UNIVERSITÀ
DI PAVIA

DEPARTMENT OF ELECTRICAL, COMPUTER AND BIOMEDICAL ENGINEERING
MASTER'S DEGREE IN ELECTRONIC ENGINEERING, COURSE OF MICROELECTRONICS

"Analysis and Design of an FPGA-Based Data Acquisition System Using an External High-Speed ADC"

Dissertation of

Vivek Panseriya

Advisor:

Prof. Danilo Manstretta

Tutor:

PhD Abolhasan Ahmadihaji

(2025-2026)

Acknowledgments

First of all, I would like to express my deepest gratitude to my advisor, **Professor Danilo Manstretta**, for his invaluable mentorship, constant encouragement, and insightful guidance throughout the course of my master's thesis. His expertise, patience, and vision have been a source of continuous inspiration during my academic journey.

I would also like to sincerely thank my tutor, **PhD Abolhasan Ahmadihaji**, for his exceptional technical guidance, constructive feedback, and continuous support throughout this project. His in-depth knowledge and helpful suggestions were fundamental in the successful completion of this work.

My heartfelt appreciation extends to all the faculty and staff of the Department of Electronics, for providing an enriching learning environment and for their encouragement during my studies. I am also grateful to my colleagues and friends for their collaboration, discussions, and motivation during the development of this thesis.

Finally, I wish to thank my family for their unconditional love, patience, and support throughout my academic journey. Their belief in me has always been the driving force behind my success.

Contents

Abstract

Nowadays, the continuous advancement of embedded and biomedical electronics demands increasingly efficient and flexible data acquisition systems. In particular, the ability to accurately capture and analyze analog signals in real time is becoming essential for applications such as smart sensing, medical diagnostics, and IoT-enabled monitoring. To address these needs, field-programmable gate arrays (FPGAs) have emerged as a powerful platform for implementing high-speed and reconfigurable data processing architectures.

This thesis presents the design and implementation of an FPGA-based analog-to-digital data acquisition system that interfaces a 10-bit AD9215 ADC with the Terasic DE2-115 development board. The system samples analog signals at 60 MHz (Clock Frequency), writes the digital data into on-board SRAM memory through a deterministic finite-state machine, and subsequently transfers the stored samples to MATLAB for time-domain and frequency-domain analysis. The complete signal chain comprising clock generation, ADC sampling, data buffering, and memory writing was designed and verified using Verilog HDL and Quartus II tools.

Experimental results demonstrate the correct operation of the acquisition system and the accurate reconstruction of input waveforms. The MATLAB-based FFT analysis confirms the presence of the fundamental frequency and allows quantification of harmonic distortion, particularly the third harmonic component arising from front-end nonlinearities. The system therefore provides a reliable and flexible platform for characterizing ADC performance, evaluating signal quality, and validating data conversion concepts in real-time applications.

1. Chapter 1

1.1 Motivation

The interaction between the analog and digital domains has always been a central challenge in data acquisition systems[1].Designing an efficient analog front-end that interfaces seamlessly with digital logic requires special consideration of noise, bandwidth, and linearity. In many measurement and signal-processing applications such as biomedical instrumentation, industrial control, communication receivers, and laboratory data acquisition analog-to-digital conversion is the critical bridge between continuous and discrete signal domains. Traditionally, systems employ external ADCs optimized for fixed specifications, limiting flexibility once the hardware is integrated. This rigid approach often results in trade-offs among resolution, speed, and power consumption, and restricts the ability to adapt to varying operating environments or experimental conditions[1].

To address these limitations, this work explores a flexible, reconfigurable ADC system based on the AD9215, a 10-bit, 65–105 MSPS CMOS analog-to-digital converter. The AD9215 offers excellent signal integrity with 58 dB SNR, 77 dBc SFDR, and a 300 MHz analog input bandwidth, while operating from a single 3 V supply. Its differential sample-and-hold amplifier and on-chip voltage reference simplify system design, making it well-suited for FPGA-based mixed-signal acquisition. In this thesis, the DE2-115 development board (Cyclone IV FPGA) serves as the digital processing and control platform, providing high-speed I/O, PLL-generated sampling clocks, and on-board SRAM for data storage. The analog front-end, implemented through a differential amplifier stage and precision biasing network, ensures proper impedance matching and full-scale signal utilization.[2]

By combining the accuracy and low noise of the AD9215 with the real-time configurability of FPGA logic, this research aims to develop a soft-core, reconfigurable ADC interface capable of adaptable sampling, digital calibration, and efficient data buffering. This integration enhances flexibility, reduces system complexity, and supports rapid adaptation to new applications without requiring hardware redesign. The proposed architecture demonstrates how high-performance, low-cost FPGA-based systems can overcome traditional analog-digital boundaries, offering scalable solutions for next-generation sensing, communication, and signal-processing technologies.[3]

1.2 Introduction

In today's connected world, nearly every action from communication and sensing to automation and computation depends on the continuous exchange of data. From smartphones and biomedical sensors to industrial equipment and global data centers, modern systems produce massive volumes of information that must be captured, digitized, transmitted, and processed in real time. This rapid expansion of data-driven applications has placed unprecedented demands on the underlying electronic infrastructure, particularly on the components responsible for signal acquisition and conversion.

Digital systems operate on binary data, yet the physical world provides analog signals. Bridging this gap requires Analog-to-Digital Converters (ADCs)[4], which represent the heart of modern receivers. Whether in wireless communication, biomedical monitoring, automotive sensing, or IoT devices, ADCs enable the translation of real-world signals into digital form for subsequent processing. As system bandwidths increase and power budgets tighten, ensuring accurate and deterministic sampling becomes increasingly challenging.

Alongside ADC technology**,** Field-Programmable Gate Arrays (FPGAs) have emerged as powerful platforms for real-time data acquisition. With hardware-level parallelism, deterministic timing, and flexible interfacing capabilities, FPGAs are ideal for capturing high-speed ADC data, controlling memory, and performing pre-processing operations.

This thesis focuses on designing and implementing an FPGA-based data acquisition system using the 10-bit AD9215 ADC and the Terasic DE2-115 FPGA board. The system performs continuous, synchronous sampling at 60 MHz, stores data into on-board SRAM, and exports the captured samples for MATLAB-based time-domain and frequency-domain analysis.[5]

1.3 Thesis Organization

This thesis presents the design and experimental validation of an FPGA-based data acquisition system that samples an analog input using an external high-speed ADC and processes the captured data for time-domain and frequency-domain evaluation.

Chapter 1 introduces the motivation of the work, the project objectives, and the overall system concept, highlighting the importance of reliable analog-to-digital conversion and digital acquisition for measurement and signal analysis.

Chapter 2 provides the theoretical background required to understand the system, including sampling fundamentals (sampling frequency and Nyquist criterion), quantization effects

(resolution, LSB size, quantization noise), and the meaning of time-domain and frequency-domain representations. This chapter also outlines the FFT/DFT approach adopted for spectrum evaluation.

Chapter 3 describes the complete hardware setup. It includes the ADC signal chain, clocking strategy, FPGA board interface connections, and the role of the analog front-end. Practical considerations such as input amplitude scaling, bandwidth limitations, and noise sources are also discussed.

Chapter 4 details the FPGA digital design and implementation. It explains the data capture method, synchronization of ADC outputs, FIFO buffering strategy, SRAM writing control, and the top-level integration. The chapter also describes key timing/control signals and the logic used to guarantee correct sequential storage of samples.

Chapter 5 presents the data extraction and MATLAB-based post-processing. It describes the HEX data formatting, signed/unsigned conversion, offset removal, time-domain plotting, FFT computation, and frequency-axis scaling. Results are shown for different signal frequencies and amplitudes to validate correct sampling and system behaviour.

Chapter 6 discusses experimental results and validation. Time-domain waveforms and frequency spectra are analysed to confirm correct frequency identification, explain observed harmonics/spurs (if present), and relate measurement outcomes to ADC resolution and input amplitude settings.

Chapter 7 concludes the thesis by summarizing the achieved system performance, main contributions, limitations, and possible future improvements such as enhanced filtering, higher effective resolution, improved noise performance, and extended real-time processing.

1.4 Growth of Data in Modern Systems

The demand for faster, accurate, and energy-efficient data acquisition reflects broader global trends in data consumption. Reports from Cisco VNI and IDC indicate that worldwide data traffic is increasing exponentially, driven by mobile users, cloud computing, and IoT expansion.[6]

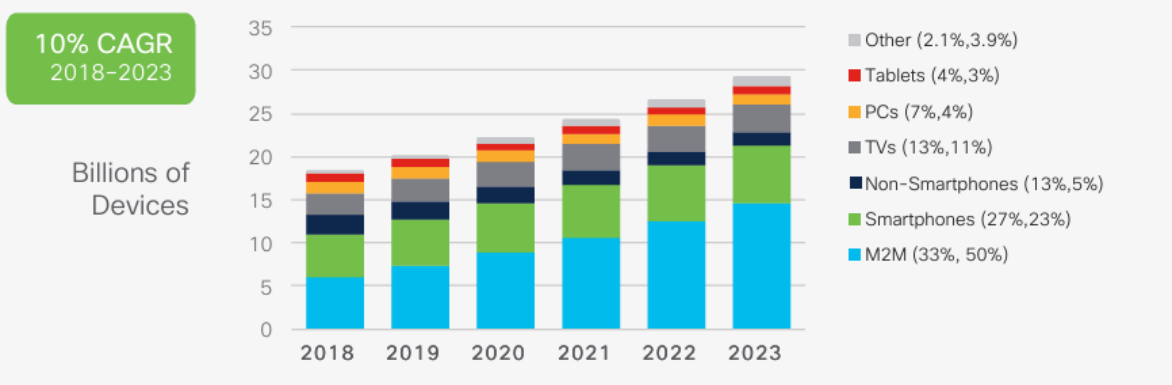


Figure 1. Cisco Annual Internet Report, 2018–2023

This explosive growth highlights the need for efficient and highly scalable data acquisition platforms, especially in systems that continuously monitor, digitize, and process real signals.

Keep in mind that global data consumption continues to rise rapidly, with each user demanding higher speeds, lower latency, and increasingly sophisticated services often at a reasonable cost. The COVID-19 pandemic accelerated these trends even further, as remote work, online education, streaming, and telemedicine expanded dramatically. At the same time, the rapid growth of the Internet of Things (IoT) has introduced billions of additional devices that continuously sense, transmit, and process data, creating new challenges for communication and computing infrastructures.

Despite the enormous number of end-users, most of the world's data traffic does not come from consumers. According to the Cisco VNI Global IP Traffic Forecast, more than 70% of global IP traffic originates inside data centers, where servers exchange, replicate, and analyze information on a massive scale. Because this internal traffic dominates overall global data flow, modern engineers must focus heavily on improving data center interconnections making them faster, more reliable, and significantly more energy-efficient.

Data centers consume vast amounts of power not only for computation but also for cooling, since the hardware must operate continuously and at high speed. This motivates the search for optimized electronic subsystems that can acquire, digitize, and transport data with minimal energy and maximal performance. As a result, even small improvements in key components such as ADCs, high-speed receivers, and real-time data acquisition systems can have a substantial impact on the efficiency of the entire infrastructure.

1.5 Role of ADCs in Modern Receivers

Modern transceivers, electrical, optical, or wireless share a similar front-end architecture. A typical receiver chain begins with analog conditioning, where amplifiers or equalizers prepare the input signal for digitization. This signal is then fed to the Analog-to-Digital Converter (ADC), which samples and quantizes the waveform so it can be processed in digital form. A general receiver architecture is illustrated in Figure 2.

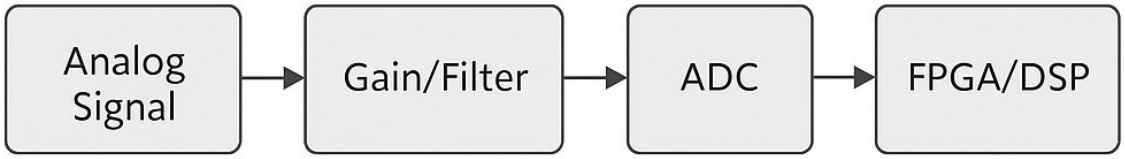


Figure 2.General Receiver Architecture Figure

The ADC plays a central role because it defines the maximum achievable bandwidth, accuracy, and distortion performance of the entire system. For example, the AD9215 10-bit ADC used in this thesis provides up to 105 MSPS sampling, 58 dB SNR, and 77 dB SFDR, enabling high-resolution capture of signals in the first and second Nyquist zones.[4]

Its internal sample-and-hold amplifier supports differential or single-ended operation and offers a 300 MHz analog input bandwidth, which makes it suitable for communication, ultrasound, and general-purpose high-speed acquisition.

In FPGA-based measurement systems, high-speed ADCs are typically interfaced directly with hardware logic so that sampling occurs with deterministic timing. This is essential for applications such as data centers, radar, medical imaging, and industrial sensing, where timing uncertainty directly degrades spectral quality. As demonstrated in literature, modern FPGA platforms combine parallelism, clock-management blocks, and FIFO memories to reliably capture and buffer high-speed ADC streams before storage or digital signal processing.[7]

Because even minor imperfections in sampling (clock jitter, aperture delay, nonlinear input stages) can lead to increased harmonic distortion, the ADC often becomes the bottleneck of overall system performance. For this reason, high-speed systems may use multiple ADC channels in parallel (time-interleaving) or rely on FPGAs to provide clock synthesis, synchronization, and calibration.

2. Chapter 2

2.1 Analog-to-Digital Conversion Fundamentals

In many real-world applications, physical signals such as sound, temperature, voltage, and electromagnetic waves naturally exist in analog form. However, modern computing and signal-processing platforms including computers, digital signal processors, and field-programmable gate arrays (FPGAs) operate using digital data. To enable communication between these two domains, an Analog-to-Digital Converter (ADC) is employed to translate continuous analog signals into discrete digital values.

The conversion process carried out by an ADC consists of two fundamental stages. First, the analog signal is sampled at regular time intervals determined by a sampling clock. Each sample captures the instantaneous amplitude of the input waveform at a specific moment in time. Second, the sampled value is mapped to the closest available digital level through a process known as quantization. The accuracy of these two stages directly determines how faithfully the digital representation reflects the original analog signal.

In this thesis, an external high-speed ADC is used to sample analog input signals and interface them with an FPGA-based digital system[7]. The digitized samples are subsequently buffered, stored in on-board memory, and exported for time-domain and frequency-domain analysis. Understanding the theoretical principles behind sampling and quantization is therefore essential for interpreting the experimental results presented in later chapters.[5]

2.2 Sampling Theory and Nyquist Criterion

Sampling is the process of converting a continuous-time signal into a discrete-time sequence by measuring its amplitude at uniform time intervals. The sampling frequency defines how often these measurements are taken and is one of the most critical parameters in any data acquisition system.

According to the Nyquist–Shannon sampling theorem, a band-limited signal can be perfectly reconstructed from its samples if the sampling frequency is at least twice the highest frequency component present in the signal. This minimum required sampling rate is commonly referred to as the Nyquist rate, while half of the sampling frequency is known as the Nyquist frequency.

If the sampling frequency is insufficient, frequency components above the Nyquist frequency fold back into the baseband, producing distortion known as aliasing. Once aliasing occurs, it cannot be removed through digital processing. For this reason, practical data acquisition systems must carefully select the sampling frequency and limit the input signal bandwidth using appropriate analog filtering.[4]

In the implemented system, the sampling clock is generated by the FPGA using an internal phase-locked loop (PLL). This approach ensures deterministic timing and stable clock generation, which is particularly important in high-speed acquisition systems where clock jitter can significantly degrade signal quality.

2.3 Quantization and ADC Resolution

After sampling, each analog value must be represented using a finite number of digital codes. This process, known as quantization, introduces an approximation error because the continuous input amplitude is mapped to discrete levels.

For an N-bit ADC, the full-scale input range is divided into (2^N) quantization levels. The smallest voltage difference that can be resolved is referred to as the Least Significant Bit (LSB). Any variation in the input signal smaller than one LSB cannot be represented in the digital domain[4].

Quantization inherently introduces an error between the actual analog value and its digital representation. This error is unavoidable and is commonly modeled as quantization noise. Increasing the ADC resolution reduces the size of the LSB and improves amplitude accuracy; however, it also increases sensitivity to noise, non-linearity, and clock imperfections.

In this project, the ADC resolution plays a significant role in determining the appearance of the sampled waveform, particularly when low-amplitude signals are applied. Under such conditions, the quantization process becomes more visible in digital output.

2.4 Quantization Effects and MSB and LSB Behavior

When the input signal amplitude is sufficiently large relative to the full-scale range of the ADC, the digital output spans a wide range of codes. In this operating condition, both the Most Significant Bits (MSBs) and Least Significant Bits (LSBs) vary continuously, and the reconstructed digital waveform closely resembles the original analog signal.[4]

As the input signal amplitude decreases, fewer quantization levels are utilized. In this case, the MSBs may remain constant while only the LSBs toggle. As a result, the time-domain waveform may appear staircase-shaped or square-like instead of smoothly sinusoidal. This effect is a direct consequence of finite resolution and does not indicate a malfunction of the ADC or the FPGA-based acquisition system.[8]

When the amplitude is reduced further, the digital output may toggle between only a small number of adjacent codes around the mid-scale value. This behavior was clearly observed during the experimental measurements conducted in this work and is fully consistent with theoretical expectations for low-amplitude signal digitization.

2.5 Time-Domain Representation of ADC Data

Time-domain analysis provides a direct visualization of how the ADC output varies with time. By plotting the sequence of sampled digital values, it is possible to observe waveform shape, amplitude variation, and quantization effects.

In systems where the sampling frequency is much higher than the input signal frequency, many samples are collected per signal period. This oversampling improves time resolution and allows accurate observation of waveform transitions. However, when the signal amplitude is small, the discrete nature of quantization becomes more apparent, leading to flat regions or abrupt transitions between digital codes.

Time-domain plots are particularly useful for validating correct data capture, ensuring synchronization between the ADC and FPGA logic, and confirming that no samples are missing or duplicated during acquisition.[7]

2.6 Frequency-Domain Representation and FFT Analysis

While time-domain analysis reveals how a signal evolves over time, frequency-domain analysis provides insight into the spectral content of the signal. This is achieved by applying the Discrete Fourier Transform (DFT) to the sampled data, typically implemented using the Fast Fourier Transform (FFT) algorithm.

The FFT decomposes the sampled waveform into its constituent frequency components, allowing identification of the fundamental frequency, harmonics, and noise contributions. For a correctly sampled sinusoidal input, the frequency spectrum should exhibit a dominant peak at the input signal frequency.

In this thesis, FFT analysis is performed in MATLAB on data extracted from the FPGA's on-board memory. The frequency-domain results are used to verify correct sampling behavior, confirm the input signal frequency, and evaluate the effects of quantization and analog front-end non-idealities.[7] Observed harmonic components are consistent with theoretical predictions and the characteristics of the implemented system

2.7 Summary

This chapter presented the theoretical background required to understand the operation and performance of the FPGA-based data acquisition system developed in this thesis. Fundamental concepts such as sampling theory, quantization, resolution limits, and time-domain and frequency-domain signal representations were discussed in direct relation to the implemented system.[7]

These principles provide the foundation for the hardware architecture described in the next chapter, where the analog front-end, ADC interfacing, clock generation, and FPGA-based control logic are presented in detail.

3. Chapter 3

3.1 Introduction

This chapter presents the complete hardware architecture of the proposed FPGA-based data acquisition system. The goal of the hardware design is to reliably capture analog signals, convert them into digital samples with controlled timing, and store the acquired data for subsequent offline analysis. The chapter describes each hardware block involved in the signal chain, starting from the signal generator used as input source, through the analog front-end and analog-to-digital converter, up to the FPGA-based digital logic and external memory.

The hardware implementation has been designed with particular attention to timing determinism, signal integrity, and repeatability of measurements. All design choices are motivated by the requirements of high-speed data acquisition and by the specifications provided in the component datasheets.

3.2 System-Level Architecture Overview

The implemented data acquisition system is composed of the following main blocks:

- A laboratory signal generator providing a controlled analog input signal
- n analog front-end based on a differential amplifier
- A high-speed external ADC (AD9215)
- Input synchronization logic using external flip-flops
- FPGA-based digital logic for buffering and control
- External SRAM for sample storage

The system architecture follows a unidirectional data flow, where the analog signal is progressively converted, buffered, and stored. This modular approach simplifies debugging, improves reliability, and allows each block to be independently validated.[2]

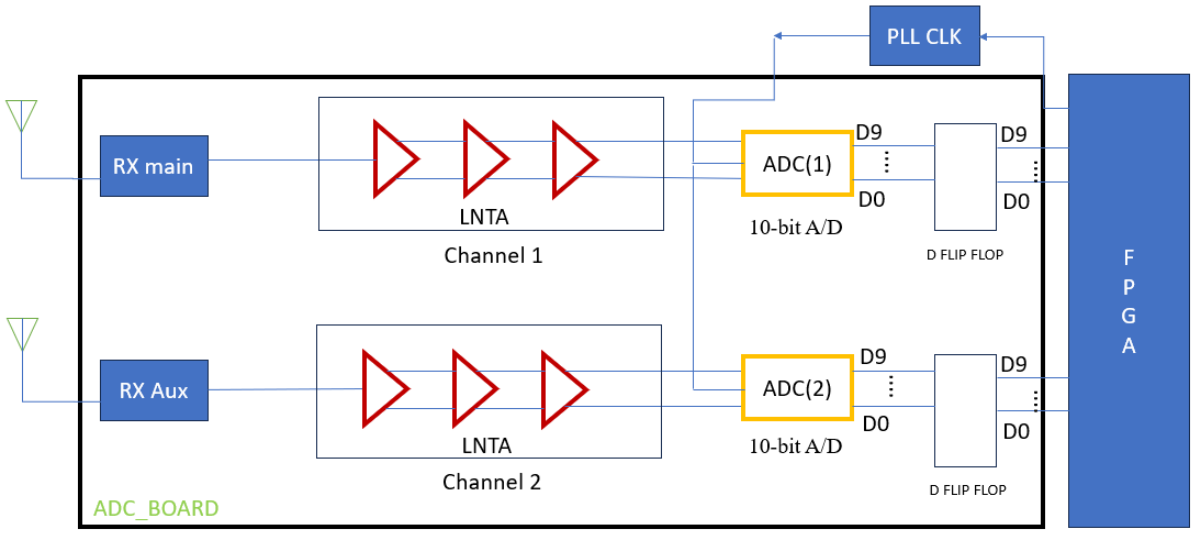


Figure 3. ADC Board

Figure 3. illustrates the internal organization of the ADC board and its interface with the FPGA. Each input channel consists of an analog reception stage followed by a chain of amplification blocks, which condition the incoming signal before digitization. The amplified signal is then converted into a 10-bit digital word by the ADC.

To ensure reliable timing and data integrity, the ADC outputs are registered using D-type flip-flops before entering the FPGA. This synchronization stage reduces timing uncertainty and improves robustness at high sampling rates. A phase-locked loop (PLL) generates the sampling clock, which is distributed to both the ADCs and the FPGA logic, guaranteeing synchronous operation across the entire acquisition chain. Although the diagram depicts a dual-channel configuration, the same architecture is applicable to the single-channel acquisition used in this work.

3.3 Low-Noise Transconductance Amplifier (LNTA)

The Low-Noise Transconductance Amplifier (LNTA) represents the first active stage in the analog signal chain and plays a critical role in determining the overall signal quality of the acquisition system. Its primary function is to amplify the incoming analog signal while introducing minimal additional noise, thereby preserving the signal-to-noise ratio (SNR) prior to digitization.

In the implemented architecture, the LNTA stage provides controlled gain to scale the input signal to a level suitable for the ADC full-scale range. By operating in the transconductance domain, the LNTA converts the input voltage signal into a proportional current, which can then be processed by subsequent amplification and conversion stages. This approach offers improved linearity and bandwidth performance compared to purely voltage-mode amplification.

Another important function of the LNTA is its contribution to bandwidth shaping. Although not designed as an explicit low-pass filter, the LNTA exhibits a finite frequency response that naturally attenuates high-frequency components. This behaviour helps reduce out-of-band noise and limits the spectral content presented to the ADC, thereby mitigating aliasing effects during sampling.

From a system perspective, placing the LNTA before the ADC ensures that the noise contribution of later stages, including the ADC input circuitry, is minimized. As a result, the LNTA strongly influences the achievable dynamic range and spectral purity of the digitized signal. The effectiveness of this stage is reflected in the experimental results presented later, where clean time-domain waveforms and well-defined frequency spectra are observed for properly scaled input signals.

Although the conceptual diagram illustrates a dual-channel LNTA configuration, the same principles apply to the single-channel acquisition used in this work. The LNTA stage thus forms an essential bridge between the external signal source and the digital acquisition system.

3.4 Signal Generator as Input Source

A laboratory-grade signal generator is used as the input signal source for the acquisition system. The signal generator provides sinusoidal test signals with programmable frequency and amplitude, enabling controlled and repeatable experiments. This choice eliminates uncertainty related to uncontrolled signal sources and allows systematic evaluation of the system performance under different operating conditions.

In the experimental setup, input frequencies ranging from tens of kilohertz to a few megahertz are applied. The signal amplitude is adjusted to study the impact of quantization, resolution limits, and harmonic distortion. Using a signal generator ensures that observed effects in the time and frequency domains originate from the acquisition system itself and not from variations in the input source.

3.5 Analog Front-End Design

Before digitization, the analog input signal is processed by an analog front-end stage. The purpose of this stage is to adapt the signal generated by the signal generator to the electrical requirements of the ADC input.[9]

The analog front-end is implemented using the LMH6550 differential amplifier. This component performs several critical functions:

- Conversion from single-ended to differential signaling
- Signal amplification and scaling to match the ADC full-scale input range
- Bandwidth limitation that reduces out-of-band noise and high-frequency interference

Although the LMH6550 is not a dedicated low-pass filter, its finite bandwidth and closed-loop gain response introduce a natural attenuation of higher-frequency components. This behavior contributes to reducing aliasing effects and helps explain the observed attenuation of signals at higher input frequencies during experimental measurements.

Analog LPF	4^{th} order Butterworth f_c = 100kHz-2MHz
ADC	10 bits f_s = 60MHz (clock)
FPGA SRAM	16 Bits (DE2 115) FPGA

Table 1.SUMMARY BASEBAND SPECIFICATIONS

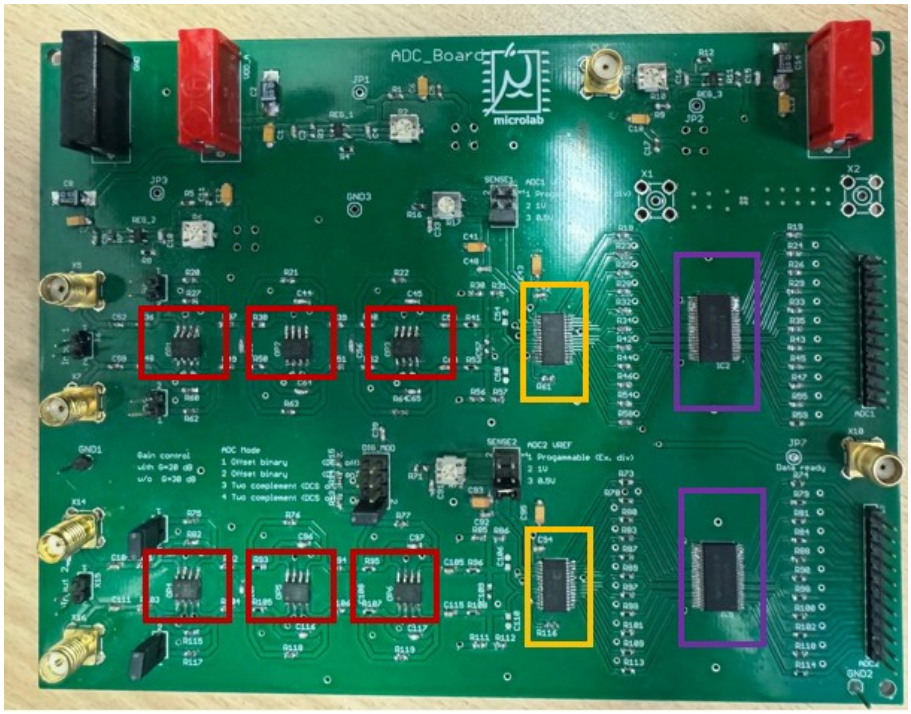


*Figure 4 ADC Board 1. Amplifier **LMH6550** 2.10-bit ADC **AD9215** 3.D-TYPE FLIP-FLOPS **SN54LVTH16374***

3.6 AD9215 Analog-to-Digital Converter

The AD9215 is a 10-bit high-speed analog-to-digital converter capable of sampling rates up to 105 MSPS. It features a differential sample-and-hold amplifier, an internal voltage reference, and a wide analog input bandwidth of approximately 300 MHz[4]

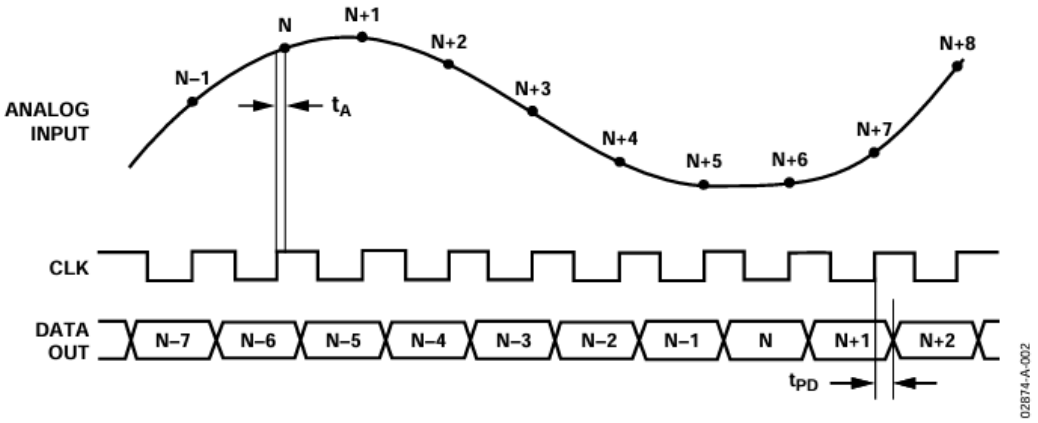
In this system, the AD9215 operates at a sampling frequency of 60 MHz The ADC converts the conditioned analog input signal into 10-bit parallel digital data. The choice of a 60 MHz sampling clock provides sufficient time resolution while maintaining compatibility with the FPGA interface and external memory timing constraints.

The resolution and sampling frequency of the ADC directly influence the accuracy of the captured waveform and the achievable frequency-domain performance. Careful adherence to the ADC datasheet specifications ensures correct operation and reliable interpretation of the acquired data.

3.7 Clock Generation and Distribution

Accurate clock generation is a fundamental requirement for high-speed data acquisition systems. In this work, the sampling clock for the ADC is generated using the Phase-Locked Loop (PLL) resources available on the FPGA.

The PLL derives a stable 60 MHz clock from the on-board 50 MHz reference oscillator of the DE2-115 board. The generated clock is routed both to the ADC and to the FPGA logic responsible for capturing the ADC output data. This shared clocking strategy ensures synchronous sampling and minimizes timing uncertainty between the ADC and FPGA domains.[2], [5]



3.8 ADC Output Synchronization Using Flip-Flops

To improve timing robustness and data integrity, the parallel output data of the ADC is registered using external D-type flip-flops (SN74LVTH16374). These flip-flops capture the ADC outputs on the sampling clock edge and provide stable, time-aligned signals to the FPGA inputs.[10]

This additional synchronization stage reduces the risk of metastability and timing violations at the FPGA interface, particularly when operating at high sampling frequencies. By ensuring that ADC data is properly aligned before entering the FPGA, the reliability of subsequent digital processing is significantly improved.[10]

3.9 FPGA Platform and Digital Interface

The digital processing and control logic is implemented on the Terasic DE2-115 development board, which is based on an Altera Cyclone IV FPGA. The FPGA serves as the central controller of the acquisition system and performs the following functions:[2]

- ❖ Capturing synchronized ADC output data
- ❖ Buffering samples using FIFO memory
- ❖ Controlling write operations to external SRAM
- ❖ Managing system timing and control signals

The FPGA architecture allows parallel processing and deterministic timing, which are essential for reliable high-speed data acquisition.[2]

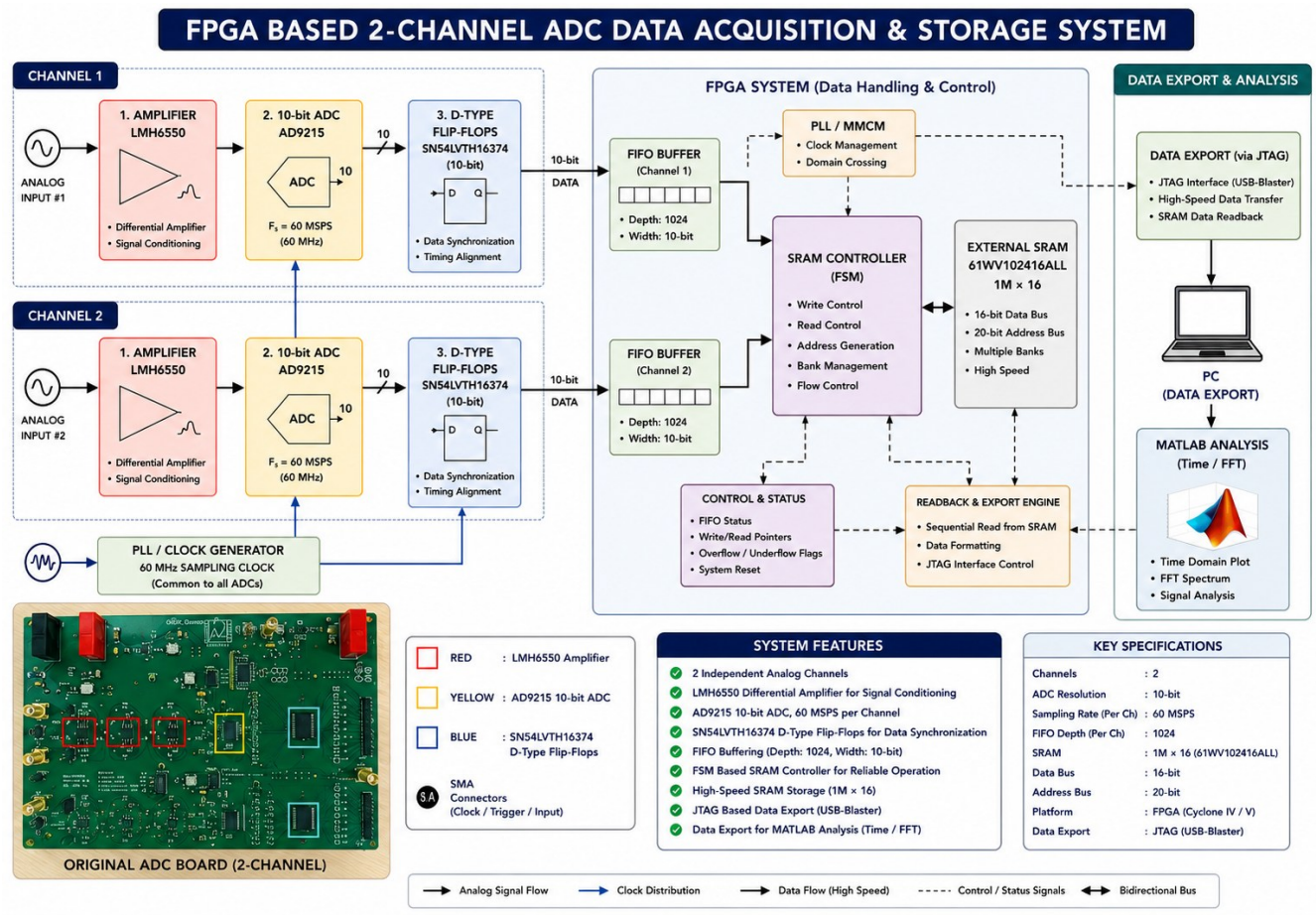


Figure 5. Digital Interface

3.10 FIFO Buffering Strategy

Due to the continuous and deterministic nature of analog-to-digital conversion, ADCs generate digital samples at a fixed rate dictated by the sampling clock. In contrast, external memory devices such as SRAM operate with finite access times and may not always be ready to accept new data on every clock cycle. Directly writing ADC samples to external memory without intermediate buffering can therefore lead to data loss, timing violations, or unstable system behaviour.[3], [11]

To address this issue, a First-In–First-Out (FIFO) buffer is inserted between the ADC interface logic and the SRAM controller. The FIFO acts as a temporary storage element that decouples the high-speed ADC sampling process from the memory write operations. ADC samples are continuously written into the FIFO at the sampling rate, while the SRAM controller independently reads data from the FIFO whenever the memory is available to perform a write operation.[7]

This architectural separation allows the acquisition system to tolerate short memory access delays without interrupting the sampling process. As a result, the FIFO-based buffering strategy guarantees lossless data capture, preserves the temporal order of samples, and ensures stable system operation even under high sampling rates. Similar buffering approaches are widely adopted in FPGA-based data acquisition systems to manage timing mismatches between high-speed converters and external memory devices.

3.11 SRAM Memory

The digitized data acquired from the ADC is stored in the external static random-access memory (SRAM) available on the Terasic DE2-115 development board. The IS61WV102416ALL SRAM device, organized as 1024K × 16 bits, is selected for this purpose due to its deterministic access timing, asynchronous operation, and simple control interface. Unlike dynamic memories, SRAM does not require refresh cycles, making it particularly suitable for continuous and sequential data storage in real-time acquisition systems.[12]

The SRAM is interfaced with the FPGA through a dedicated memory controller implemented as a finite-state machine (FSM). This controller is responsible for generating the memory address, driving the data bus, and asserting the appropriate control signals, including chip enable, write enable, and byte enable lines. During the acquisition phase, ADC samples are written sequentially into memory locations, ensuring that the temporal order of the sampled data is preserved.

To match the effective resolution of the acquisition system, the lower byte of the 16-bit SRAM data bus is used to store the ADC output samples. The FSM guarantees that all setup and hold

timing constraints specified in the SRAM datasheet are respected during each write cycle, thereby ensuring reliable and error-free memory operation.[2], [12]

Once the acquisition process is completed, the stored data can be read back from the SRAM and transferred to an external computer for offline processing. This approach enables detailed time-domain and frequency-domain analysis of the acquired signals using MATLAB, while maintaining a simple and deterministic hardware architecture. The use of external SRAM therefore provides an effective and robust solution for high-speed data storage in the proposed FPGA-based acquisition system.[2]

Because ADC sampling is continuous and operates at a relatively high speed, writing samples directly to external memory can lead to data loss when memory access is temporarily unavailable. To avoid this problem, modern receiver architectures typically include an intermediate buffering stage between the ADC and the memory system. As described in the low-power receiver architecture presented in [7], buffering is an effective way to decouple the fast-Sampling process from slower digital control and memory operations.[13]

In this work, a FIFO buffer is implemented between the ADC interface and the SRAM controller inside the FPGA. ADC samples are written into the FIFO at the sampling clock rate, while the SRAM controller reads data from the FIFO whenever the memory is ready. This approach ensures continuous and reliable data capture without sample loss, even in the presence of memory access delays. The use of FIFO-based buffering is a well-established technique in FPGA-based data acquisition systems and contributes significantly to the stability and robustness of the proposed design.

3.12 Data Extraction and Experimental Interface

After acquisition, the stored SRAM data is transferred to a host computer using the FPGA development tools. The extracted data is saved in hexadecimal format and processed using MATLAB.[4]

This offline processing approach simplifies the FPGA design and allows flexible analysis in both the time and frequency domains. The MATLAB environment is used to reconstruct waveforms, remove DC offsets, and compute Fast Fourier Transforms (FFT).[8]

3.13 Summary

This chapter described the complete hardware architecture of the implemented FPGA-based data acquisition system. Each hardware component, from the signal generator and analog front-end to the ADC, FPGA logic, and external SRAM, was discussed in detail.

The design choices emphasize timing determinism, signal integrity, and modularity. The described hardware platform provides a reliable foundation for the digital design and experimental analysis presented in the following chapter.

4. Chapter 4

4.1 Overview of the FPGA-Based Architecture

This chapter describes the digital implementation of the proposed data acquisition system on the Terasic DE2-115 FPGA platform[14]. The FPGA serves as the central element of the system, coordinating the acquisition of ADC data, buffering the samples, and storing them into external SRAM memory for subsequent offline analysis. All digital modules were designed in Verilog HDL and synthesized using the Intel Quartus II design environment.

The primary goal of the FPGA design is to ensure reliable and deterministic handling of high-speed ADC data while maintaining flexibility for post-processing and evaluation. Similar FPGA-based mixed-signal measurement architectures have been widely adopted to enable accurate signal acquisition and adaptive analysis in real-time systems. In this work, the FPGA architecture is organized into modular blocks, including clock generation, ADC interface logic, FIFO buffering, and an SRAM controller.

In this project the complete digital architecture of the data acquisition system was implemented in Verilog HDL using the Quartus II design environment on the Terasic DE2-115 FPGA board. The FPGA is responsible for generating the ADC sampling clock, capturing the ADC output data, buffering the samples using a FIFO, and storing them sequentially into external SRAM memory.

The Verilog design is organized into two main modules developed for this work:

- top.v - In Quartus II integrates clock generation, ADC interface, FIFO, and SRAM controller.
- sram_writer.v - In Quartus II implements the finite-state machine responsible for SRAM write operations.
- Additionally, a FIFO IP core generated in Quartus is used for buffering.

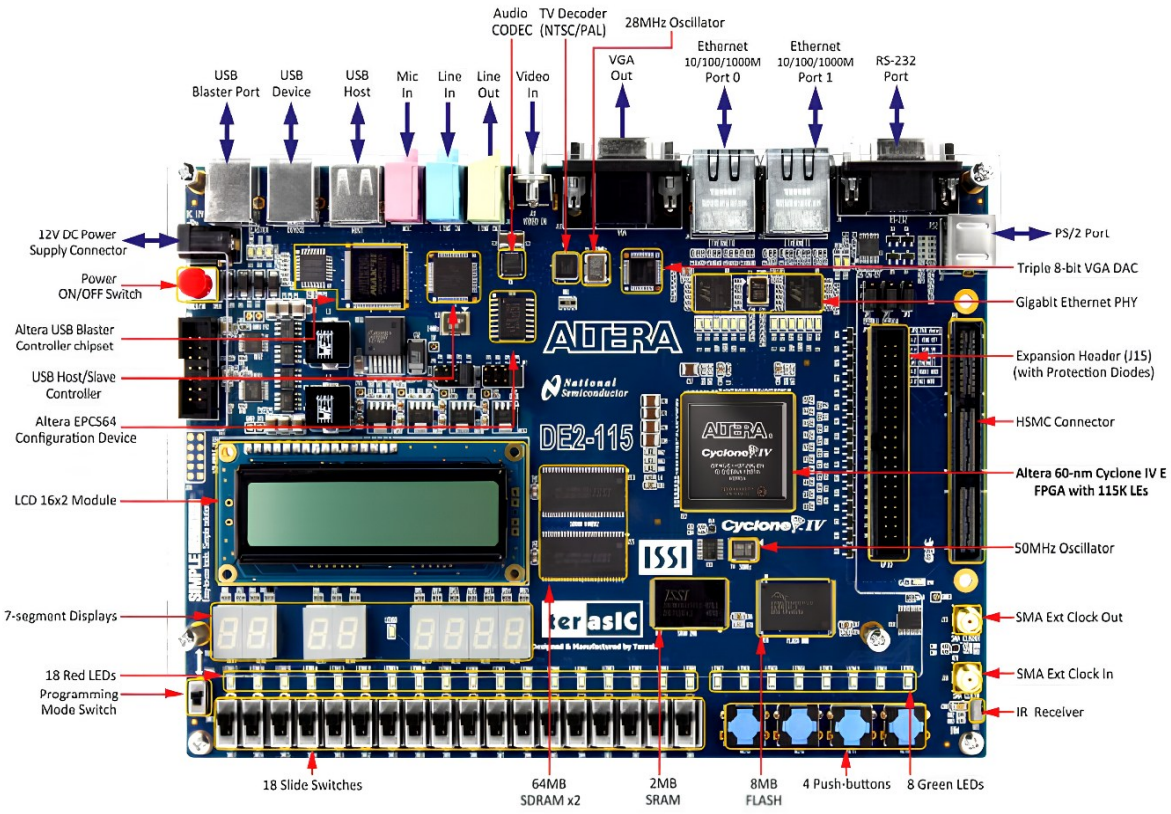


Figure 2-1 The DE2-115 board (top view)

Figure 6 Top view DE2 115 FPGA Board

4.2 Clock Generation Using PLL

Accurate clocking is essential for reliable ADC data capture.[7] The DE2-115 board provides a 50 MHz reference oscillator, which is processed using an on-chip PLL to generate a 60 MHz sampling clock.[2] This clock is sent to the ADC as the sampling clock and is also used internally by the FPGA logic.

Using the same clock source for both the ADC and the FPGA ensures synchronous operation and eliminates timing mismatches between data generation and data capture. This shared clocking strategy simplifies the design and guarantees stable sampling behavior.

The ADC sampling clock is generated using a PLL instantiated inside the top module. The DE2-115 provides a 50 MHz reference clock, which is converted into the required sampling clock.

- Clock Generation Using PLL (Verilog code)

```
pll_60MHz pll_inst (
.areset(~reset_n),
.inclk0(CLOCK_50),
.c0(clk_60MHz),
.locked(pll_locked)
);
```

This generated clock is:

- Sent to the ADC through SMA_CLKOUT on FPGA Altera Board.[2]
- Used internal SRAM (Static Random-Access Memory on the FPGA to capture ADC data)

Using the same clock for both ADC and FPGA guarantees synchronous sampling and eliminates timing mismatch.

4.3 ADC Data Capture and Synchronization

The AD9215 ADC provides parallel digital output synchronized with the sampling clock. To ensure reliable timing[5] and avoid metastability, the ADC outputs are first registered using flip-flops before being processed by the FPGA logic.

This registration stage aligns the incoming data with the FPGA clock and provides a stable data stream for further processing.[2] The registered ADC data is then forwarded directly to the FIFO buffer.[7]

The DE2-115 Board provides one 40-pin expansion header. The header connects directly to 36 pins of the Cyclone IV E FPGA, and also provides DC +5V (VCC5), DC +3.3V (VCC3P3), and two GND pins. Figure 6 shows the I/O distribution of the GPIO connector. The maximum power consumption of the daughter card that connects to GPIO[2] port is shown in Table 2.

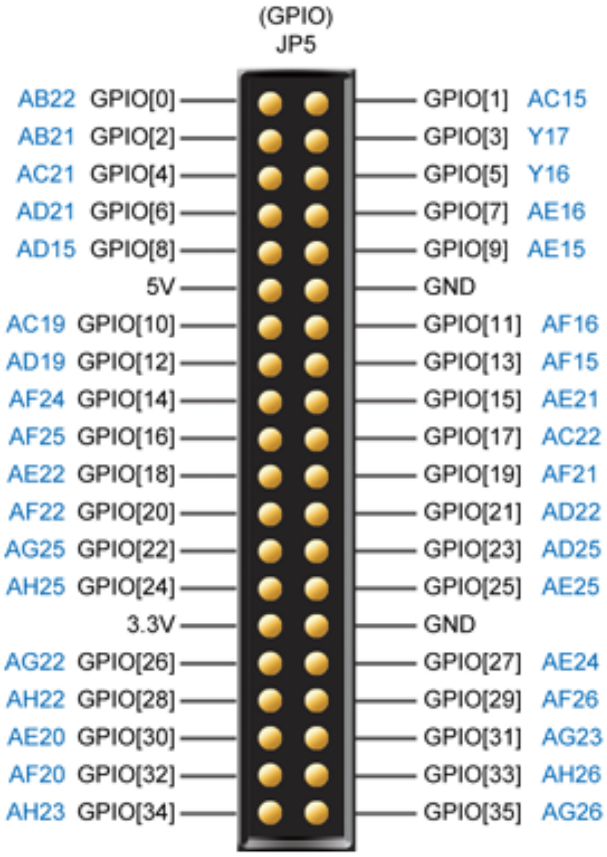


Figure 7 GPIO Pin Arrangement

Table 2 Power Supply of the Expansion Header

Supplied Voltage	Max. Current Limit
5V	1A
3.3V	1.5A

The ADC outputs parallel digital data through GPIO pins. This data is first registered inside the FPGA to ensure stable timing.

Verilog code:

```
always @ (posedge clk_60MHz or negedge reset_n) begin
    if (!reset_n)
        adc_data_ff <= 10'd0;
    else
        adc_data_ff <= GPIO_IN;
end
```

This registered data is then formatted and sent to the FIFO. Here FIFO data is 10bit.

Verilog code:

```
assign fifo_din = {6'b0, adc_data_ff};
assign fifo_wr_en = ~fifo_full;
```

This stage guarantees proper alignment of ADC data with the FPGA clock.[2], [7]

4.4 FIFO Buffer Integration

ADC sampling is continuous and occurs at a fixed rate of 60 MHz. However, external memory operations do not always occur at the same rate due to control overhead and access timing. Writing ADC data directly to SRAM could therefore result in data loss.

To prevent this, a FIFO buffer generated using the Quartus IP core is inserted between the ADC interface and the SRAM controller. ADC samples are written into the FIFO at the sampling rate, while the SRAM controller reads data from the FIFO whenever memory is ready to accept new data.

This buffering stage decouples the fast-Sampling process from the memory write process and guarantees lossless data acquisition.

A FIFO generated using the Quartus IP core is instantiated in the top module:

Verilog code[15]:

```
fifo fifo_inst (
    .clock(clk_5MHz),
    .data(fifo_din),
    .wrreq(fifo_wr_en),
    .rdreq(fifo_rd_en),
    .q(fifo_dout),
    .empty(fifo_empty),
    .full(fifo_full)
);
```

The FIFO continuously stores ADC samples and prevents data loss when the SRAM controller is busy.

4.5 SRAM Controller Implemented as FSM

The data stored in the FIFO is written into the external IS61WV102416ALL SRAM using a controller implemented as a finite-state machine (FSM). The FSM generates all required control signals, including memory address, data bus enables, chip enable, and write enable signals according to the SRAM timing requirements.

During operation, the controller:

- ❖ Reads a sample from the FIFO,
- ❖ Writes it to the current SRAM address,
- ❖ Increments the address for the next sample.

This sequential write process ensures that samples are stored in the exact order they were acquired, preserving the time relationship between them.

Tri-state data bus control:

```
assign SRAM_DQ = write_active ? data_q : 16'hzzzz;
assign SRAM_WE_N = ~write_active;
```

FIFO read and address increment:

Verilog code:

```
if (~fifo_empty) begin
    data_q       <= fifo_data;
    write_active <= 1'b1;
    SRAM_ADDR    <= SRAM_ADDR + 1;
End
```

This logic ensures:

- Data is read from FIFO
- Written to SRAM
- Address automatically increments

Thus, samples are stored in correct chronological order.

4.6 Top-Level Module Integration

All functional blocks ADC interface, FIFO buffer, SRAM controller, and PLL are integrated into a single top-level Verilog module in Quartus. This module manages clock distribution, reset behavior, GPIO interfacing, and data flow between modules.

The modular design approach allows each block to be verified independently before full system integration, simplifying debugging and improving design clarity.

The top module connects:

- PLL clock
- ADC interface
- FIFO buffer
- SRAM writer

It also provides debug signals through LEDs:

Verilog code:

```
assign LEDG[1] = ~fifo_empty;

assign LEDG[2] = fifo_full;

assign LEDG[4] = ~SRAM_WE_N;
```

4.7 MATLAB Verification of Verilog Operation

The complete FPGA design was written in Verilog HDL[7]. The most important modules developed for this system are:

- ADC data capture logic
- FIFO buffer interface
- SRAM writer FSM
- Top-level integration module

Small portions of the Verilog code are shown here for clarity[15], while the full source code is provided in the Appendix.

Example of ADC data registration:

Verilog code:

```
always @(posedge clk_60MHz or negedge reset_n) begin
    if (!reset_n)
        adc_data_ff <= 10'd0;
    else
        adc_data_ff <= GPIO_IN;
end
```

Example of SRAM write control:

Verilog code:

```
assign SRAM_DQ = write_active ? data_q : 16'hzzzz;
assign SRAM_WE_N = ~write_active;
assign fifo_rd_en = ~fifo_empty;
```

These code fragments illustrate how ADC data is captured and written into memory sequentially.

4.8 MATLAB-Based Verification of FPGA Operation

After acquisition, the SRAM data is exported as a HEX file and processed using MATLAB scripts developed specifically for this project. The MATLAB program extracts the stored samples, converts them into signed values, removes DC offset, and reconstructs the time-domain waveform. The Fast Fourier Transform (FFT) is then computed to analyze the frequency content of the signal.

The accuracy of the MATLAB plots directly confirms the correctness of the FPGA design. The clean sine waveform observed in the time domain and the correct frequency peak in the FFT validate that the FIFO buffering and SRAM writing logic operate as intended.

MATLAB code:

```
val16 = typecast(uint8(hex2dec((hexStr))), 'int8');
adc = adc - mean(adc);
X = fft(adc, N);
```

This script reconstructs the waveform and computes the FFT. The correct sine wave and frequency peak observed in MATLAB confirm the proper operation of the Verilog design[15].

4.9 Summary

This chapter demonstrated how the data acquisition system was implemented in Verilog on the FPGA. The use of synchronized clocking, FIFO buffering, and an FSM-based SRAM controller ensures deterministic and lossless data capture. The successful MATLAB analysis validates the correctness of the entire FPGA design.[16], [17]

5. Chapter 5

5.1 Introduction

This chapter presents the experimental validation of the FPGA-based data acquisition system using time-domain and frequency-domain analysis of the captured ADC samples. After acquisition, the data stored in SRAM is exported and processed using MATLAB to reconstruct the waveform and evaluate its spectral content.

The methodology adopted in this work follows the approach discussed in recent studies on ADC characterization using time-domain sampled data, where frequency estimation and FFT analysis are used to validate converter performance and signal integrity.[18]

5.2 Test Conditions

The ADC is driven by a 60 MHz sampling clock generated by the FPGA. A signal generator provides sinusoidal inputs with frequencies between 100 kHz and 2 MHz and an amplitude of approximately -12 dBm.

Since the Nyquist frequency is 30 MHz, the tested signals are highly oversampled, allowing accurate reconstruction of the waveform and precise frequency estimation during analysis.

5.3 Data Extraction and MATLAB Processing

The data stored in SRAM is exported as a HEX file and processed using MATLAB. The script converts the stored samples into signed numerical values, removes DC offset, constructs a time axis, and computes the FFT.

This time-domain reconstruction combined with frequency estimation closely follows the ADC characterization approach described in [18], where sampled data is used to verify the behavior of the converter without relying solely on analog instrumentation.[16]

A MATLAB script was developed to process the extracted HEX file and reconstruct the waveform. The script converts the stored samples into signed numerical values, removes DC offset, constructs a time axis based on the sampling frequency, and computes the Fast Fourier Transform (FFT).

This type of digital post-processing of ADC data is widely used in modern converter characterization, where the performance of the ADC is evaluated using time-domain samples and spectral analysis rather than relying only on analog instrumentation, as discussed in [16].

5.4 Data Retrieval Using Altera Control Panel

After the acquisition phase, the contents of the external SRAM[12] are retrieved using the Altera DE2-115 Control Panel utility. This software tool allows direct access to the SRAM memory mapped on the FPGA board and enables downloading the stored data as a hexadecimal (HEX) file.[2]

Using the Control Panel ensures that the data extracted from memory corresponds exactly to the sequential samples written by the FPGA during acquisition. This method avoids any manual data handling and provides a reliable way to transfer the captured samples from the hardware platform to the MATLAB environment for further analysis.[18]

The use of the Altera Control Panel for memory readback is a practical and efficient approach commonly adopted in FPGA-based data acquisition systems for offline signal evaluation.

5.5 Time-Domain Analysis of the ADC Output

The reconstructed waveform in the time domain shows the sequence of sampled ADC values. At moderate input amplitudes, the waveform resembles a sine wave. However, when the input amplitude is reduced, the waveform appears staircase-shaped or square-like.

As explained in [18], this behavior is a direct consequence of quantization in the ADC. When only a limited number of quantization levels are utilized, the waveform appears discrete rather than smooth. This effect does not indicate malfunction but confirms the expected behavior of digital sampling at low signal amplitudes.

Thus, the time-domain plot provides important insight into the effective resolution of the ADC under different input conditions.

5.6 Frequency-Domain Analysis Using FFT

The FFT of the reconstructed signal shows a clear peak at the input signal frequency, confirming correct sampling and data storage. Minor harmonic components may also be observed, which can be attributed to front-end nonlinearity and quantization noise.

The paper in [18]highlights that frequency estimation from sampled data is a powerful method to validate ADC operation and linearity. In this work, the FFT results confirm that the FPGA correctly captures and preserves the sampled data without distortion or loss.

Similar time-domain analysis of sampled ADC data is used in [16] to study the behavior of digitally implemented ADC architectures, where discrete sample representation provides insight into quantization and resolution effects.

5.7 Correlation Between Time and Frequency Domain

One of the key observations is the direct relationship between the staircase appearance in the time domain and the spectral purity observed in the frequency domain. As described in [18], even when the time-domain waveform appears non-smooth due to quantization, the frequency spectrum still accurately reveals the fundamental frequency component.

This confirms that the FPGA-based acquisition system preserves the essential characteristics of the input signal despite the discrete nature of digital sampling.

FFT-based evaluation of sampled ADC data is a standard approach for verifying frequency integrity and spectral behavior, as demonstrated in [16], where digital processing is used to validate ADC operation.

5.8 Validation of the Complete Acquisition Chain

The agreement between:

- Expected input frequency
- Time-domain waveform
- FFT spectral peak

demonstrates that the entire acquisition chain from ADC sampling, FIFO buffering, SRAM storage, to MATLAB reconstruction operates correctly. This validation approach is consistent with modern ADC characterization techniques described in [18].

5.9 Summary

This chapter demonstrated how time-domain and frequency-domain analysis of the stored ADC samples can be used to validate the performance of the FPGA-based acquisition system. The observed results confirm correct system operation and illustrate fundamental ADC behaviors such as quantization and spectral representation.

6. Chapter 6

6.1 Introduction

This chapter discusses the experimental results obtained from the implemented FPGA-based data acquisition system. The goal is not only to show that the system works, but also to explain why the time-domain waveforms and FFT spectra look the way they do. The discussion connects the measured output to the main factors that influence the acquisition chain: sampling clock, ADC resolution, input amplitude, analog front-end behavior, and the FPGA memory-writing strategy.

The analysis approach follows common ADC evaluation practice where stored time-domain samples are used to validate converter behavior and frequency-domain properties through FFT-based methods and frequency estimation techniques.

6.2 Measurement Setup and Data Flow Validation

The ADC is clocked using a 60 MHz sampling clock generated by the FPGA PLL and routed to the ADC clock input. A laboratory signal generator provides sinusoidal inputs with frequency swept from 100 kHz to 2 MHz at approximately −12 dBm amplitude. The captured data is written sequentially into external SRAM through the FIFO + FSM architecture described in Chapter 4.

After acquisition, the SRAM contents are exported to a HEX file using the Altera DE2-115 Control Panel utility. MATLAB scripts then decode the HEX stream, reconstruct the time-domain waveform, remove DC offset, and compute the FFT. This “FPGA capture → memory (16bit memory) → offline MATLAB analysis” workflow is widely used in mixed-signal measurement systems because it provides repeatable validation while keeping FPGA logic simple and deterministic [7].

A key point validated during this stage is sample continuity: the extracted data shows sequential samples with no missing blocks or repeated patterns, confirming that FIFO buffering and the SRAM write logic prevent sample loss and preserve chronological order.

6.3 Experimental Measurement Setup

The experimental setup consists of a laboratory signal generator, the ADC front-end circuit, the Terasic DE2-115 FPGA board, and a host PC for data analysis. A sinusoidal input signal is generated with frequencies ranging from 100 kHz to 2 MHz and an amplitude of approximately −12 dBm.

The ADC is clocked using a 60 MHz sampling clock generated internally by the FPGA through an on-chip PLL. The digitized samples are captured by the FPGA, buffered using a FIFO, and written sequentially into external SRAM. After the acquisition phase, the stored data is extracted as a HEX file using the Altera DE2-115 Control Panel tool and processed offline in MATLAB.

This measurement flow ensures deterministic sampling, reliable data storage, and repeatable analysis.

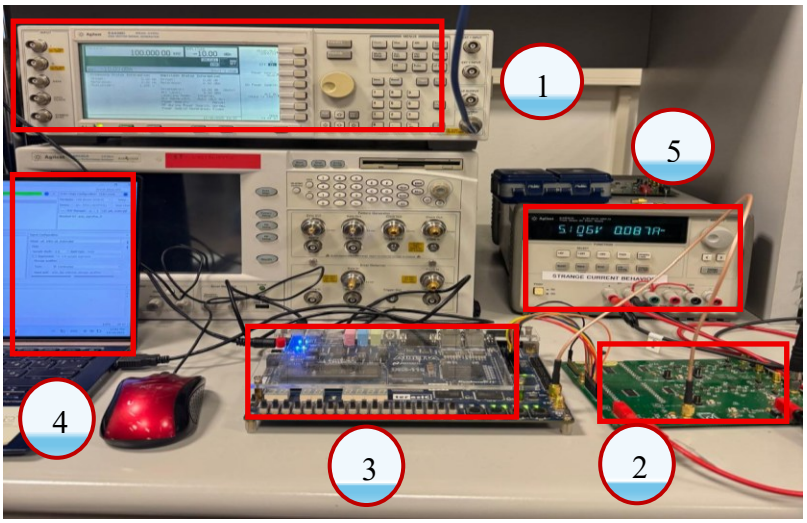


Figure 8 (1. Signal Generator 2. ADC Board 3.FPGA Board 4.PC 5.Power Supply)

6.4 Time-Domain Results

Figure 9 shows the reconstructed time-domain waveform for a 100 kHz sinusoidal input signal. The waveform exhibits a clear periodic behavior consistent with the applied input signal, confirming correct ADC operation and proper synchronization between the ADC and FPGA.

Before digital post-processing, the waveform shows small amplitude fluctuations and occasional spikes. These variations are mainly caused by quantization noise, thermal noise from the analog front-end, and wideband noise captured due to the high sampling frequency.

Such behavior is expected in practical high-speed ADC systems and does not indicate malfunction.

To improve waveform readability, a digital low-pass filter with a cutoff frequency of 2MHz was applied during MATLAB post-processing. After filtering, the time-domain waveform becomes significantly smoother, clearly revealing the sinusoidal shape of the signal while preserving its amplitude and phase. This confirms that the observed noise originates from wideband components rather than from errors in the FPGA acquisition logic.

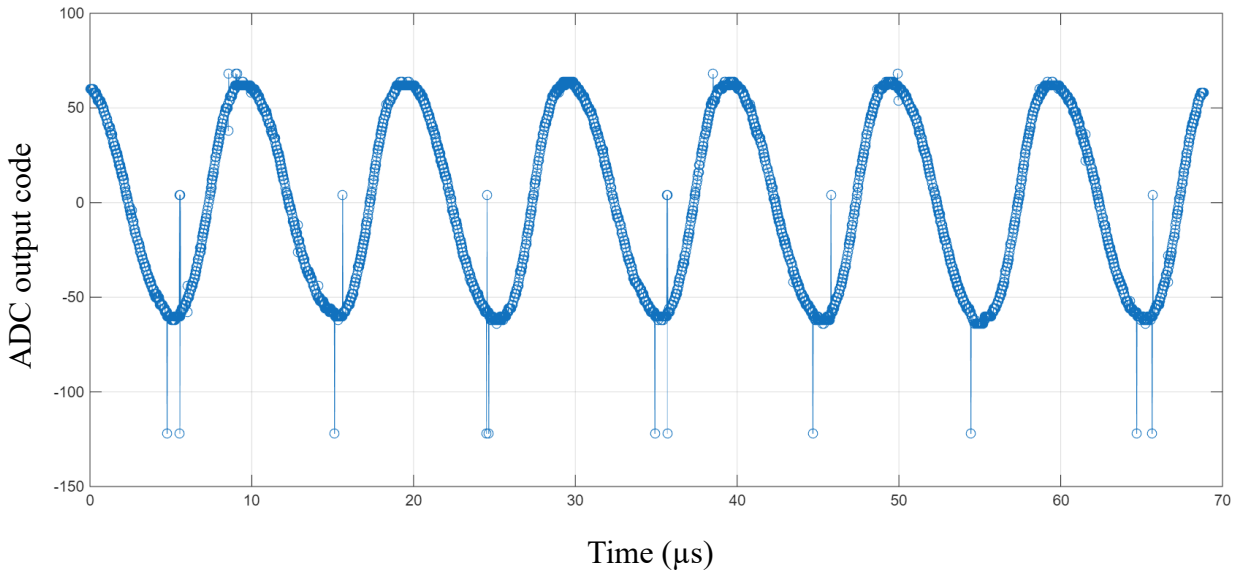


Figure 9 Time Domain 100 KHz

6.5 Frequency-Domain Results

The frequency-domain behavior of the acquired signal is analyzed using the Fast Fourier Transform (FFT). Figure 10 presents the FFT magnitude spectrum corresponding to the same 100 kHz input signal.

A dominant spectral peak is clearly visible at 0.1 MHz, matching the input signal frequency. This result confirms that the sampling frequency, FIFO buffering, SRAM storage, and MATLAB reconstruction preserve the frequency content of the signal accurately.

The remaining spectral components are significantly lower in magnitude and represent quantization noise, analog front-end noise, and minor harmonic distortion. No aliasing effects are observed, as the input signal frequency is well below the Nyquist frequency of 30 MHz.

After applying the digital low-pass filter, the noise floor in the FFT is noticeably reduced, improving spectral clarity while leaving the fundamental component unaffected. This further validates the effectiveness of post-processing without altering the raw acquisition behavior.

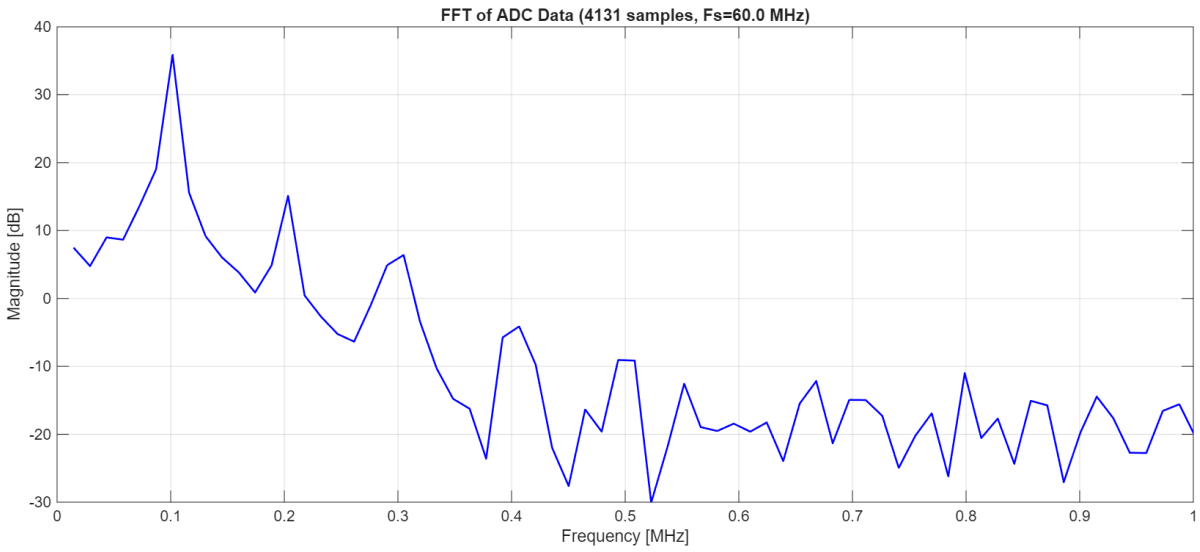


Figure 10 Frequency Domain 100kHz

In this Frequency-domain waveform (Figure 10) of the 10-bit AD9215 ADC output acquired on the DE2-115 FPGA board for a 100 kHz input sine wave, sampled at 60 MS/s with an input power of −12 dBm, using 4131 samples.

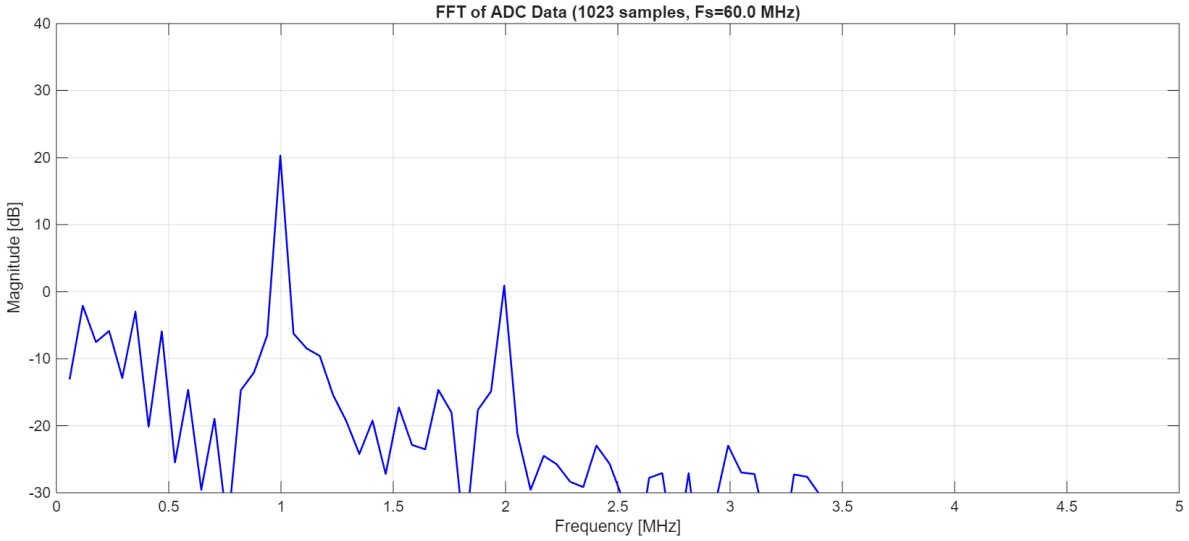


Figure 11 Frequency Domain 1MHz

This figure is for Frequency-domain waveform (Figure 11) of the 10-bit AD9215 ADC output acquired on the DE2-115 FPGA board for a 1 MHz input sine wave, sampled at 60 MS/s with an input power of −18 dBm, using 1023 samples.

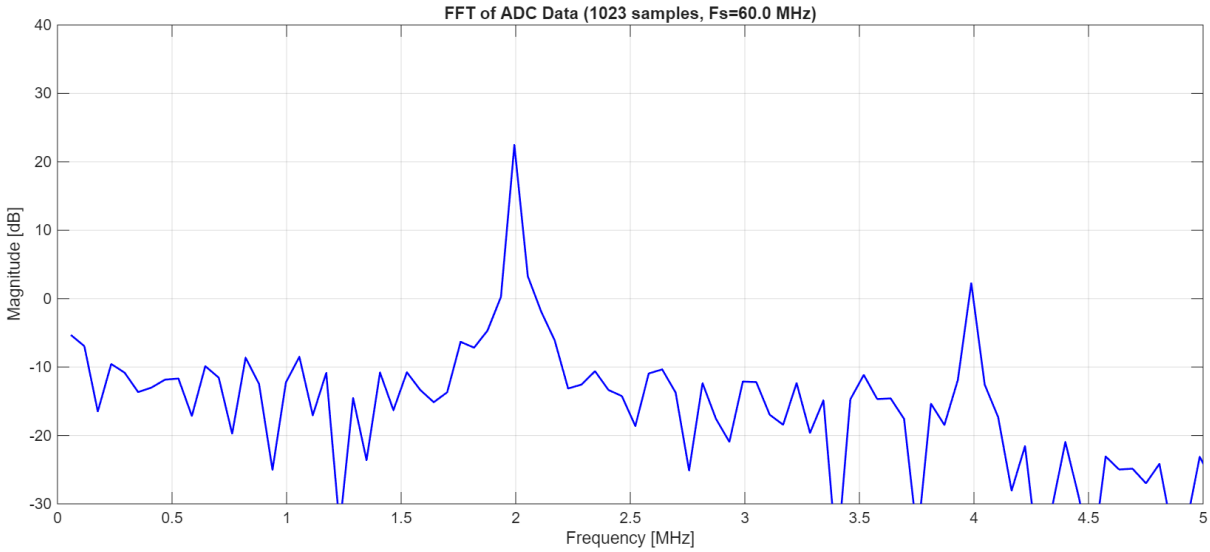


Figure 12 Frequency Domain 2MHz

This figure is for Frequency-domain waveform (Figure 12) of the 10-bit AD9215 ADC output acquired on the DE2-115 FPGA board for a 2 MHz input sine wave, sampled at 60 MS/s with an input power of −18 dBm, using 1023 samples.

6.6 Discussion of Results

The combined time-domain and frequency-domain results demonstrate the correct operation of the proposed data acquisition system. The FPGA reliably captures high-speed ADC data, stores it sequentially in external SRAM, and enables accurate reconstruction of the signal in MATLAB.

The staircase-like behavior observed in unfiltered time-domain plots at low amplitudes is a direct consequence of finite ADC resolution and quantization effects. Similarly, the presence of a noise floor and small harmonic components in the FFT is consistent with expected non-idealities in real ADC systems, including analog front-end limitations and clock jitter.

The application of digital filtering in MATLAB highlights how post-processing techniques can improve signal visualization and analysis without masking hardware performance. This approach is particularly useful for ADC characterization and validation tasks.

6.7 System Validation

The agreement between the known input signal parameters, the reconstructed time-domain waveform, and the FFT spectral peak confirms the correctness of the complete acquisition chain. The FIFO-based buffering prevents data loss during continuous sampling, and the FSM-based SRAM controller ensures correct sample ordering.

The successful use of the Altera Control Panel for data extraction further demonstrates the practicality and reliability of the system for offline signal analysis.

6.8 Effect of Digital Low-Pass Filtering on ADC Data

During the initial time-domain analysis of the acquired ADC samples, the reconstructed waveform exhibited small amplitude fluctuations and occasional spikes. These distortions are mainly due to wideband noise components captured by the ADC when operating at a high sampling frequency of 60 MHz Since the input signal frequency is significantly lower (100 kHz), the sampled data also includes high-frequency noise originating from the analog front-end, quantization effects, and clock jitter.

To improve signal clarity and better visualize the fundamental waveform, a digital low-pass filter was applied during MATLAB post-processing. The filter was implemented using MATLAB’s

built-in `lowpass ()` function with a cutoff frequency of 5 MHz, which is well above the input signal frequency but sufficiently low to attenuate unwanted high-frequency noise components.

Before filtering, the time-domain waveform showed visible noise superimposed on the sinusoidal signal, making it difficult to clearly observe the signal shape over multiple periods. Although the fundamental frequency was still present and correct, the noise reduced waveform readability.

After applying the low-pass filter, the time-domain signal became significantly smoother, and the sinusoidal shape was clearly restored. The filtering operation effectively removed high-frequency noise while preserving the amplitude and phase of the desired signal. This confirms that the observed noise was not caused by incorrect ADC operation or FPGA data capture, but rather by expected wideband noise inherent to high-speed sampling systems.

It is important to emphasize that this low-pass filtering was applied only during offline MATLAB analysis and does not alter the raw data captured by the FPGA. The use of digital filtering in post-processing is a standard technique in ADC characterization and allows for clearer interpretation of system performance without masking hardware-level behaviour.

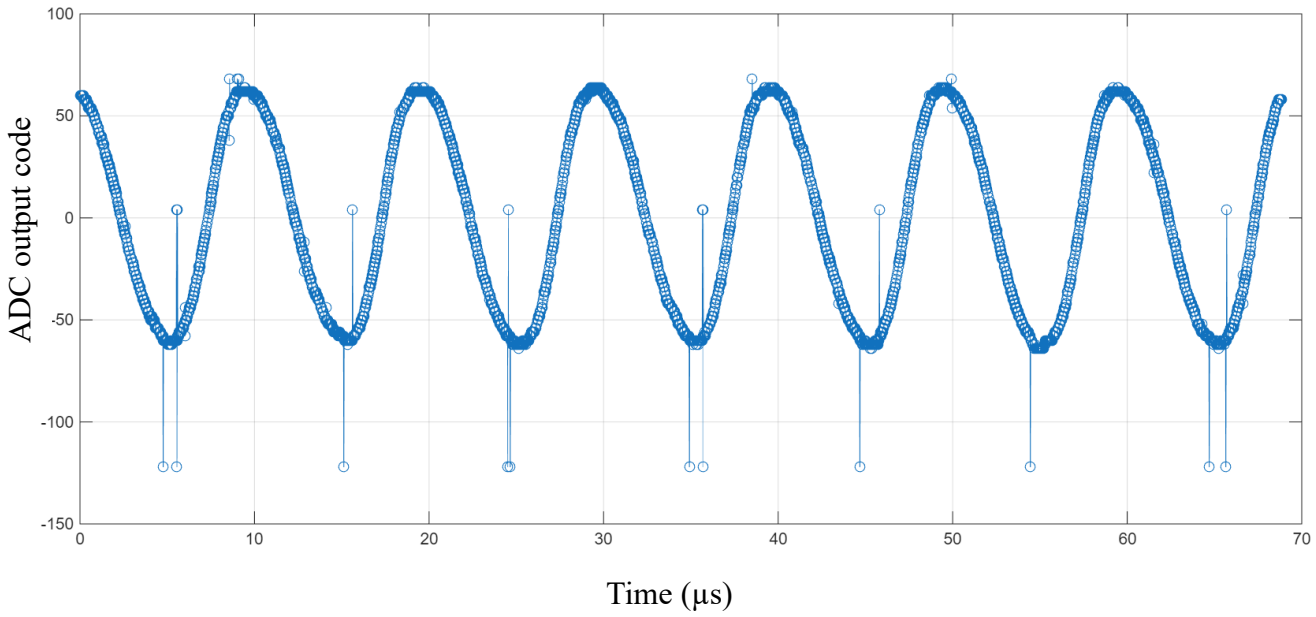


Figure 13 Time domain with Noise

MATLAB code:

```
%% Digital Low-Pass Filtering (Noise Reduction)
adc = lowpass (adc, 5e6, Fs); % 5 MHz cutoff
```

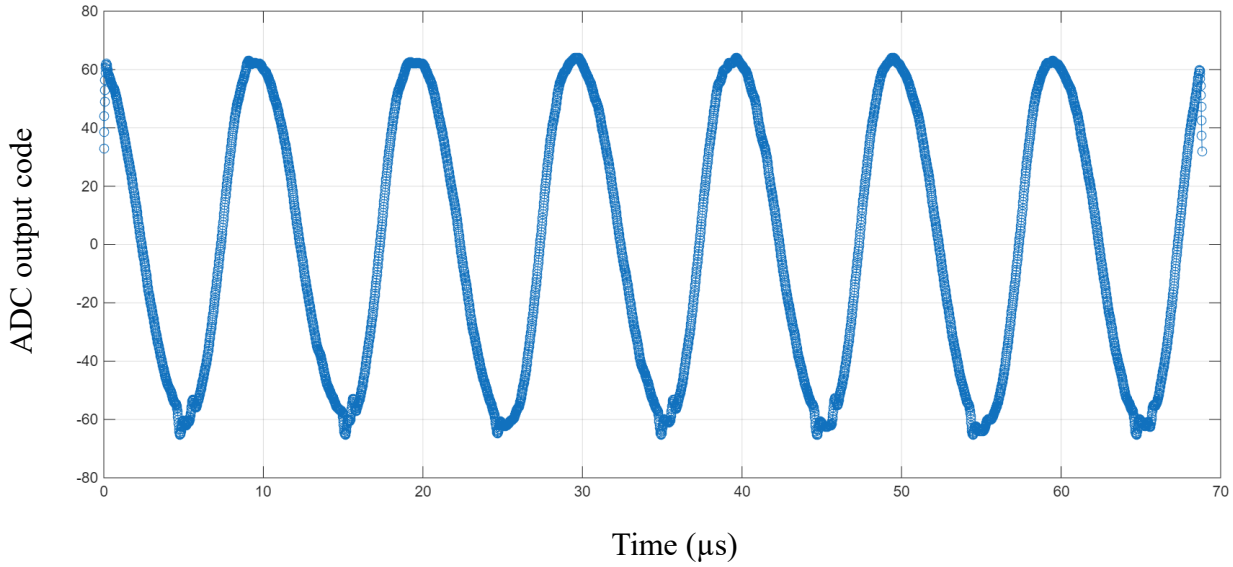


Figure 14 Time domain without noise

Here we can remove quantization Noise using ADC lowpass filter.

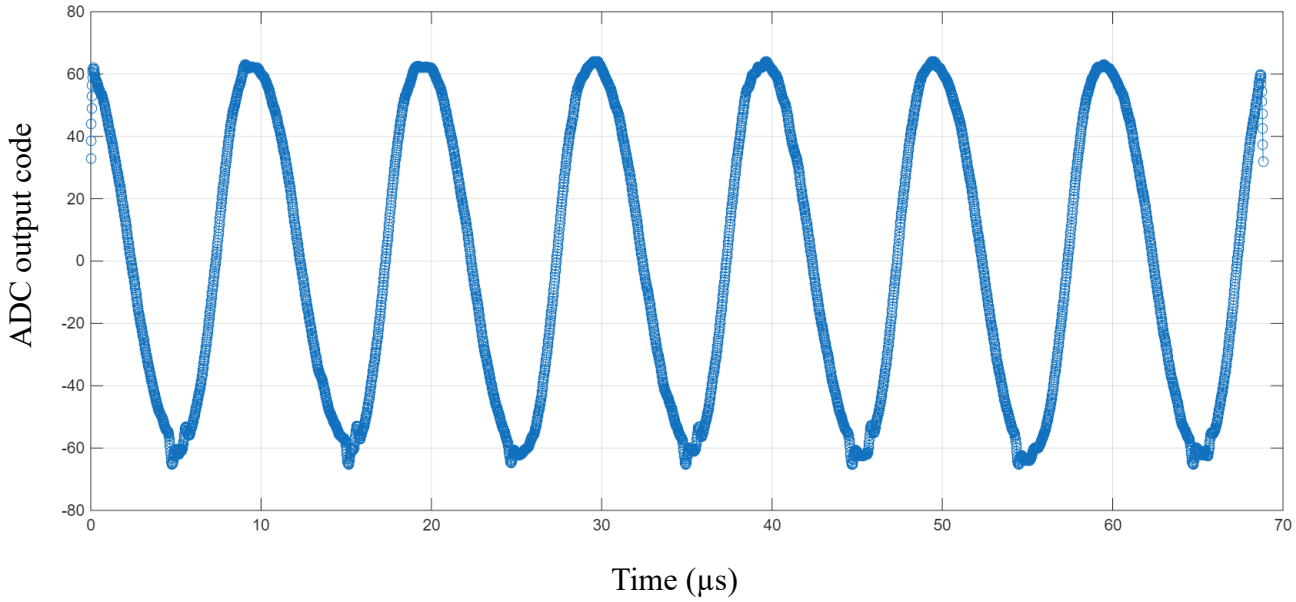


Figure 15 Time domain without noise 100kHz

In this Time-domain waveform (Figure 15) of the 10-bit AD9215 ADC output acquired on the DE2-115 FPGA board for a 100 kHz input sine wave, sampled at 60 MS/s with an input power of −12 dBm, using 1023 samples.

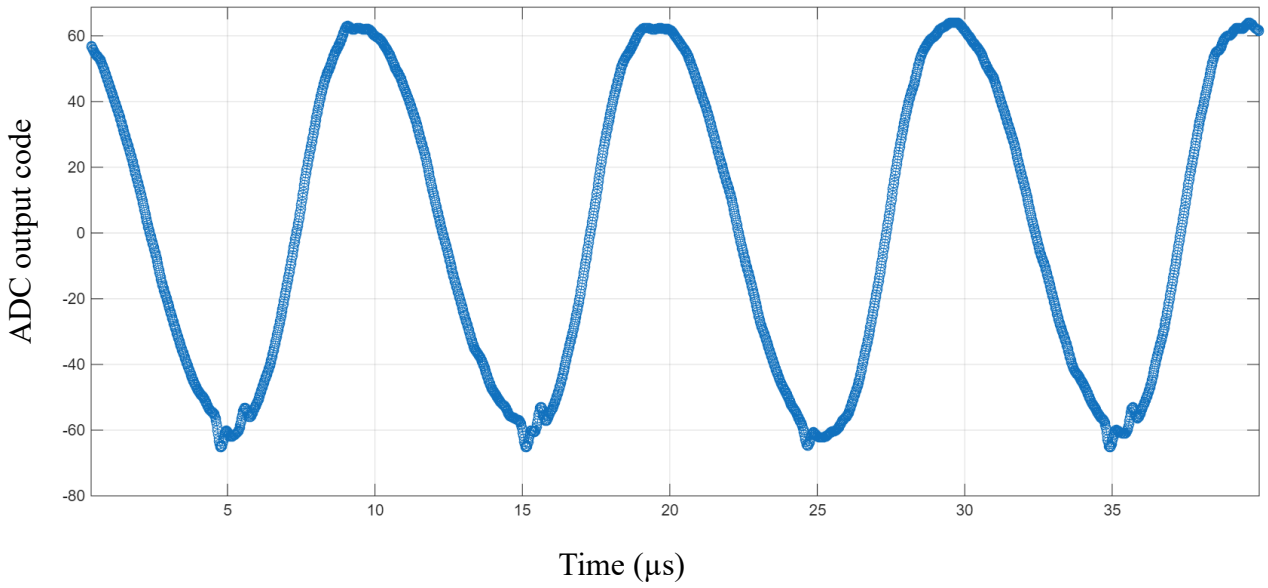


Figure 16 Time domain in zoom

This figure is for Time domain (Figure 16) 10-bit ADC output ,100KHz signal frequency,60MHz Clock frequency and Amplitude is -12dbm.For 500 Samples (Zoom mode).

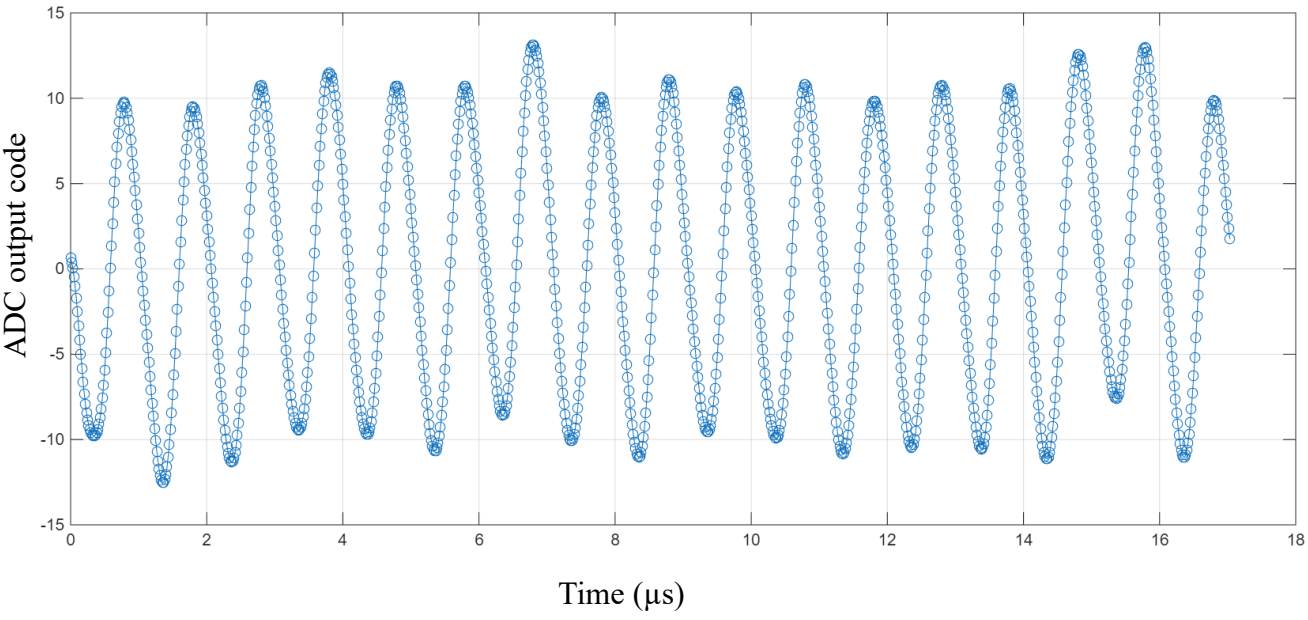


Figure 17 Time domain 1MHz

This figure is for Time domain (Figure 17) 10-bit ADC output ,1MHz signal frequency,60MHz Clock frequency and Amplitude is -12dbm.For 1023 Samples.

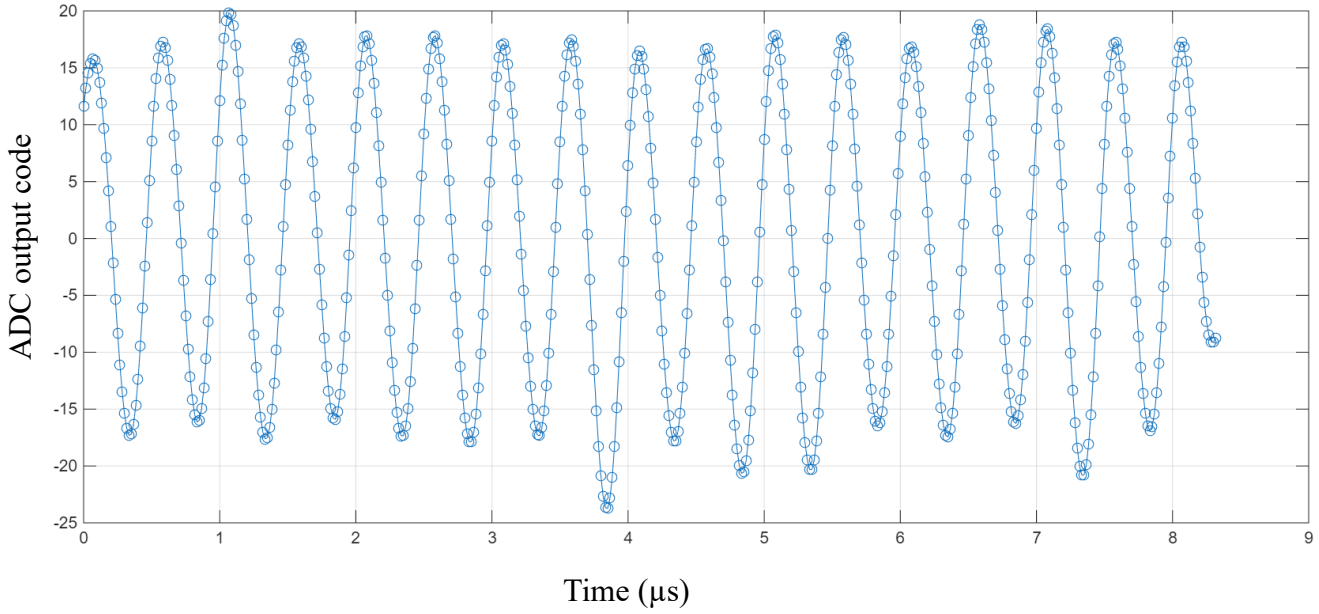


Figure 18 Time domain for 2MHz

This figure is for Time domain (Figure 18) 10-bit ADC output ,2MHz signal frequency,60MHz Clock frequency and Amplitude is -12dbm.For 1023 Samples

6.9 Limitations and Observations

While the system performs reliably, certain limitations are observed. The effective resolution depends on the input signal amplitude, and low-amplitude signals make quantization effects more visible. Additionally, analog front-end bandwidth and linearity influence harmonic distortion levels.

These limitations are inherent to practical ADC-based systems and do not detract from the validity of the proposed architecture. Instead, they highlight potential areas for future improvement.

6.10 Summary

This chapter presented and discussed the experimental results obtained from the FPGA-based data acquisition system. Time-domain analysis confirmed accurate waveform reconstruction, while frequency-domain analysis verified correct spectral behavior and frequency identification. The results demonstrate that the proposed system provides a reliable and flexible platform for ADC evaluation and signal analysis, successfully validating the design choices made throughout this thesis.

7. Chapter 7

7.1 Conclusion

This thesis presented the analysis, design, and experimental validation of an FPGA-based data acquisition system using an external high-speed ADC and on-board SRAM memory. The primary objective was to develop a reliable and flexible platform capable of accurately capturing analog signals, storing the digitized samples, and enabling detailed time-domain and frequency-domain analysis using MATLAB.

A complete acquisition chain was successfully implemented, including ADC clock generation using an FPGA PLL, synchronized ADC data capture, FIFO-based buffering, and deterministic SRAM writing through a finite-state machine. The use of FIFO buffering ensured lossless data acquisition by decoupling the high-speed sampling process from memory access timing.

Experimental results confirmed the correct operation of the system. Time-domain analysis demonstrated accurate waveform reconstruction for sinusoidal input signals, while frequency-domain analysis using FFT verified correct frequency identification and spectral behavior. The observed quantization effects and noise characteristics were consistent with theoretical expectations for finite-resolution ADCs operating at high sampling rates. The application of digital post-processing techniques, such as low-pass filtering and Hann windowing in MATLAB, further improved signal visualization and reduced spectral leakage without altering the raw acquired data.

Overall, the results validate the proposed FPGA-based architecture and demonstrate that the system provides a robust and effective platform for ADC characterization, signal acquisition, and offline analysis.

7.2 Contributions of the Work

The main contributions of this thesis can be summarized as follows:

- Design and implementation of an FPGA-based data acquisition system using an external ADC and SRAM memory
- Development of a reliable FIFO-based buffering strategy to prevent data loss during continuous sampling
- Implementation of a deterministic SRAM controller using Verilog HDL

- Successful extraction and reconstruction of ADC data using MATLAB
- Experimental validation through time-domain and frequency-domain analysis
- Clear demonstration of quantization and noise effects in practical ADC systems

These contributions highlight the practical value of the proposed system for educational, laboratory, and research-oriented applications.

7.3 Limitations

While the system operates reliably, several limitations were identified. The effective resolution of the acquired signal is influenced by the ADC resolution and the input signal amplitude, making quantization effects more visible at low amplitudes. Additionally, the performance of the system is affected by analog front-end noise, bandwidth limitations, and clock jitter. No real-time digital calibration or error-correction techniques were implemented in this work.

These limitations are inherent to practical mixed-signal systems and do not detract from the correctness of the proposed architecture.

7.4 Future Work

Several improvements and extensions can be considered in future developments of this system:

- Implementation of higher-resolution ADCs to improve signal fidelity
- Integration of real-time digital filtering or averaging directly on the FPGA
- Addition of digital calibration or error-correction techniques
- Extension of the system to support higher input bandwidths
- Implementation of real-time data streaming to a host PC
- Support for multi-channel ADC acquisition and synchronization

These enhancements would further increase the flexibility and performance of the acquisition platform and enable more advanced signal processing applications.

7.5 Final Remarks

The work presented in this thesis demonstrates that FPGA-based data acquisition systems offer a powerful and flexible solution for modern signal acquisition and analysis. By combining high-speed ADC sampling with deterministic digital control and software-based post-processing, the proposed system successfully bridges the gap between the analog and digital domains. The results achieved in this work provide a solid foundation for further research and development in FPGA-based mixed-signal systems.

[1] J. Jin, S. Lecchi, R. Castello, and D. Manstretta, “An FDD Auxiliary Receiver with a Highly Linear Low Noise Amplifier,” in *ESSCIRC 2022- IEEE 48th European Solid State Circuits Conference (ESSCIRC)*, IEEE, Sep. 2022, pp. 309–312. doi: 10.1109/ESSCIRC55480.2022.9911524.

[2] ALTERA CORP, *DE2-115 - User Manual*, vol. 53, no. 9. 2013.

[3] Missing Link Electronics Inc., “Integrated ADC for Altera Cyclone-IV Devices,” Technical Brief 20110419. Accessed: Nov. 12, 2025. [Online]. Available: file:///C:/Users/Vivek/Desktop/thesis/MLE-TB20110419.pdf

[4] Analog Devices, “10-Bit, 65/80/105 MSPS, 3 V A/D Converter,” 2013. [Online]. Available: www.analog.com

[5] S. Choi, H. Yang, Y. Noh, G. Kim, E. Kwon, and H. Yoo, “FPGA-Based Multi-Channel Real-Time Data Acquisition System,” *Electronics (Switzerland)*, vol. 13, no. 15, Aug. 2024, doi: 10.3390/electronics13152950.

[6] C. Annual and I. Report, “White paper Cisco public,” 2018.

[7] S. Gupta and S. Kumar, “Data capture via High speed ADCs using FPGA,” in *Proceedings - 2nd International Conference on Micro-Electronics and Telecommunication Engineering, ICMETE 2018*, 2018. doi: 10.1109/ICMETE.2018.00033.

[8] Y. Han *et al.*, “An FPGA platform for next-generation grating encoders,” *Sensors (Switzerland)*, vol. 20, no. 8, 2020, doi: 10.3390/s20082266.

[9] Texas Intruments, “LMH6550 Differential, High-Speed Operational Amplifier,” Jan. 2015.

[10] “3.3-V ABT 16-Bit Edge-Triggered D-Type Flip-Flops With 3-State Outputs datasheet (Rev. R)”.

[11] J. Song, Q. An, and S. Liu, “A high-resolution time-to-digital converter implemented in field-programmable-gate-arrays,” *IEEE Trans. Nucl. Sci.*, vol. 53, no. 1, pp. 236–241, Feb. 2006, doi: 10.1109/TNS.2006.869820.

[12] “IS61WV102416ALL IS61WV102416BLL IS64WV102416BLL 1M x 16 HIGH-SPEED ASYNCHRONOUS CMOS STATIC RAM WITH 3.3V SUPPLY,” 2014. [Online]. Available: www.issi.com

[13] G. Castellano, D. De Caro, A. G. M. Strollo, and D. Manstretta, “A Low Power Control System for Real-Time Tuning of a Hybrid Transformer-based Receiver.”

[14] “AD9215 Analog Devices.” [Online]. Available: www.analog.com

[15] S. Brown and Z. Vranesic, “FUNDAMENTALS OF DIGITAL LOGIC WITH VHDL DESIGN FOURTH EDITION.”

[16] *2015 Nordic Circuits and Systems Conference*. IEEE, 2015.

[17] Shaofeng. Yu, Xiaona. Zhu, and T.-Ao. Tang, *ICSICT-2020 : 2020 IEEE 15th International Conference on Solid-State and Integrated Circuit Technology (ICSICT) proceedings : Nov. 3-Nov. 6, 2020, Kunming, China*. IEEE Press, 2020.

[18] D. Dallet, D. Slepička, Y. Berthoumieu, D. Haddadi, and P. Marchegay, "[ADC characterization in time domain] Frequency estimation to linearize time-domain analysis of A/D converters," *IEEE Trans. Instrum. Meas.*, vol. 55, no. 5, pp. 1536–1545, Oct. 2006, doi: 10.1109/TIM.2006.880274.