximate value of 5 ps per tap.

Although count-based delay programming is used in both implementations, the much finer IDELAYE3 granularity is not exposed directly to the DAQ software. Instead, the previous software-visible calibration model is preserved in the migrated firmware by mapping one software step to multiple UltraScale delay taps.

To preserve the existing software interface and calibration procedure, the same register-level convention was retained in the AXKU040 implementation. On the FPGA side, this is achieved by forcing the four least significant bits of the IDELAYE3 control word to zero. As a result, one software-visible delay step corresponds to 16 UltraScale taps, allowing the delay programming model seen by the DAQ software to remain effectively unchanged.

With the effective IDELAYE3 tap delay used in this work, this corresponds to approximately 80 ps per software-visible step, which is close to the approximately 78 ps tap delay used in the KC705-based implementation at 200 MHz.

In addition to the per-line delay setting, a fixed relative offset is introduced between the P and N paths. More precisely, the P-side delay is programmed with an additional offset corresponding to three software delay steps. Because one software step is mapped to 16 IDELAYE3 taps, this corresponds to 48 UltraScale taps in hardware, which is approximately 240 ps for the operating point used here. This intentional offset is introduced so that the P and N paths are sampled at slightly different time instants. As a result, a staggered sampling structure is created, and the effective timing information available during the oversampling stage is improved.

The main delay-control differences relevant to the migration are summarized in Table 3.4.

Table 3.4 IDELAY implementation comparison

Metric	**KC705 implementation**	**AXKU040 implementation**
Delay primitive	IDELAYE2	IDELAYE3
Delay mode	count mode	count mode
Control width	5-bit	9-bit
Number of taps	31	511
Effective tap delay in this work	∼78 ps/tap	∼5 ps/tap
Software-visible range	31 steps	31 steps
P/N offset	3 steps	3 steps
Hardware mapping per step	1 step = 1 tap	1 step = 16 taps

The delay settings are controlled through the per-channel TAP-calibration registers in the IPbus address space. The write registers AO_ADDR_TAPW[0] to AO_ADDR_TAPW[36] are mapped to addresses 10 to 46 and are used to program the IDELAYE3 tap values for the 37 input lines. The corresponding readback registers AO_ADDR_TAPR[0] to AO_ADDR_TAPR[36] are mapped to addresses 50 to 86 and are used to monitor the applied tap settings. During calibration, the oversampling response is observed through the monitoring interface, and the per-line tap values are adjusted until stable hit sampling is achieved across the sensor outputs.

3.6 Asynchronous Oversampling

To capture the short MALTA2 hit pulses with sufficient time resolution, the aligned LVDS signals are oversampled using `ISERDESE3` primitives [34]. As shown in **Figure 3.10**, the deserialization stage is driven by two clock domains derived from the same MMCM output, following the general high-speed source-synchronous deserialization structure illustrated in [37]. The 640 MHz clock is taken as a raw MMCM output and is buffered through a dedicated `BUFG` to drive the global clock network. The 320 MHz clock is generated from the same raw 640 MHz MMCM output using a `BUFGCE_DIV` primitive, while the 640 MHz clock is distributed through a dedicated `BUFG`. By deriving both clocks directly from the same source through parallel global clock buffers, additional insertion delay is avoided and skew between the `CLK` and `CLKDIV` domains is reduced.

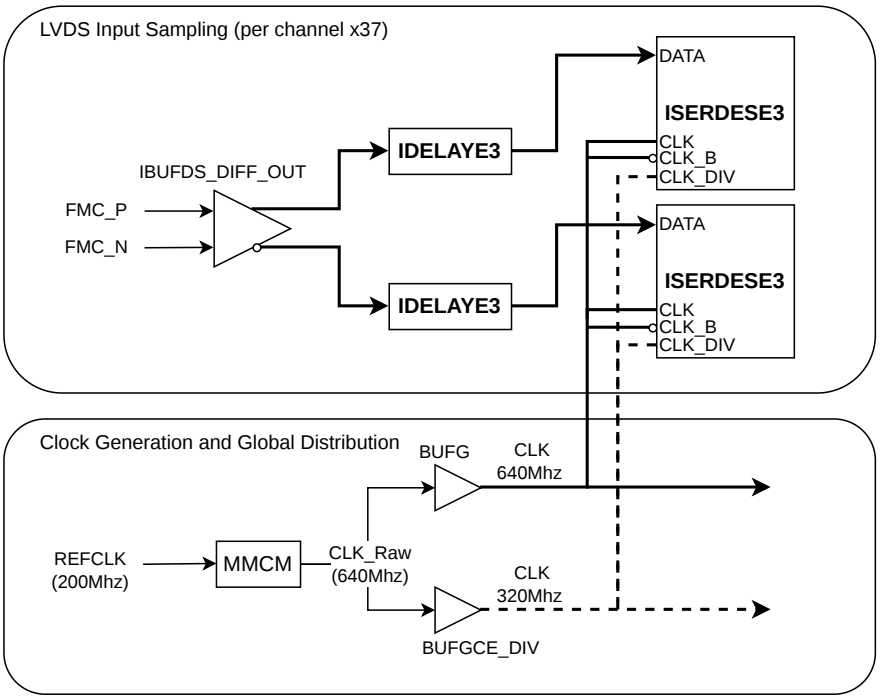


Figure 3.10 Oversampling architecture

In the UltraScale implementation, the high-speed clocking topology was chosen such that the 640 MHz sampling clock is distributed as a single clock net, while the inverted version is handled locally inside the receiving resources rather than being routed as a separate global clock path. This choice follows the recommended UltraScale clocking practice for

high-speed DDR interfaces using ISERDESE3, where a single clock net together with local inversion is preferred in order to meet the skew requirement between the clock and inverted clock pins while using fewer global clock resources [38]. In the present design, clock skew at the deserializer interface is minimized by this choice, and more consistent sampling across channels is supported. Compared with the previous KC705-based implementation, the need to present four separate phase-shifted clock signals to the deserializer is also removed and replaced with a simpler two-clock interface.

After the phase-alignment stage, the delayed P and N signals are forwarded to ISERDESE3 for deserialization. Each path is sampled in DDR mode using the 640 MHz clock, so data are captured on both the rising and falling edges of the clock. As a result, two samples are produced per clock cycle; therefore, two 640 MHz cycles are required to generate a 4-bit deserialized word. The resulting 4-bit parallel outputs are transferred into the 320 MHz CLKDIV domain [34].

Table 3.5 ISERDES implementation comparison

Metric	**KC705 implementation**	**AXKU040 implementation**
Deserializer primitive	ISERDESE2	ISERDESE3
Sampling mode	DDR	DDR
Data width	4	4
FIFO usage	Disabled	Disabled
Clock inputs used	4	2
Clocking approach	Four-phase external clocks	Single 640 MHz clock
Divided clock	CLKDIV not used	CLKDIV = 320 MHz

The main primitive-level and implementation-level differences between the original KC705-based deserialization stage and the migrated AXKU040 implementation are summarized in Table 3.5.

A sampling-level view of this process is shown in **Figure 3.11**. Since the P and N paths are sampled independently, four samples are obtained from the P path and four from the N path during each 320 MHz cycle. For MALTA2, a hit is represented by a differential transition in which P goes low and N goes high. To obtain a uniform digital representation, the P-side samples are inverted in the FPGA, while the N-side samples are kept unchanged.

After this polarity normalization, logic 0 denotes the absence of a hit and logic 1 denotes the presence of a hit on both paths.

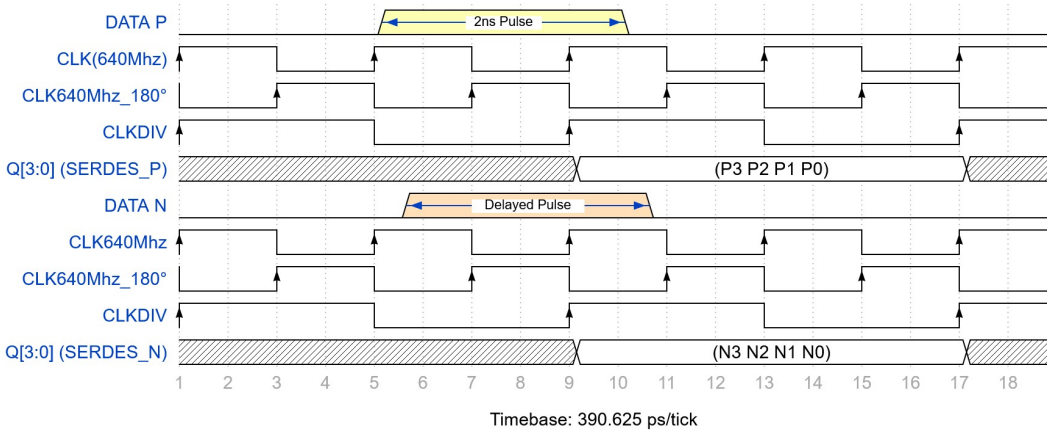


Figure 3.11 DDR sampling of the P and N signal paths

The normalized P-side and N-side samples are then combined to form an 8-bit sample window for each input line in every 320 MHz cycle. In this window, the temporal evolution of the signal over the sampling interval is captured, preserving not only hit presence but also the approximate position of the transition within the window. As a result, the asynchronously arriving sensor pulse is represented as a synchronous oversampled pattern in the FPGA clock domain. This representation provides the basis for subsequent pattern-based hit reconstruction and for extracting finer timing information from the sampled pulse position.

3.7 Hit Reconstruction

For each input line and for each 320 MHz clock cycle, an 8-bit sample window representing the recent time evolution of the signal is produced by the oversampling stage. These windows are then processed by the hit-reconstruction logic to determine whether a valid hit has occurred and, if so, at which phase within the sampling window it should be reconstructed.

The reconstruction is referenced to one dedicated line, namely pin 0. The 8-bit sample window of this reference line is analyzed by the `edge_detector` logic, in which the observed pattern is checked against the valid reference-hit patterns. If a valid pattern is found, the

reconstruction-enable signal (`m_re`) is asserted, and both a reference reconstruction pattern (`m_re_pos`) and a 3-bit position code (`m_position`) are produced. In this way, incomplete or spurious bit sequences on the reference line are rejected, while only patterns consistent with a valid hit are accepted for reconstruction.

Table 3.6 Reference-pattern mapping in the edge detector

Reference pattern	m_position	m_re_pos	m_re	Case type
00000110	001	00001111	1	Emergency 2-bin word
00001100	010	00011110	1	Emergency 2-bin word
00011000	011	00111100	1	Emergency 2-bin word
00110000	100	01111000	1	Emergency 2-bin word
01100000	101	11110000	1	Emergency 2-bin word
00000111 | 00001111 | 00011111 | 00111111	000	00001111	1	Single word
00001110 | 00011110 | 00111110 | 01111110	001	00011111	1	Single word
00011100 | 00111100 | 01111100 | 11111100	010	00111111	1	Single word
00111000 | 01111000 | 11111000	011	01111110	1	Single word
01110000 | 11110000	100	11111100	1	Single word
11100000	101	11111000	1	Single word
11000000	110	11110000	1	Single word
10000000	111	11110000	1	Single word
11000001	110	11110000	1	Double word
11000011	110	11110000	1	Double word
11100001	101	11111000	1	Double word
00000011	000	00001111	1	Ambiguous
others	000	00000000	0	Default

The mapping implemented in the `edge_detector` is summarized in Table 3.6. For each recognized reference pattern on pin 0, a position code (`m_position`), a reference reconstruction pattern (`m_re_pos`), and a reconstruction-enable flag (`m_re`) are produced by the

logic. The table includes standard single-word cases, special emergency 2-bin cases, selected double-word cases, and the default invalid case. The position code represents the phase of the reconstructed hit within the oversampling window and is later stored in the output word as phase information.

The reconstruction parameters derived from the reference line are then distributed to the `pin_detector` logic of all non-reference input channels. As shown in **Figure 3.12**, the local 8-bit sample window is compared with the 8-bit reference window in each `pin_detector` instance, and the bitwise matches are combined into a 1-bit hit result. The same logic is repeated for the 36 non-reference channels, so that one reconstructed hit bit is contributed to the final output word by each LVDS input line.

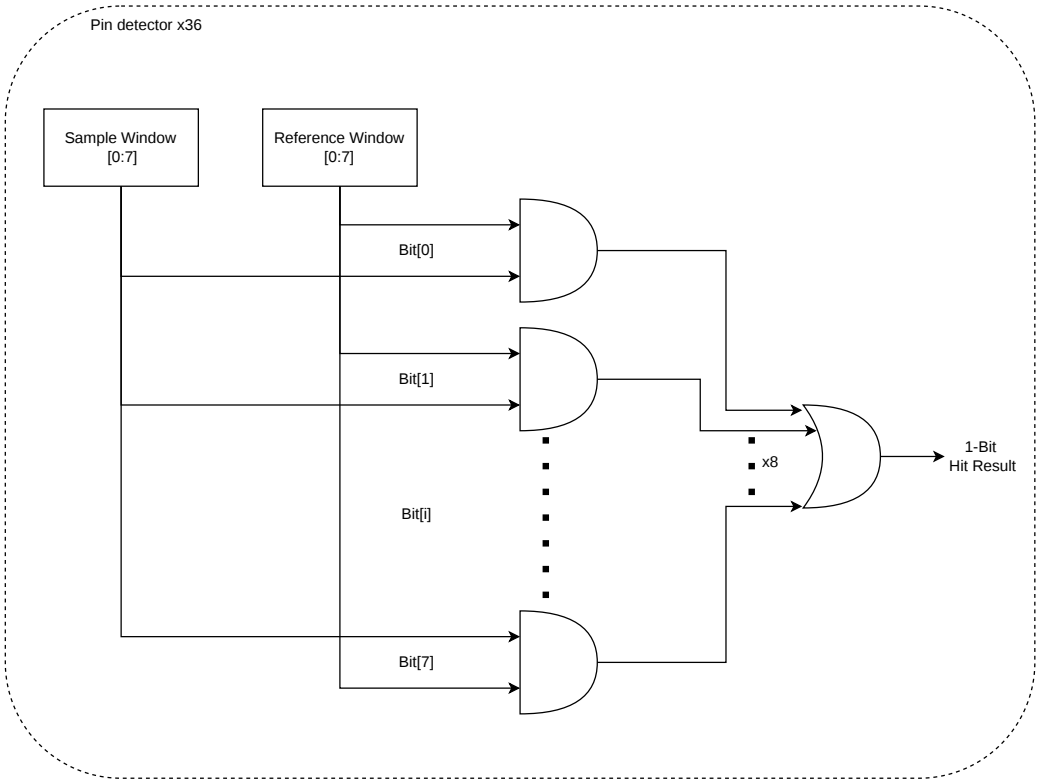


Figure 3.12 Pin detector logic

The reconstructed outputs of all 37 channels are first assembled into the MALTA2 data word before entering the buffering and readout chain. In addition to the reconstructed hit information, the word also includes the phase information derived from the reference line together with the associated timing and event-identification fields. Table 3.7 summarizes the field definition used for the MALTA2 data word in the analysis.

Through this reconstruction scheme, the oversampled asynchronous pulse patterns are converted into synchronized hit information in the 320 MHz processing domain, from which the reconstructed data are passed to the buffering and readout logic.

Table 3.7 MALTA2 data-word field definition

Bit range	Field	Description
[0]	refbit	Reference-bit flag
[16:1]	pixel	Pixel address
[21:17]	group	Pixel group index
[22]	parity	Parity bit
[25:23]	delay	Delay-related field
[33:26]	dcolumn	Double-column address
[35:34]	chipbcid	Chip BCID field
[36]	chipid	Chip identifier
[39:37]	phase	Reconstructed phase
[42:40]	winid	Window identifier
[49:44]	bcid	Bunch-crossing identifier
[61:50]	l1id	Level-1 trigger identifier

3.8 FIFO Chain

After hit reconstruction, the final MALTA2 event word is transferred through a dedicated FIFO chain before being made available to the DAQ software through IPbus. In this stage, the different clock domains of the readout path are bridged, and the internal event-word format is converted into a 32-bit stream compatible with IPbus readout.

As described in the previous subsection, the reconstructed hit information is assembled in the 320 MHz processing domain into the final MALTA2 event word. This word contains the reconstructed channel information together with the phase and timing-related fields summarized in Table 3.7. In the firmware, the final word has a width of 62 bits.

The overall structure of the FIFO chain is shown in **Figure 3.13**. The 62-bit final word produced in the 320 MHz domain is first buffered in FIFO1, then reformatted by the `FifoConnector`, and finally transferred through FIFO2 to the 32-bit IPbus readout path. The first stage, referred to here as FIFO1, corresponds to the `fifo_Fast` block in the

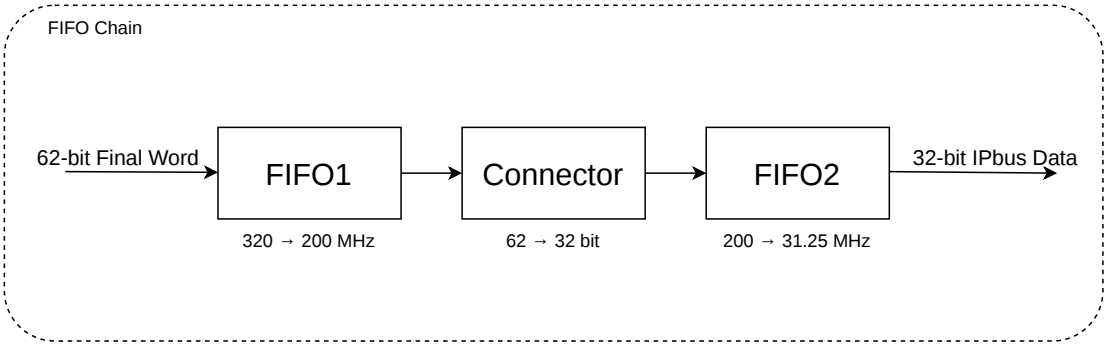


Figure 3.13 FIFO chain in the MALTA2 readout path

firmware. It is implemented as an asynchronous FIFO that is written in the 320 MHz domain and read in the 200 MHz domain. Its role is to decouple the hit-processing logic from the downstream readout path while preserving the full 62-bit event-word format.

The output of FIFO1 is passed to the `FifoConnector` block. This block is not implemented as a FIFO; instead, a width conversion is performed between the internal 62-bit event word and the 32-bit data format used by the IPbus readout path. In practice, each reconstructed event word is split into two 32-bit words before being forwarded to the next stage.

The second stage, referred to here as FIFO2, corresponds to the `fifo_IPbus` block. This block is also implemented as an asynchronous FIFO. It is written in the 200 MHz domain and read in the IPbus clock domain at 31.25 MHz. Therefore, the final clock-domain crossing between the internal readout path and the software-visible IPbus interface is provided by FIFO2.

On the software side, the event data are read through the FIFO-mapped IPbus register `IPBADDR_AOFIFO`. Before reading, the number of available 32-bit words in FIFO2 is first queried by the DAQ software, and the corresponding amount of data is then read from `IPBADDR_AOFIFO`. The available word count is provided by the FIFO status path, which is exposed in the firmware through `IPBADDR_AOSTATUS`.

3.9 Slow Control

The MALTA2 slow-control interface is used to program the chip configuration parameters from the DAQ system. A fixed-length configuration word is transported from the host

software to the on-chip configuration registers, as summarised in **Figure 3.14**.

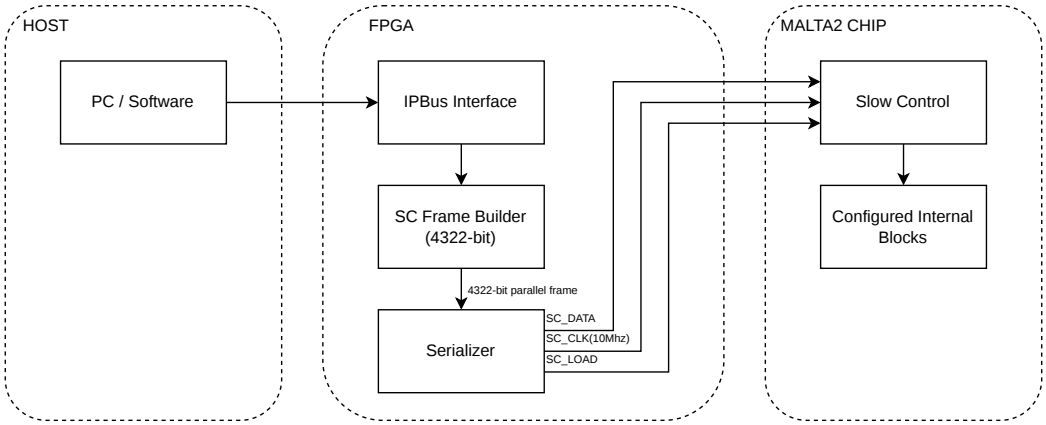


Figure 3.14 System-level overview of the MALTA2 slow-control path

As shown in **Figure 3.14**, configuration values are set on the host PC and transferred to the FPGA via IPbus transactions. In the FPGA, these values are assembled by the slow-control logic into a 4322-bit configuration frame and serialized towards the MALTA2 chip. The serial interface uses `SC_DATA` clocked by `SC_CLK` at 10 MHz, while a dedicated `SC_LOAD` strobe is used to apply the received configuration on the chip.

The 32-bit IPbus write stream is converted by the FPGA slow-control firmware into the fixed-length configuration frame required by the MALTA2 chip. This process is illustrated in **Figure 3.15**.

As shown in **Figure 3.15**, the slow-control content is streamed by the host software as 32-bit IPbus writes to the slow-control FIFO write register, `IPBADDR_SC_FIFO_W`. The data are buffered in a dual-clock FIFO to cross from the IPbus clock domain at 31.25 MHz to the slow-control clock domain at 10 MHz. Then, 136 words, corresponding to $136 \times 32 = 4352$ bits, are read by an FSM, and the 4322-bit slow-control payload, `[4321:0]`, is extracted. The payload is transmitted by the serializer on `SC_DATA`, clocked by `SC_CLK` at 10 MHz, and `SC_LOAD` is asserted once the frame transmission is complete. The remaining 30 bits are unused padding introduced by the 32-bit word alignment.

Compared with the previous implementation, the slow-control write interface was significantly simplified. In the earlier design, several separate slow-control write and control registers were accessed by the software, which resulted in a more fragmented and slower configuration procedure. In the migrated firmware, this structure was replaced by a single

streaming write register, IPBADDR_SC_FIFO_W, through which the complete slow-control content is transferred as a sequence of 32-bit words. With this simplification, readability and maintainability of the software–firmware interface are improved, debugging is made easier, and the number of required register transactions during configuration is reduced. As a result, the slow-control procedure becomes simpler in operation and requires fewer software-side register transactions.

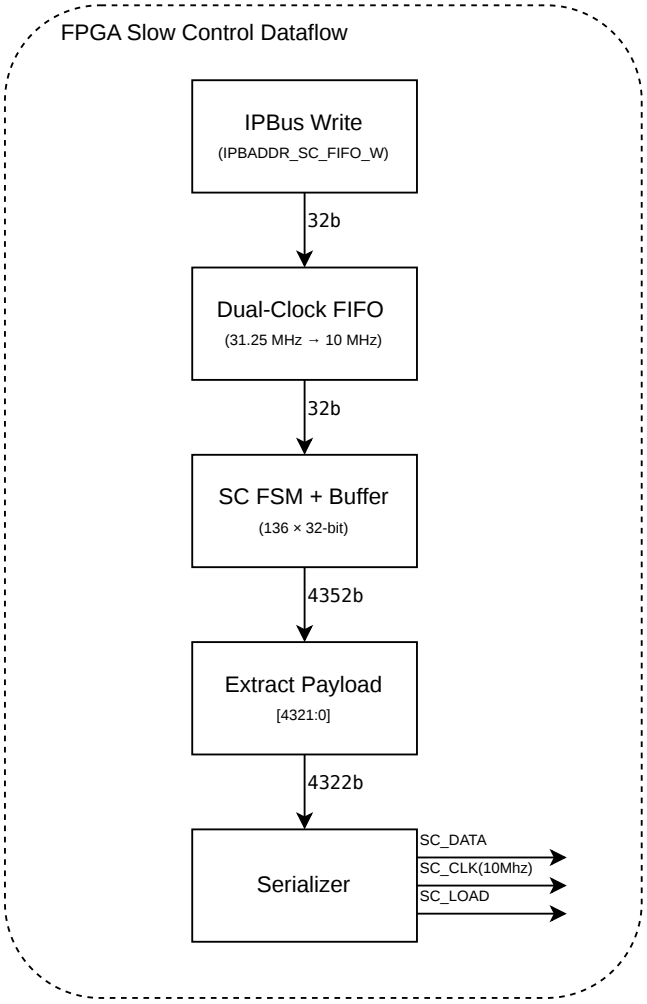


Figure 3.15 FPGA implementation of the slow-control dataflow for MALTA2 configuration

Inside the chip, as shown in **Figure 3.16**, the incoming serial stream is first accumulated in a temporary 4322-bit shift register. Upon assertion of LOAD, the contents are transferred in parallel to a static configuration register bank. The configuration bits are then distributed to three main domains: global digital configuration, such as dataflow and LVDS settings; bias DAC configuration, including voltage and current biases for the front-end; and pixel

matrix configuration, including masking and pulsing/injection control.

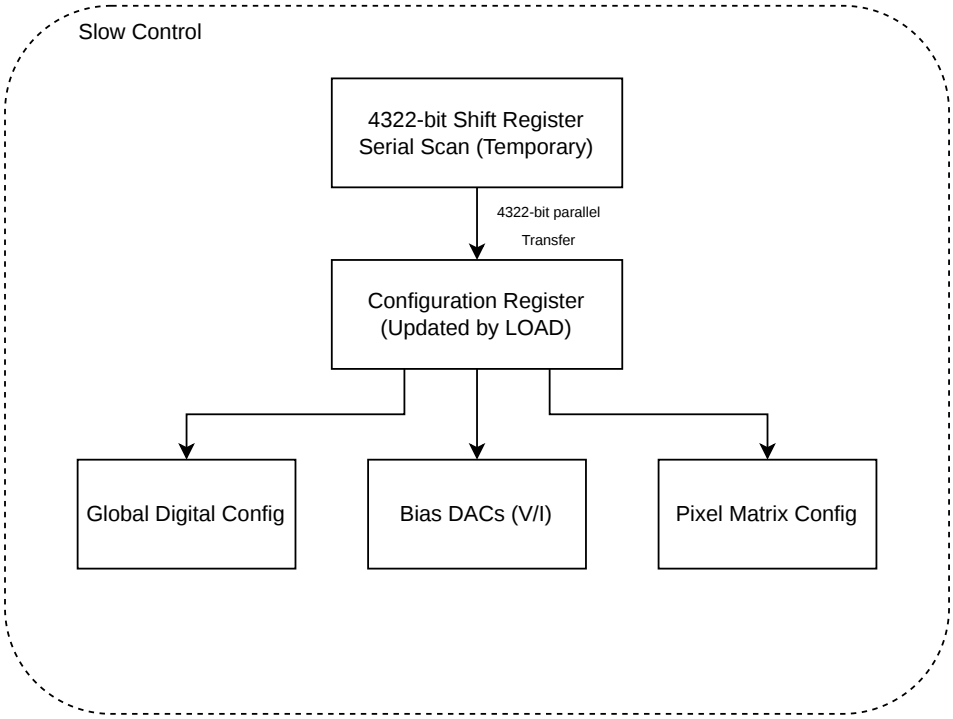


Figure 3.16 Internal organisation of the MALTA2 slow-control configuration

In the following, the slow-control configuration is described in terms of a fixed-length configuration word. The complete 4322-bit register map, including bit positions, field sizes, default values, and brief descriptions, is listed in Table 3.8.

Voltage and current bias settings are encoded differently in the slow-control word. Voltage DACs are encoded using a *one-hot* representation, where exactly one bit out of a 128-bit field is asserted to select the target voltage step. Current DACs are encoded using a *thermometer* representation, where the programmed value is represented by a contiguous sequence of asserted bits, for example N ones followed by zeros, within a 128-bit field.

Table 3.8 Slow control registers

Name	Pos	Size	Default	Comment
DF_MERGERTOLEFT	0	1	0x1	Dataflow registers control the
DF_MERGERTORIGHT	1	1	0x0	direction of propagation of
DF_LMERGERTOLVDS	2	1	0x1	the output data

Name	Pos	Size	Default	Comment
DF_LMERGERTOCMOS	3	1	0x1	
DF_RMERGERTOLVDS	4	1	0x0	
DF_RMERGERTOCMOS	5	1	0x0	
DF_LCMOS	6	1	0x0	
DF_RCMOS	7	1	0x0	
DF_ENLVDS	8	1	0x1	
DF_ENLCMOS	9	1	0x1	
DF_ENRCMOS	10	1	0x0	
DF_ENMERGER	11	1	0x0	
POWER_SWITCH_LEFT	12	1	0x1	Power left DACs from Matrix
POWER_SWITCH_RIGHT	13	1	0x0	Power right DACs from Matrix
LVDS_EN	14	1	0x1	LVDS enable output
LVDS_PRE	15	16	0x00FF	Pre Emphasis LVDS OUTPUT, tune slew rate
LVDS_BRIDGE_EN	31	5	0x1C	LVDS output, tune output current
LVDS_CMFB_EN	35	5	0x1C	LVDS output internal feedback lock
LVDS_SET_IBCMFB	41	4	0x7	bias current of VCM feedback amp
LVDS_SET_IVPH	45	4	0x2	Ref bias current: Ioff Hbridge PMOS
LVDS_SET_IVPL	49	4	0xD	Ref bias current: Ion Hbridge PMOS
LVDS_SET_IVNH	53	4	0xD	Ref bias current: Ion Hbridge NMOS
LVDS_SET_IVNL	57	4	0x7	Ref bias current: Ioff Hbridge NMOS
SWCNTL_VCASN	61	1	0x0	It selects monitor/override for the DACs
SWCNTL_VCLIP	62	1	0x0	
SWCNTL_VPLSE_HIGH	63	1	0x0	

Name	Pos	Size	Default	Comment
SWCNTL_VPLSE_LOW	64	1	0x0	
SWCNTL_VRESET_P	65	1	0x0	
SWCNTL_VRESET_D	66	1	0x0	
SWCNTL_ICASN	67	1	0x0	
SWCNTL_IRESET	68	1	0x0	
SWCNTL_IBIAS	69	1	0x0	
SWCNTL_ITHR	70	1	0x0	
SWCNTL_IDB	71	1	0x0	
SWCNTL_IREF	72	1	0x0	
SET_IRESET_BIT	73	1	0x1	
SWCNTL_DACMONV	74	1	0x0	
SWCNTL_DACMONI	75	1	0x0	
SET_IBUFP_MON_0	76	4	0x5	Current DACs for the monitoring pixels
SET_IBUFN_MON_0	80	4	0x9	
SET_IBUFP_MON_1	84	4	0x5	
SET_IBUFN_MON_1	88	4	0x9	
SET_VCASN	92	128	One-Hot: 110	
SET_VCLIP	220	128	One-Hot: 127	
SET_VPULSE_HIGH	348	128	One-Hot: 90	
SET_VPULSE_LOW	476	128	One-Hot: 10	
SET_VRESET_P	604	128	One-Hot: 29	
SET_VRESET_D	732	128	One-Hot: 65	
SET_ICASN	860	128	Thermo: 10	
SET_IRESET	988	128	Thermo: 30	
SET_ITHR	1116	128	Thermo: 80	
SET_IBIAS	1244	128	Thermo: 43	
SET_IDB	1372	128	Thermo: 120	
MASK_COL	1500	512	All 1	

Name	Pos	Size	Default	Comment
MASK_HOR	2012	512	All 1	
MASK_DIAG	2524	512	All 1	
MASK_FULL_COL	3036	256	All 1	
PULSE_COL	3292	512	0	
PULSE_HOR	3804	512	0	
PULSE_MON_L	4316	1	0x0	Pulsing of the monitoring pixels
PULSE_MON_R	4317	1	0x0	
LVDS_VBCMFB	4318	4	0xE	Pulse width 0b0111 : 2 ns 0b1011 : 1 ns 0b1101 : 750 ps 0b1110 : 500 ps (default)

4. RESULTS

4.1 Implementation Summary

In order to quantitatively demonstrate the effect of the FPGA-platform migration, the implemented AXKU040 design was evaluated both at the FPGA implementation level and through comparative measurements against the previous KC705-based readout. The comparison was based on tap calibration, analog scans, and threshold scans performed with the same MALTA2 chip under comparable operating conditions.

The migrated firmware implementation targeting the AXKU040 board was first evaluated at the FPGA implementation level. Post-route timing analysis confirmed timing closure for the migrated firmware, with no failing endpoints in setup, hold, or pulse-width checks. The worst setup slack, worst hold slack, and worst pulse-width slack were 0.058 ns, 0.016 ns, and 0.164 ns, respectively, as summarized in Table 4.1.

Table 4.1 Post-route timing summary of the implemented AXKU040 firmware

Check type	Worst slack (ns)	Total negative slack (ns)	Failing endpoints
Setup	0.058	0	0
Hold	0.016	0	0
Pulse width	0.164	0	0

The overall post-implementation resource utilization is summarized in Table 4.2. The results show that the migrated design fits comfortably within the available resources of the target device. The highest utilization is observed for BRAM resources, with 208.50 blocks used out of 600 available blocks, corresponding to 34.75%. The remaining resource categories remain well below the available limits.

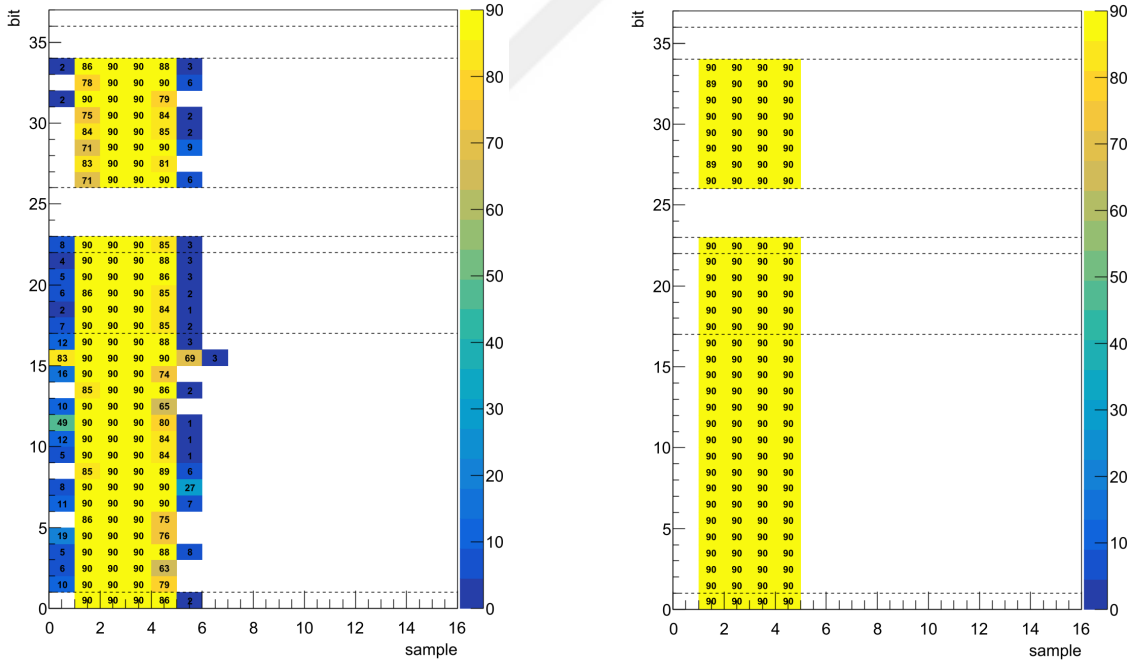
These implementation results indicate that the migrated firmware was successfully placed and routed on the AXKU040 platform with positive timing margins and without excessive use of FPGA resources. Therefore, the migrated design provides sufficient implementation headroom for reliable operation and for possible future firmware extensions.

Table 4.2 Post-implementation resource utilization of the migrated AXKU040 firmware

Resource	Used	Available	Utilization (%)
LUT	19744	242400	8.15
FF	26386	484800	5.44
BRAM	208.50	600	34.75
IO	128	520	24.62
BUFG	16	480	3.33
MMCM	3	10	30.00

4.2 Tap Calibration

Tap calibration was used to align the sampling points of the 37 MALTA2 LVDS output channels at the FPGA input. By adjusting the per-channel delay settings, relative timing differences between the input lines were compensated, and a consistent oversampled hit pattern was recovered across the readout channels.



(a) KC705 after calibration

(b) AXKU040 after calibration

Figure 4.1 Tap-calibration maps after timing adjustment with 90 injected hits. The vertical axis shows the readout-channel index, the horizontal axis shows the position within the oversampled hit pattern, and the color scale indicates the number of recorded hits at each sampling position

The tap-calibration maps obtained after timing adjustment are shown in **Figure 4.1**. The

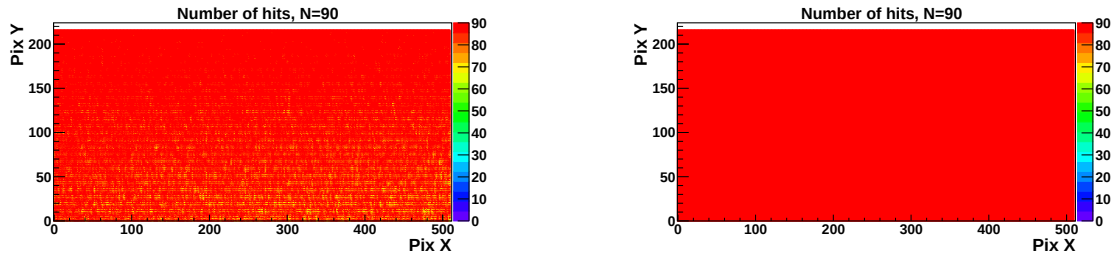
vertical axis denotes the readout-channel index, the horizontal axis denotes the position within the oversampled hit pattern, and the color scale gives the number of recorded hits at each position. In an ideal calibration result, the recorded counts are expected to be concentrated in a regular sampling region and to remain close to the injected value.

As shown in **Figure 4.1**, the KC705-based implementation still shows visible irregularities after calibration, with some channels deviating from the expected sampling pattern. In contrast, a more uniform calibrated response is observed with the AXKU040-based implementation, where the recorded pulse counts remain closer to the injected value across the sampling structure. This indicates that a cleaner and more stable sampling behaviour was obtained in the migrated implementation.

The observed improvement is consistent with the updated asynchronous readout architecture, in particular with the reduction of clock skew in the sampling structure and the improved alignment of the deserialized data across the input channels.

4.3 Analog Scan

Analog scans were used to evaluate how uniformly the readout system measured the response of all pixels across the full MALTA2 matrix. In this procedure, the same number of injection pulses was applied pixel by pixel, and the measured hit count was recorded for each pixel. The result was obtained as a two-dimensional map of the measured response over the sensor matrix.



(a) KC705 **(b)** AXKU040

Figure 4.2 Two-dimensional analog-scan response maps obtained with 90 injected pulses per pixel. The horizontal axis corresponds to the pixel column, the vertical axis corresponds to the pixel row in the MALTA2 matrix, and the color scale indicates the measured hit count per pixel

The analog-scan results show the clearest functional improvement of the migrated firm-

ware. As shown in **Figure 4.2**, the main effect is not a change in the intrinsic response of the sensor pixels themselves, but an improvement in the ability of the readout system to measure the hit count uniformly across the full pixel matrix. In these measurements, injection was used as a controlled and reproducible test method; however, the observed improvement reflects the behaviour of the readout chain itself rather than a feature specific only to the injection procedure.

In the previous KC705-based implementation, some pixels systematically showed fewer hits than expected. In the migrated AXKU040-based implementation, this discrepancy was strongly reduced, and the measured hit count followed the expected response much more closely over the matrix.

The representative analog-scan measurement shown in **Figure 4.2** is quantified in Table 4.3. The number of pixels with a measured hit count within 5% of the expected trigger count increased from 85102 in the KC705 implementation to 110160 in the AXKU040 implementation. For the AXKU040 implementation, this corresponds to the full pixel matrix. In the KC705 implementation, a further 16107 pixels remained within the 5–10% deviation band, while 8951 pixels exhibited a hit count below 90% of the expected value. In contrast, for the AXKU040 implementation, all pixels were measured within 5% of the expected count.

Table 4.3 Analog-scan result summary for the same MALTA2 sample with 90 injected pulses

Metric	**KC705**	**AXKU040**
Pixels with 0 hits	0	0
Pixels with hits $< 90\%$ of $nTrigger$	8951	0
Pixels with hits $> 110\%$ of $nTrigger$	0	0
Pixels within 5–10% of $nTrigger$	16107	0
Pixels within 5% of $nTrigger$	85102	110160
Total scan time	39 s	16 s

These results indicate that, under 90 trigger injections per pixel, the pixels across the full matrix were measured in a highly consistent manner with the AXKU040-based readout. The observed improvement is consistent with the refined asynchronous readout implementation, in particular with the reduction of clock skew in the sampling structure. By improving the timing consistency of the oversampling stage, the migrated firmware pro-

vides a more uniform response across all channels. As a result, the number of pixels that previously had to be masked because of readout-related undercounting is significantly reduced. In practice, such masking is applied on a pixel-by-pixel basis to exclude pixels with missing, excess, or otherwise unreliable hit counts from subsequent measurements and tracking use.

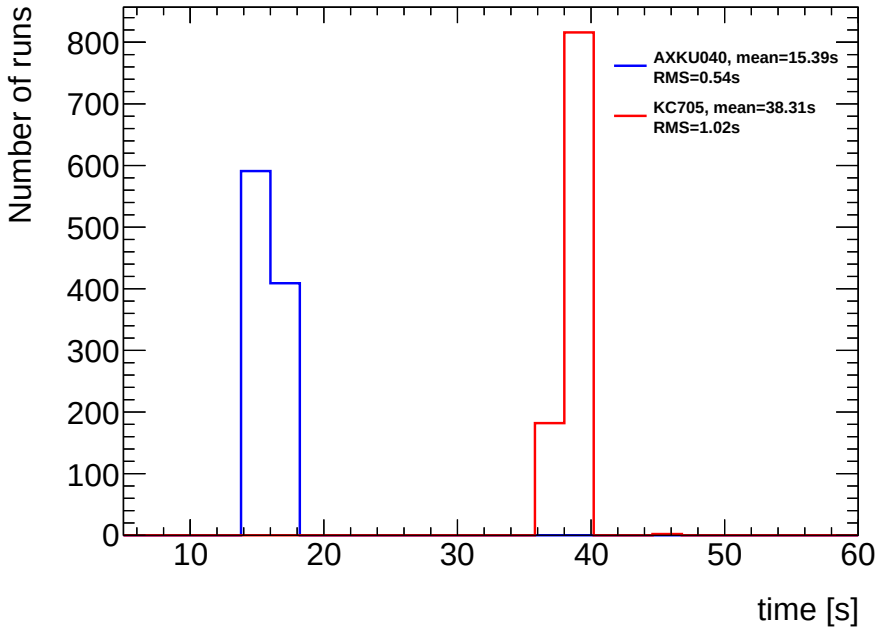


Figure 4.3 Analog-scan execution-time distribution over 1000 repeated runs for the KC705-based and AXKU040-based implementations. The distributions show the scan duration measured for each run and are used to compare both the average execution time and the run-to-run timing stability of the two readout implementations

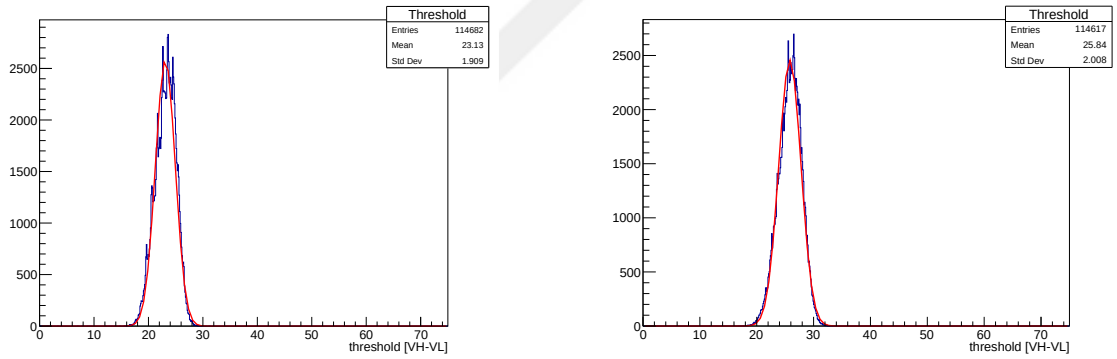
To evaluate this behaviour beyond a single scan, 1000 analog-scan runs were performed for each implementation. Averaged over these repeated measurements, the AXKU040-based implementation maintained 99.979% of pixels within 10% of the expected trigger count and 99.978% within 5%, while only 0.013% of pixels remained below 90% of $nTrigger$. For the KC705-based implementation, the corresponding mean fractions were 91.758% within 10%, 77.178% within 5%, and 8.242% below 90% of $nTrigger$. These averaged results confirm that the improved analog-scan uniformity was not limited to an isolated measurement, but was consistently observed over repeated runs.

The corresponding execution-time distributions are shown in **Figure 4.3**. The AXKU040-based implementation yielded a mean analog-scan time of 15.39 s with an RMS of 0.54 s, whereas the KC705-based implementation gave a mean time of 38.31 s with an RMS of

1.02 s. In addition to the broader overall distribution, the KC705 data also showed a small tail of slower scans around 46 s, which contributed to the larger spread. These results confirm that the migrated firmware not only reduced the analog-scan duration significantly, but also provided more stable scan timing over repeated runs.

4.4 Threshold Scan

Threshold scans were performed to study the threshold behaviour of the pixel matrix. In this procedure, the injection-pulse amplitude was varied while the threshold setting was kept fixed, and the pixel response was recorded in order to determine the signal level at which hits were registered consistently. In the MALTA2 injection circuit, the injected signal amplitude is defined by the voltage difference $V_H - V_L$ [21]. Repetition of this procedure for the measured pixels allowed the threshold distribution of the chip to be obtained.



(a) KC705 **(b)** AXKU040

Figure 4.4 Threshold distributions for the same MALTA2 sample under the same threshold setting. The horizontal axis shows the extracted threshold in units of $V_H - V_L$, while the vertical axis shows the number of pixels in each threshold bin

The threshold distributions obtained from the measured pixels are shown in **Figure 4.4**. Both distributions exhibit an approximately Gaussian shape, indicating that the pixel thresholds are centered around a well-defined mean value with a limited pixel-to-pixel spread. The entry counts shown in the figure correspond to the total number of pixels contributing to each histogram.

As shown in **Figure 4.4**, broadly consistent threshold distributions were obtained with

the KC705-based and AXKU040-based implementations. The overall distribution shape, mean value, and spread remain comparable, indicating that the migrated firmware preserves the expected threshold-measurement behaviour and does not distort the underlying threshold response of the sensor.

Unlike the analog scan, however, the threshold scan should not be interpreted as a direct full-matrix readout-efficiency measurement. The threshold analysis was performed only for pixels that could already be read out reliably. Therefore, the larger number of entries observed for the AXKU040-based implementation mainly reflects the fact that fewer pixels had to be excluded from the measurement. In this sense, the migrated firmware enabled threshold scans to be performed on a larger usable fraction of the matrix while maintaining a threshold response consistent with that of the previous implementation.

A further practical improvement was observed in the scan duration. In the KC705-based system, a single threshold scan required approximately 55 minutes, whereas the same procedure was completed in about 25 minutes with the AXKU040-based implementation. This corresponds to an approximate reduction by a factor of two. For studies repeated at five different parameter settings, the total measurement time was thereby reduced from about 5 hours to about 2.5 hours, making extended threshold studies considerably more practical.

5. MALTA2 HIGH-RATE SIMULATION STUDY WITH AN ATLAS ITK-BASED REFERENCE

The laboratory measurements presented in the previous chapter validate the migrated MALTA2 readout firmware under controlled injection-based conditions. However, the response of a pixel sensor and its readout chain also depends on the particle occupancy, energy-deposition pattern, and hit rate expected in a realistic tracking environment. Therefore, an additional simulation-based study was performed to evaluate the MALTA2 response under high-rate conditions derived from ATLAS ITk occupancy information.

The purpose of this study is to assess the feasibility and limitations of MALTA2 with respect to a possible ITk-replacement scenario. In this context, replacement does not mean that MALTA2 is already implemented in the baseline ATLAS ITk detector. Instead, the ITk occupancy information is used as a demanding reference environment against which the MALTA2 sensor and readout response can be tested. The simulation therefore provides a controlled method for determining whether MALTA2 remains efficient under ITk-derived occupancy conditions and for identifying the conditions under which the present MALTA2 architecture becomes limited.

5.1 Simulation Methodology

The simulation chain used in this study follows the sequence from particle generation and energy deposition to hit formation and digital readout response. The first stage is based on Geant4, which is a simulation toolkit widely used for modelling the passage of particles through matter. In this work, Geant4 is used to describe the interaction of incident particles with the MALTA2 sensor volume and to estimate the spatial distribution of energy deposited in the sensitive epitaxial layer.

The primary particles are generated using a particle-gun approach. In this type of simulation, the particle gun defines the primary particle type, its energy, direction, and starting position before the event is transported through the detector geometry. For the ITk-replacement evaluation, the number of particles generated per 25 ns interval is derived from ITk hit-occupancy inputs. The hit density is interpreted in units of hits per mm^2 per 25

ns bunch crossing, scaled to the MALTA2 sensor area, and used to determine the particle multiplicity through Poisson sampling. This provides a controlled method for studying the MALTA2 response under occupancy conditions motivated by the ATLAS ITk environment. For the beam-rate study, 120 GeV protons were used as the incident particles, while the MALTA2 sensor was represented with a 30 µm epitaxial layer.

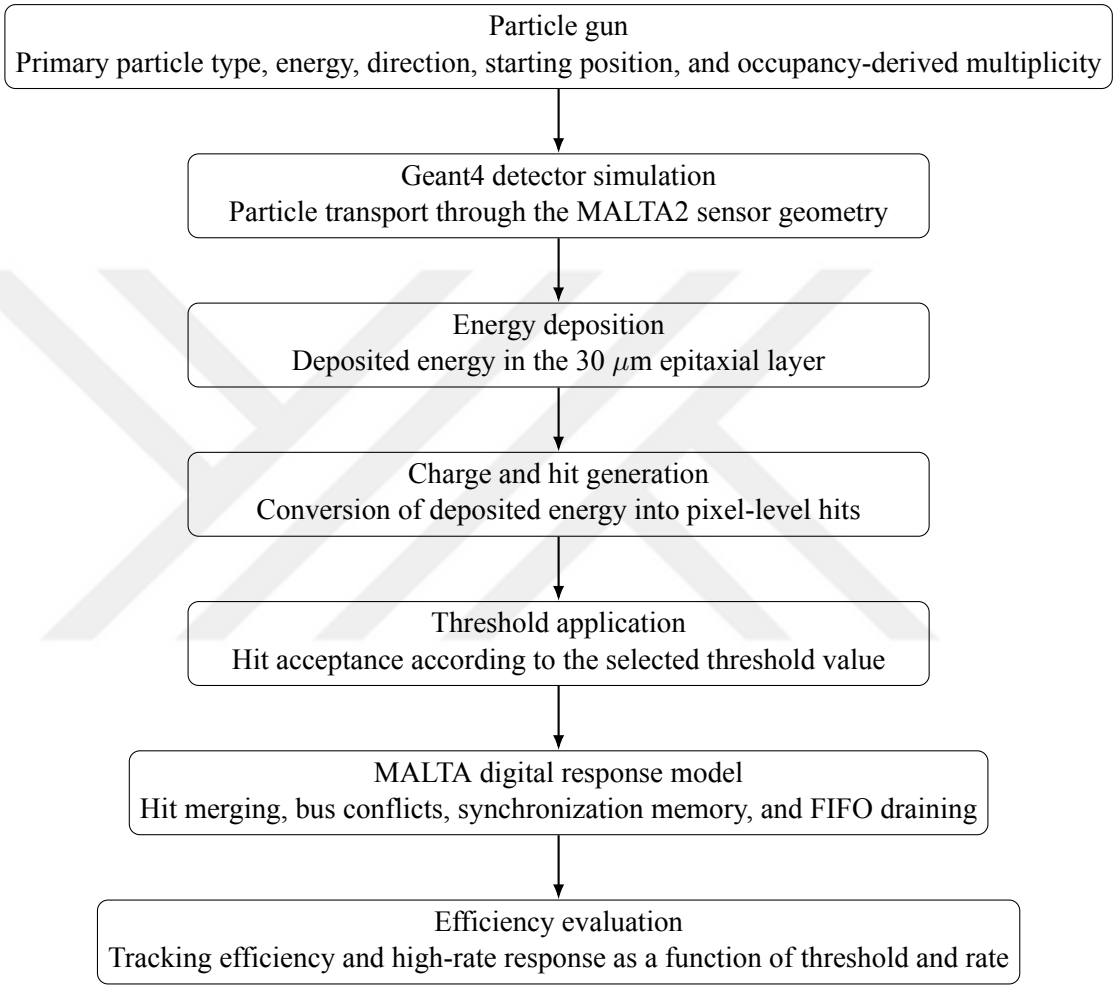


Figure 5.1 Simulation workflow used to evaluate the MALTA2 response under ATLAS ITk-derived occupancy and beam-rate conditions

The workflow summarized in **Figure 5.1** separates the simulation into physical and digital-response stages. The physical stage describes how primary particles are generated and transported through the MALTA2 sensor volume, while the digital-response stage describes how the resulting hits are interpreted by the MALTA readout model. This separation is useful because a loss of efficiency at high rate can have different origins. It may be caused by the charge-deposition and threshold response of the sensor, or it may arise from

digital-readout limitations such as hit merging, bus conflicts, synchronization-memory behaviour, and FIFO draining. Therefore, the simulation does not only estimate the number of particles crossing the sensor, but also provides a way to relate the particle environment to the final reconstructed MALTA response.

After the Geant4 energy-deposition step, the deposited charge is converted into sensor-level hits according to the MALTA2 pixel geometry and the applied threshold setting. The threshold is varied in the simulation in order to evaluate how the apparent tracking efficiency changes as the hit-detection condition becomes more or less restrictive. The generated hit information is then passed to the MALTA digital response model, where the effects of the asynchronous readout architecture can be evaluated.

Table 5.1 Main inputs and outputs of the MALTA2 high-rate simulation study

Item	**Description**
Simulation toolkit	Geant4-based particle-transport and energy-deposition simulation
Primary-particle definition	Particle-gun configuration defining particle type, energy, direction, starting position, and occupancy-derived multiplicity
Sensor configuration	MALTA2 geometry with a 30 μm epitaxial layer
Beam-rate study	120 GeV protons evaluated under different beam-rate conditions
Occupancy benchmark	ATLAS ITk-derived region-dependent occupancy inputs used for the replacement-feasibility evaluation
Particle multiplicity	Derived from ITk hit occupancy, scaled to the MALTA2 sensor area, and sampled using a Poisson distribution
Threshold scan	Simulated threshold variation used to study the efficiency dependence on hit-detection conditions
Digital-response effects	Hit merging, bus conflicts, synchronization-memory behaviour, and FIFO draining
Output quantity	Simulated MALTA2 tracking efficiency as a function of threshold, occupancy, and beam rate

In addition to the sensor-level response, the digital response model can include readout-related effects such as hit merging, bus conflicts, synchronization-memory behaviour, and FIFO draining. These effects are important because inefficiencies in high-rate environments may arise not only from charge collection in the sensor, but also from the finite

capacity and timing behaviour of the on-chip and external readout chain.

Table 5.1 summarizes the main inputs and outputs of the simulation study. The table also clarifies the role of the ATLAS ITk information in this chapter. The ITk input is used to define demanding occupancy conditions for evaluating the feasibility and limitations of MALTA2 in a possible replacement scenario, whereas the MALTA2 sensor and readout response are evaluated within the MALTA simulation and digital-response chain. Therefore, the comparison is focused on whether the present MALTA2 architecture remains efficient under ITk-derived high-rate conditions.

5.2 ATLAS ITk-based Occupancy Benchmark

The ATLAS ITk-based part of the study uses detector-region occupancy information as a benchmark input for MALTA2. The purpose of this step is to estimate how many particles would cross a MALTA2-sized sensor area during one 25 ns bunch crossing under different ITk-inspired occupancy conditions. The region labels used in the simulation identify the ITk-derived occupancy conditions used for the replacement-feasibility evaluation. They do not indicate that MALTA2 is implemented in the corresponding ATLAS ITk detector regions; rather, they define increasingly demanding operating scenarios against which the MALTA2 response is tested.

For the ITk-based benchmark, the particle multiplicity is therefore not fixed manually for each simulated event. Instead, it is sampled from the occupancy-derived expectation value, assuming a uniform particle distribution over the MALTA2 sensor area. This approach follows the simulation strategy used to convert ITk hit-occupancy information into an approximate particle-gun input for MALTA-family high-rate studies.

This distinction is important because MALTA2 and the baseline ATLAS ITk detector are different detector technologies. The comparison is used to connect the MALTA2 response to a realistic high-rate tracking-detector environment and to evaluate whether the sensor and readout response remain efficient as the local particle density increases. In this sense, the ITk occupancy information provides a practical reference point for assessing the limitations of MALTA2 with respect to a possible ITk-replacement scenario.

The occupancy benchmark also provides a bridge between the controlled laboratory me-

asurements and a more realistic tracking-detector environment. Laboratory injection measurements are useful because they isolate the readout chain under reproducible conditions. However, a tracking detector is exposed to particles distributed over space and time, and the resulting hit density can affect both the sensor response and the digital readout architecture. By using ITk-derived occupancy information as an external reference, the simulation study evaluates MALTA2 under conditions that are more demanding than a single controlled injection sequence.

The simulation should therefore be interpreted as an occupancy-driven MALTA2 replacement-feasibility benchmark rather than a full ATLAS detector-performance simulation. A complete detector-level evaluation, including full ATLAS geometry, reconstruction, and physics-performance effects, would require a dedicated ATLAS simulation framework.

5.3 Simulation Results

As shown in **Figure 5.2**, the simulated efficiency depends strongly on the occupancy condition associated with the detector region used as the benchmark input. The strip-region benchmark cases remain close to full efficiency over the studied threshold range, while the pixel-region benchmark cases show lower efficiency because of the higher local hit density. This indicates that the expected operating environment has a direct impact on the apparent tracking efficiency, even when the same sensor concept is considered.

The behaviour observed in **Figure 5.2** is consistent with the expected dependence of high-rate detector response on local occupancy. When the local particle density is low, the probability that multiple hits compete for the same readout resources within a short time interval is reduced. Under these conditions, the simulated efficiency remains close to full efficiency. When the local particle density increases, the probability of overlapping or competing hits also increases, and the response becomes more sensitive to both the threshold setting and the readout architecture.

These results indicate that the suitability of MALTA2 for an ITk-replacement scenario is strongly dependent on the local occupancy environment. Under lower-occupancy benchmark conditions, the simulated response remains close to full efficiency, suggesting that the sensor and readout concept can remain effective in less demanding regions. Under

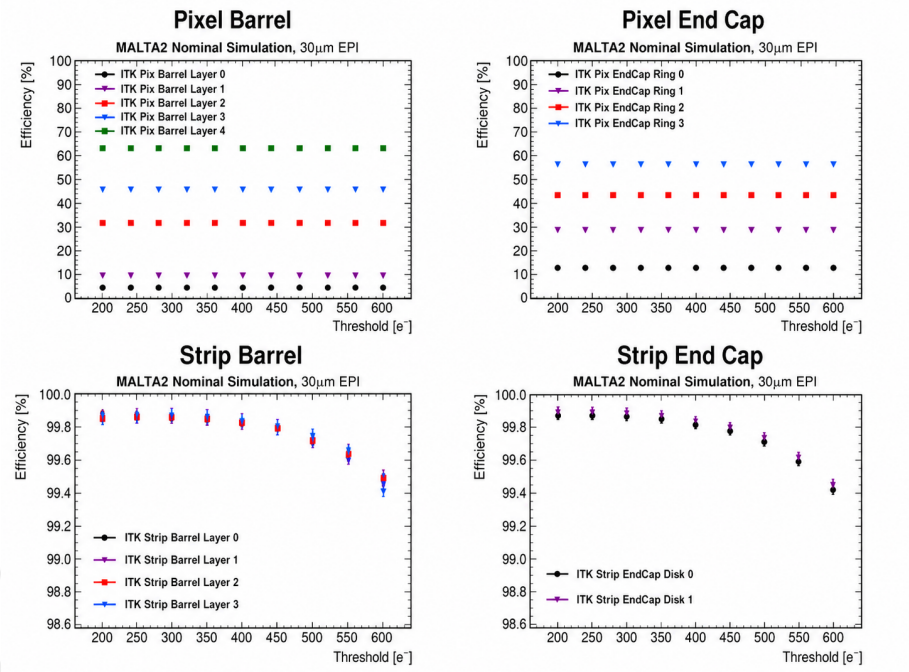


Figure 5.2 Simulated MALTA2 efficiency as a function of threshold for ATLAS ITk-derived occupancy conditions in different detector regions using a 30 μm epitaxial layer. The ITk region labels define the occupancy inputs used for the replacement-feasibility evaluation and do not indicate that MALTA2 is implemented in the baseline ATLAS ITk detector

higher-occupancy pixel-region benchmark conditions, however, the efficiency degradation shows that the present MALTA2 architecture would not be sufficient as a direct replacement without further improvements in the sensor and digital readout design.

Figure 5.3 shows the simulated MALTA2 tracking efficiency for different beam-rate conditions. The response remains close to full efficiency for the 100 MHz beam-rate condition over the studied threshold range. At higher beam rates, the efficiency becomes more dependent on the threshold setting. In the 125 MHz case, the efficiency increases with threshold and approaches the fully efficient regime, whereas the 142 MHz and 166 MHz cases remain lower over the same threshold range.

The beam-rate results indicate that the same sensor and digital-response configuration can behave differently depending on the incident particle rate. At moderate rate, the readout chain can process the generated hit information with limited efficiency loss. At higher rate, the time separation between successive hits is reduced, and the probability of readout-related loss mechanisms increases. Therefore, the simulated high-rate response should be interpreted as the combined effect of sensor-level hit generation and digital-readout

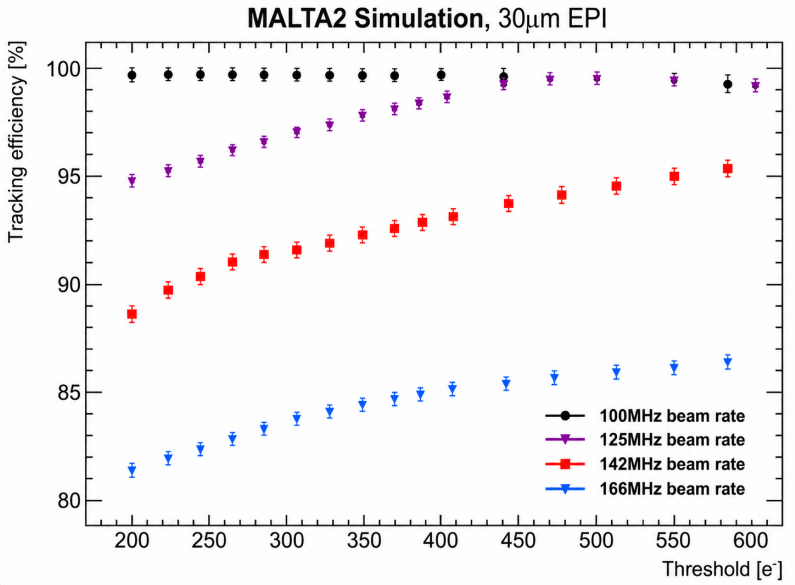


Figure 5.3 Simulated MALTA2 tracking efficiency as a function of threshold for different beam-rate conditions using 120 GeV protons and a 30 μm epitaxial layer

limitations rather than as a purely geometrical acceptance effect.

5.4 Interpretation of High-rate Effects

The simulation results show that improving the external FPGA readout chain is necessary, but not sufficient by itself, for maintaining full efficiency under increasingly demanding high-rate conditions. The migrated UltraScale readout improves timing capture, sampling stability, and data-transfer reliability. However, the remaining high-rate inefficiencies are also connected to sensor-level and on-chip digital-readout effects, including hit merging, bus conflicts, synchronization-memory limitations, and internal data-transfer constraints. This distinction is important for interpreting the results of the firmware migration. The laboratory measurements demonstrate that the migrated AXKU040 readout provides a more stable and efficient external readout platform for MALTA2 characterization. The simulation study, on the other hand, shows that the response under high-rate tracking conditions also depends on the internal sensor and digital-periphery architecture. Therefore, the upgraded firmware should be understood as a necessary readout improvement for reliable MALTA2 operation, while the ultimate high-rate performance remains determined by the combined behaviour of the sensor, on-chip readout architecture, and external FPGA-based

data acquisition chain.

From the perspective of ITk replacement evaluation, the simulation results do not support a direct replacement of the baseline ATLAS ITk pixel detector by the present MALTA2 architecture. Instead, they show that MALTA2 provides a useful monolithic sensor platform for studying high-rate readout limitations and for guiding future MALTA-family developments. In particular, the observed efficiency loss under the most demanding occupancy conditions points to the need for further chip-level architectural improvements before such a technology could be considered for an ITk-like replacement role.

The separation between external readout improvements and intrinsic high-rate limitations is also important for future MALTA developments. The migrated firmware provides a stable platform for operating and characterizing MALTA2, but high-rate tracking performance ultimately requires the sensor, digital periphery, buffering logic, and FPGA-based readout system to operate coherently. In this context, the simulation study supports the conclusion that firmware migration is an enabling step for reliable characterization, while further sensor- and architecture-level optimization remains necessary for operation in increasingly demanding high-occupancy environments.

6. DISCUSSION AND CONCLUSION

The results presented in this chapter show that the migration from the KC705 platform to the AXKU040 platform was not limited to a direct replacement of the FPGA board. Although the software-visible behaviour of the MALTA2 readout system was preserved, several internal firmware changes were required in order to adapt the design to the UltraScale architecture. These changes included the replacement of the input-delay and deserialization resources, the revision of the clocking structure, the hardware-assisted injection-control mechanism, and the simplification of the slow-control data path.

The tap-calibration and analog-scan results are consistent with each other. After timing adjustment, the AXKU040-based implementation provides a cleaner and more stable sampling response than the previous KC705-based implementation. This behaviour is reflected in the analog-scan measurements, where the measured hit count becomes more uniform across the full pixel matrix. Since the same MALTA2 chip and comparable operating conditions were used, the observed improvement is attributed primarily to the readout implementation rather than to a change in the intrinsic response of the sensor pixels.

The analog-scan results represent the clearest practical improvement of the migrated firmware. In the KC705-based implementation, a non-negligible fraction of pixels showed hit counts below the expected trigger count, indicating readout-side undercounting in parts of the matrix. In the AXKU040-based implementation, this effect was strongly reduced, and all pixels in the representative scan were measured within 5% of the expected count. The repeated-scan results further show that this improvement was stable over 1000 runs rather than being limited to a single measurement.

The threshold-scan results provide an important cross-check for the migration. The threshold distributions remain broadly consistent between the two implementations, indicating that the migrated firmware preserves the expected threshold-measurement behaviour. Therefore, the improved analog-scan uniformity should not be interpreted as a change in the underlying sensor threshold response. Instead, the larger number of contributing pixels in the AXKU040-based threshold measurement reflects the larger usable fraction of the matrix enabled by the improved readout stability.

The reduction in scan duration is also important for routine detector characterization. The

analog-scan time was reduced both in the representative comparison and in the repeated-run study, while the threshold-scan time was reduced by approximately a factor of two. This improvement is mainly associated with the hardware-assisted injection-control mechanism and the reduction of software–firmware communication overhead. As a result, repeated parameter scans and extended laboratory studies can be performed more efficiently.

The scope of these results should also be noted. The comparisons were performed using the same MALTA2 chip under comparable laboratory conditions and with injection-based measurements. Therefore, the conclusions are directly relevant to firmware-level readout performance and laboratory characterization workflows. Further validation with additional chips and beam-test or telescope measurements would be useful to evaluate the impact of the improved readout uniformity under experimental tracking conditions.

The high-rate simulation study further extends this interpretation by relating the MALTA2 readout response to ATLAS ITk-based occupancy conditions. The simulation results indicate that the migrated external readout provides a more stable basis for characterization, but high-rate tracking performance is still governed by the combined behaviour of the sensor, the on-chip digital architecture, and the external FPGA readout chain. This distinction is important because the firmware migration improves the readout infrastructure, while the remaining inefficiencies under more demanding occupancy conditions point to the need for further chip-level architectural developments in future MALTA designs.

Overall, the results indicate that the migrated firmware achieves the main objective of preserving compatibility with the existing MALTA2 DAQ environment while improving readout stability, matrix-wide response uniformity, and measurement efficiency. These improvements provide a stronger basis for continued MALTA2 characterization and for future developments of the MALTA readout chain.

Taken together, the discussion above establishes the broader significance of the firmware migration and provides the basis for summarizing the main conclusions of this thesis.

Reliable readout, stable operation, and efficient detector characterization are essential for monolithic pixel sensors used in high-energy physics studies. For MALTA2, these requirements are particularly important because the asynchronous readout architecture places stringent demands on the external readout system. In this thesis, an upgraded MALTA2 re-

adout firmware was presented through a behaviour-preserving migration from the Xilinx KC705 evaluation board to the ALINX AXKU040 Kintex UltraScale platform.

The migration was motivated by the end-of-life of the KC705 platform and by the need for a maintained replacement that could preserve compatibility with the existing MALTA2 DAQ software and operating procedures. The main objective was therefore not to redesign the complete data-acquisition chain, but to migrate the firmware to a new FPGA platform while keeping the DAQ-visible behaviour unchanged. This objective was achieved by preserving the established timing conventions, register-level interface, output data format, and slow-control procedure.

At the firmware level, the internal architecture was adapted to the UltraScale device family. The readout path was updated using `ISERDESE3` and `IDELAYE3` resources, the asynchronous oversampling scheme was revised, and the clocking structure was simplified to reduce skew at the deserializer interface. In addition, a hardware-assisted injection-control mechanism was introduced, and the slow-control path was simplified by replacing the previous fragmented register structure with a single streaming write interface.

The implemented AXKU040 firmware was successfully placed and routed with positive post-route timing margins. Timing closure was achieved with no failing setup, hold, or pulse-width endpoints, and the resource utilization remained within the available limits of the target FPGA. These results show that the migrated firmware can be implemented reliably on the AXKU040 platform and that sufficient implementation margin remains available for possible future extensions.

The functional measurements demonstrated that the migrated firmware preserves the expected readout behaviour while improving practical detector-characterization performance. Tap-calibration measurements showed a cleaner and more stable calibrated sampling response after timing adjustment. Analog-scan measurements showed a clear improvement in full-matrix readout uniformity, with the number of pixels within 5% of the expected trigger count increasing from 84 229 to 110 160, while the number of pixels below 90% of the expected response decreased from 9 565 to 0. The averaged results over 1000 repeated analog-scan runs further confirmed that this improvement was stable and reproducible rather than being limited to a single measurement.

Threshold-scan measurements remained broadly consistent between the KC705-based and

AXKU040-based implementations, indicating that the expected threshold-measurement behaviour was preserved. At the same time, the larger number of contributing pixels in the AXKU040-based measurement showed that threshold scans could be performed on a larger usable fraction of the matrix, since fewer pixels had to be excluded due to readout-side limitations.

The upgraded implementation also reduced the duration of routine characterization measurements. In the representative analog-scan comparison, the scan time was reduced from 35 s to 16 s. Over 1000 repeated analog-scan runs, the mean scan time was reduced from 38.31 s to 15.39 s, while the run-to-run timing stability was also improved. For threshold scans, the execution time was reduced from approximately 55 minutes to about 25 minutes. These improvements are mainly associated with the refined readout timing, the hardware-assisted injection-control mechanism, and the reduction of software–firmware communication overhead.

Overall, the AXKU040-based implementation provides a suitable long-term replacement for the KC705-based MALTA2 readout. It preserves compatibility with the existing software environment while improving readout uniformity, reducing masking requirements, and shortening measurement time in practical scan studies. Beyond the immediate platform migration, the design choices presented in this work provide a useful basis for future developments of the MALTA readout chain, including multi-chip operation and future members of the MALTA family such as MALTA3.

Beyond the specific case of MALTA2, the work presented in this thesis is also relevant to the broader field of accelerator technologies. Modern accelerator-based experiments require not only advanced accelerator structures, but also reliable detector systems, readout electronics, firmware infrastructures, and data-acquisition chains. In this respect, the development of FPGA-based detector readout systems contributes directly to the experimental capability needed around accelerator facilities. For Türkiye, where accelerator and radiation technologies are being developed through national research infrastructures, strengthening local expertise in detector readout, firmware design, timing alignment, and data acquisition is important for supporting future experimental studies. Therefore, this thesis also represents a contribution to the technical knowledge base required for accelerator-related detector and instrumentation developments in Türkiye.

REFERENCES

[1] CERN, "Layout of ATLAS," 1998. CERN Document Server, License: CC-BY-4.0.

[2] ATLAS Collaboration, "Atlas schematics: Inner tracker (itk)," 2026. https://cds.cern.ch/record/2908925.

[3] M. Standke, "RD53B Wafer Testing for the ATLAS ITk Pixel Detector," in *Proceedings of Technology and Instrumentation in Particle Physics 2021*, vol. 2374, p. 012087, 2022. ATL-ITK-PROC-2021-007, CERN CDS record 2775147.

[4] W. Snoeys *et al.*, "A Process Modification for CMOS Monolithic Active Pixel Sensors for Enhanced Depletion, Timing Performance and Radiation Tolerance," *Nucl Instrum Methods Phys Res A*, vol. 871, pp. 90–96, 2017.

[5] AMD, "KC705 Evaluation Kit Product Page," 2025. Accessed: October 30, 2025.

[6] Alinx, "AXKU040 Product Page," 2025. Accessed: October 30, 2025.

[7] W. R. Leo, *Techniques for Nuclear and Particle Physics Experiments: A How-to Approach*. Springer, 2 ed., 1994.

[8] R. Wigmans, *Calorimetry: Energy Measurement in Particle Physics*. Oxford University Press, 2 ed., 2017.

[9] ATLAS Collaboration, "The ATLAS Experiment at the CERN Large Hadron Collider," *Journal of Instrumentation*, vol. 3, p. S08003, 2008.

[10] ATLAS Collaboration, "The Inner Detector," 2026. ATLAS Experiment public page.

[11] I. Béjar Alonso *et al.*, *High-Luminosity Large Hadron Collider (HL-LHC): Technical Design Report*. No. CERN-2020-010, CERN Yellow Reports: Monographs, 2020.

[12] A. Heggelund *et al.*, "Overview of the ATLAS ITk Pixel Detector," *Journal of Instrumentation*, vol. 18, p. C02014, 2023.

[13] ATLAS Collaboration, "Expected Tracking Performance of the ATLAS Inner Tracker at the High-Luminosity LHC," *Journal of Instrumentation*, vol. 20, p. P02018, 2025.

[14] ATLAS Collaboration, "The ATLAS ITk Detector System for the Phase-II LHC Upgrade," *Nucl. Instrum. Methods Phys. Res. A*, vol. 1045, p. 167597, 2023. Open access under Creative Commons license.

[15] N. Nakoty, "ATLAS Pixel Detector Overview," *JINST*, vol. 19, p. C12022, 2024.

[16] C. Solans Sánchez *et al.*, "MALTA Monolithic Pixel Sensors in TowerJazz 180 nm Technology," *Nucl Instrum Methods Phys Res A*, vol. 1057, p. 168787, 2023.

[17] H. Pernegger *et al.*, "MALTA-Cz: A radiation hard full-size monolithic CMOS sensor with small electrodes on high-resistivity Czochralski substrate," *Journal of Instrumentation*, vol. 18, no. 09, p. P09018, 2023.

[18] B. Hiti *et al.*, “Development of the Monolithic MALTA CMOS Sensor for the ATLAS ITk Outer Pixel Layer,” in *Proceedings of Science*, 2019. PoS TWEPP2018 155.

[19] D. V. Berlea *et al.*, “Depletion Depth Studies with the MALTA2 Sensor, a Depleted Monolithic Active Pixel Sensor,” *Nucl Instrum Methods Phys Res A*, vol. 1063, p. 169262, 2024.

[20] G. Dash *et al.*, “Timing Characterization of MALTA and MALTA2 Pixel Detectors Using Micro X-ray Source,” *Journal of Instrumentation*, vol. 20, no. 9, p. P09039, 2025.

[21] L. Fasselt *et al.*, “Charge Calibration of MALTA2, a Radiation Hard Depleted Monolithic Active Pixel Sensor,” *Nucl Instrum Methods Phys Res A*, vol. 1082, p. 170972, 2026.

[22] D. Dobrijević *et al.*, “MALTA3: Concepts for a New Radiation Tolerant Sensor in the TowerJazz 180 nm Technology,” *Nuclear Instruments and Methods in Physics Research Section A*, vol. 1040, p. 167226, 2022.

[23] D. Dobrijević *et al.*, “Future Developments of Radiation Tolerant Sensors Based on the MALTA Architecture,” *Journal of Instrumentation*, vol. 18, no. 03, p. C03013, 2023.

[24] L. Fasselt *et al.*, “Charge Collection Parameterization of MALTA2, a Depleted Monolithic Active Pixel Sensor,” *Journal of Instrumentation*, vol. 21, no. 04, p. C04001, 2026.

[25] C. Ghabrous Larrea *et al.*, “IPbus: A Flexible Ethernet-Based Control System for xTCA Hardware,” *Journal of Instrumentation*, vol. 10, no. 02, p. C02019, 2015.

[26] Xilinx Inc., *UG474: 7 Series FPGAs Configurable Logic Block User Guide*, 2025. Accessed: March 19, 2026.

[27] Xilinx Inc., *DS890: UltraScale Architecture and Product Data Sheet: Overview*, 2026. Accessed: March 19, 2026.

[28] Xilinx Inc., *DS182: Kintex-7 FPGAs Data Sheet: DC and AC Switching Characteristics*, 2021. Accessed: March 19, 2026.

[29] Xilinx Inc., *DS892: Kintex UltraScale FPGAs Data Sheet: DC and AC Switching Characteristics*, 2020.

[30] Advanced Micro Devices, *Tri-Mode Ethernet MAC LogiCORE IP Product Guide (PG051)*, 2024. Version 9.0.

[31] Advanced Micro Devices, *1G/2.5G Ethernet PCS/PMA or SGMII LogiCORE IP Product Guide (PG047)*, 2025. Version 17.0.

[32] ALINX, *KINTEX UltraScale FPGA Development Board AXKU040 User Manual*, 2021. Part 9: Gigabit Ethernet Interface; Accessed: April 27, 2026.

[33] Microchip Technology Inc., *KSZ9031RNX Gigabit Ethernet Transceiver with RGMII Support Datasheet*, 2017. Accessed: April 27, 2026.

[34] Xilinx Inc., *UG571: UltraScale Architecture SelectIO Resources User Guide*, 2024. Accessed: March 20, 2026.

[35] Xilinx Inc., *UG974: Vivado Design Suite UltraScale Libraries Guide*, 2023. Accessed: March 20, 2026.

[36] Xilinx Inc., *UG471: 7 Series FPGAs SelectIO Resources User Guide*, 2018. Accessed: March 20, 2026.

[37] Xilinx Inc., *XAPP1315: LVDS Source Synchronous 7:1 Serialization and Deserialization Using Clock Multiplication*, 2017. Version 1.0, Release Date: April 15, 2017; Accessed: April 7, 2026.

[38] Xilinx Inc., *UG949: UltraFast Design Methodology Guide for FPGAs and SoCs*, 2025. Version 2025.1 English, Release Date: May 29, 2025; Accessed: April 7, 2026.