POLITECNICO DI TORINO

POLITECNICO DI TORINO

Master Degree course in Communications Engineering

Master Degree Thesis

Integrating Sensing and Communication Capabilities in a Phase Modulated Continuous Wave (PMCW) Radar System

Supervisors
Prof. Riccardo Maggiora

Candidate
Hussein Ali Abdulhadi Al-Hakeem

Academic Year 2023-2024

Acknowledgements

First and foremost, I would like to express my deepest gratitude to **God** for His guidance and strength throughout this journey.

I extend my heartfelt thanks to my thesis supervisor, Prof. Riccardo Maggiora, for his unwavering support and valuable insights, which have been instrumental in completing this work. I am also deeply appreciative of the committee members who will be evaluating my defense for their time and effort. My sincere thanks to Politecnico di Torino for providing me with the opportunity and resources to pursue this thesis.
A special acknowledgment goes to Dr. Maurice, whose mentorship, knowledge, and encouragement have greatly enriched my learning experience. I cannot thank him enough for the countless ways he has guided me during this thesis. I also wish to express my gratitude to the rest of the Radar Lab team — Sara, Francesco, and Simone — who made the long months of work a truly collaborative and enjoyable experience. Your company made all the difference. To every professor I have had the privilege of learning from during my time at Politecnico, I offer my sincerest thanks. Your teachings have shaped the foundation of this work.

To my family, words cannot capture my gratitude. To my father, thank you for your strength, your unwavering support, and your belief in me. To my beloved mother, though you are no longer with us, I dedicate this achievement to your memory and your boundless love. You are the light that has always guided me, and this milestone is for you. To my Uncle Eng. Saleh, thank you for being not only an incredible uncle but also a second father to me. Your guidance and unwavering support have been a cornerstone in my life. To Mr. Laith, my trusted 'friend' and number one emergency contact, and to all my other relatives, thank you for being part of my journey and for the love and encouragement you've always shown me.
To my friends back in Lebanon, thank you for your lifelong friendship and constant support from afar. You have been my foundation, even across borders. I would also like to express my deepest thanks to my dear friends Giusi and Rosella, whose warmth and companionship made my time in Italy not only productive but truly joyful. These past two years would not have been the same without you.

This thesis is a product of the knowledge, guidance, love, and support I have received from all of you, and I am eternally grateful.

Abstract

This thesis focuses on the innovative design and implementation of communication capabilities integrated into an existing Phase-Modulated Continuous-Wave (PMCW) radar system for automotive applications. The project addresses the critical need for radar-communication co-existence, offering a solution that enables simultaneous high-resolution radar sensing and real-time data transmission with the same hardware. The key contribution of this work is the development of a system prototype that allows radar and communication functionalities to operate concurrently, without compromising radar performance. Among the various innovations introduced, the most challenging are proper waveform design, advanced software algorithms for Carrier Frequency Offset (CFO) compensation and phase recovery, and real-time radar and communication processes management.

The system operates in the 81-86 GHz band, leveraging a Xilinx FPGA platform and custom-designed RF hardware for real-time processing of both radar and communication signals. Extensive laboratory testing validated the system's performance, demonstrating reliable payload demodulation, wide-open eye diagrams, and robust co-existence of radar and communication capabilities. This project lays a crucial foundation for future advancements in Vehicle-to-Infrastructure (V2I) and Vehicle-to-Vehicle (V2V) communication and Advanced Driver Assistance Systems (ADAS), offering significant improvements in automotive safety and traffic management.

Contents

List of Acronyms

List of Units of Measurement

Executive Summary

This thesis presents the successful design, implementation, and validation of a radar-communication integrated system, advancing the capabilities of the Phase-Modulated Continuous-Wave (PMCW) radar prototype by integrating robust communication functionalities. The work delivered a comprehensive hardware, firmware, and software solution designed for advanced automotive systems, capable of simultaneous radar sensing and real-time data communication. This is a critical leap toward enabling Advanced Driver Assistance Systems (ADAS) and Vehicle-to-Infrastructure (V2I) and Vehicle-to-Vehicle (V2V) communication. This solution is accompanied by performance validation through real-time testing, which demonstrates the effectiveness and robustness of the system.

Theory and Background

PMCW radar, known for its superior interference rejection and high-resolution sensing, offers significant advantages over the Frequency Modulated Continuous-Wave (FMCW) radars, including no range-Doppler coupling, more flexible design choices since the maximum range does not depend on the bandwidth of the transmitted signal, ability to maximize contrast resolution due to the employed digital correlation technique of the proper sequences along range, and easier integration of communication capabilities.
The integration of the communication system aims to enhance the radar system's functionality by enabling it to perform both radar sensing and communication tasks simultaneously. This dual-mode operation is achieved through careful synchronization and coordination between the radar and communication functions
This dual-function system operates at the millimeter-wave band (81-86 GHz), chosen for its high bandwidth available. It employs Binary Phase Shift Keying (BPSK) modulation for communication with one transmitter (Tx) and one receiver (Rx) channel, ensuring compatibility with radar functionality. It operates in two receiving modes according to a Time Division scheme: Radar-only process mode and Radar-communication process mode.
The challenge of integrating communication into a radar system without degrading radar performance was met with innovative firmware upgrades and advanced signal processing techniques.

Hardware Overview

The hardware adopted for this project consists of a radar system designed for mm-Wave signal acquisition and processing. Its processing core is the Xilinx KC705 FPGA kit and the AD-FMCDAQ2-EBZ board, which controls data acquisition and transmission. Critical RF components, including the ADMV7320 upconverter and ADMV7420 downconverter chips, handle transmission and reception in the 81-86 GHz band, ensuring high signal integrity.
The system employs the JESD204B interface between the FPGA and DAQ board, enabling

precise, high-speed ADC data capture. Essential components like bias tees, DC blocks, and attenuators ensure optimal signal integrity throughout the system.
This robust hardware satisfies the need for the project's advanced firmware and software developments. This hardware configuration allows the simultaneous transmission and reception of radar signals and communication payloads, all transferred in real-time through a 1Gbps Ethernet interface to a high-performance laptop PC responsible for the high-level real-time processing.

Project Implementation

Key Firmware Innovations and Upgrades

The radar firmware was upgraded to support the integration of communication capabilities. Key modifications implemented to the blocks constituting the system include:

- **Sequence Generator**: A 1024-bit fixed communication payload was integrated into 64 16-bit blocks after each of the 64 radar pulses of a frame. This was achieved by modifying the VHDL code to read from a dedicated payload register, ensuring that the system could transmit the communication data embedded in the radar signal.
- **Trigger Block**: Enhanced to capture external signals through a threshold triggering mechanism prioritizing the capture of the ranging data.
- **MAC Module**: Updated to support the transmission of both ranging and communication data streams to two different UDP ports always prioritizing the ranging data stream.
- **Optimized Memory Utilization**: With the introduction of a parallel communication data stream, buffer management was optimized to handle large amounts of radar and communication data without overloading the FPGA. Optimizing the RAM-based shift register and the FIFO management criterion ensured the efficient handling of 14-bit precision ADC data streams.

High-Level Software Processing

The software architecture, written in C on the laptop PC, handles both real-time data acquisition from the FPGA and signal processing. A major feature is the real-time generation of eye diagrams for the communication payload and range-Doppler maps for the radar waveform, providing immediate visual feedback on system performance. Two applications were developed: **radar_84_ghz_R** for ranging data and **radar_84_ghz_C** for communication data running concurrently. The **radar_84_ghz_C** application processes the raw communication data and performs Carrier Frequency Offset (CFO) compensation, phase synchronization, and payload decoding. The CFO estimator uses an FFT-based non-data-aided approach to correct frequency offsets across the received signals. After compensating for CFO, a sophisticated correlation-based phase recovery algorithm aligns

the phase of received signals of a frame using the radar-transmitted Golay complementary sequences. The result is a precisely synchronized signal, ready for data demodulation.

Moreover, some implemented MATLAB scripts compare the received communication payloads against the transmitted data, confirming a zero error rate in payload recovery.

Results and Validation

The integration of the communication task into the PMCW radar prototype was validated with real-world testing, demonstrating the system's capability to implement both radar sensing and communication tasks simultaneously. The final setup, as shown in Figure 1, highlights two radar units facing each other and engaging in efficient communication. An example of primary results is shown in Figure 2. These results demonstrate the correct reception of data. The open eye diagram validates the system's successful demodulation processes. This confirms the robustness of the system's communication process. Simultaneously, the radar unit successfully detected targets as shown in the range-Doppler map.

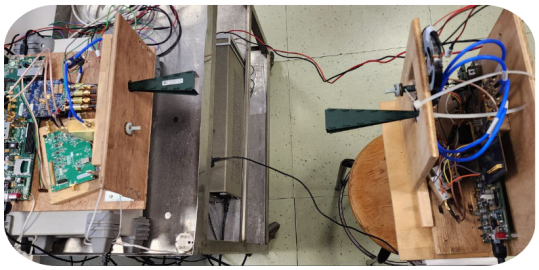


Figure 1: Experimental Setup

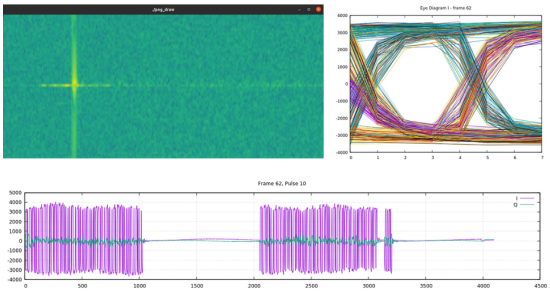


Figure 2: Final Validation Test Results

Conclusion

The key result of this project is the successful integration of communication functionality into an existing PMCW radar system without compromising its primary sensing task. This achievement represents a significant advancement in radar technology, particularly for automotive applications like autonomous driving and V2I and V2V communication. By combining radar sensing and communication, the system offers greater functionality and versatility, contributing to the development of more robust ADAS systems.

The work not only pushes the boundaries of radar and communication technology but also demonstrates its real-world applicability, meeting the stringent demands of future vehicles. This integration is a crucial step toward fully connected and autonomous vehicles, paving the way for improved road safety and traffic efficiency.

Chapter 1

Introduction

1.1 Fundementals of millimeter-Wave Radar Sensors

Renowned for their robustness in adverse weather conditions, millimeter (mm)-wave radar sensors are indispensable in Advanced Driver-Assistance Systems (ADAS). These sensors transmit electromagnetic waves, which are scattered by objects in their path. By capturing the scattered signals, the radar system can determine the objects' range, velocity, and angle. mm-wave radars operate at wavelengths in the millimeter range, which offer advantages such as small system component size and high resolution [6, 13]. Figure 1.1 shows, at a high level, the general process implemented by radars where the waveform generator is connected to the transmitter antenna which transmits the signal into the air. The signal is then scattered by objects and received by the receiving antenna. The received signals are mixed with the Local Oscillator (LO), the same one used to generate the transmitted signal, to shift down the frequency to a lower, more manageable Intermediate Frequency (IF). This downconverted signal is then sampled and processed. This general process may require specific modifications for particular applications.

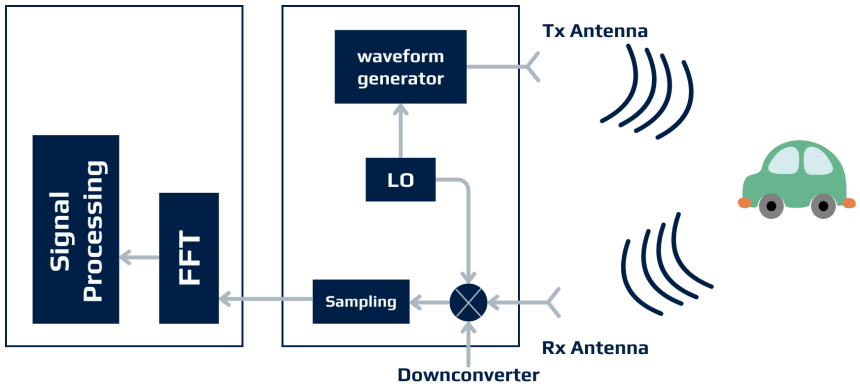


Figure 1.1: Typical Process Implemented by mm-Wave Radars

The first use of mm-wave radar technology in automobiles dates back to the early 1970s. [6]. Since then, several manufacturers have developed new generations of automotive

radar sensors. From a waveform modulation perspective, Linear Frequency-Modulated Continuous-Wave (FMCW) radars are favored due to simple implementation and low cost.

1.2 Adopted Radar Technology

Phase-Modulated Continuous-Wave (PMCW) Radars

Following the introduction of mm-wave radar sensors, the **PMCW radar** technology has emerged. This technology has been the subject of recent research for its application in automotive radar systems. PMCW radars, which implement all signal processing algorithms in digital domains, offer notable advantages with respect to FMCW radars. These advantages include higher resolution, improved High-Contrast Resolution (HCR), lower false alarms, and easier integration of communication capabilities. Remarkably, HCR denotes the radar's ability to differentiate between small targets situated near large targets. PMCW radars utilize pseudorandom modulation sequences, which have been heavily tested and verified in cellular communication systems to modulate the phase of a carrier frequency. Orthogonal digital sequences significantly reduce the interference from other radar systems operating in close proximity, ensuring reliable and accurate detection even in crowded and congested environments [13].
Pulse compression is a technique used to increase the signal bandwidth of a long pulse by modulating its amplitude, frequency, or phase. This method significantly enhances range resolution and target detection capabilities as explained later. Unlike FMCW radars, PMCW radars can digitally generate a wide variety of waveforms and perform pulse compression processing in the digital domain. This digital approach offers numerous benefits, including high performance, reproducibility, and flexibility.
In binary PMCW radars, a pulse of a specific duration T_s is divided into N sub-pulses of duration T_{ch} each, with the phase of a carrier frequency being either 0 or π according to a binary phase sequence as shown in Figure 1.2. The pulse compression filter (correlator) output is a compressed pulse with a peak width T_{ch} and a peak amplitude N times that of the uncompressed pulse, where the pulse compression ratio is $\frac{T_s}{T_{ch}}$. By correlating the received and transmitted signals, timing information on the targets can be accurately retrieved, which is used to calculate the targets' range information.

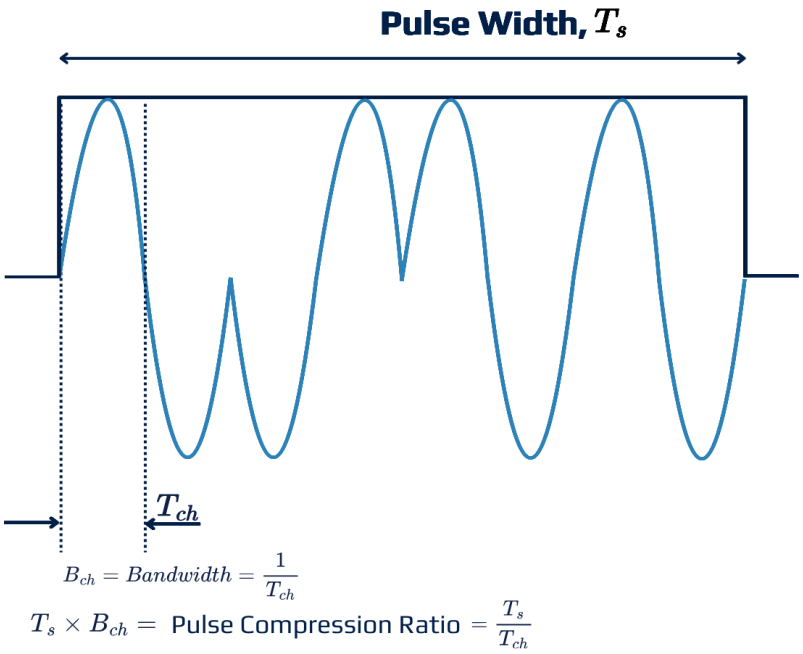


Figure 1.2: Binary Phase Coded Waveform for $N = 8$

The primary motivation for pulse compression is to achieve high-range resolution, which is crucial for all radar systems. High-range resolution is essential for target characterization, identification, and measurement accuracy. Using shorter pulses thus higher bandwidth provides better range resolution. Additionally, high peak power with large pulse energy is costly and challenging to obtain at high frequencies, making pulse compression the practical solution.

Advantages of PMCW

PMCW radar technology offers several significant advantages that enhance radar performance and reliability. One of the key benefits is superior sidelobe control, which is achieved through the use of proper sequences. These sequences enable efficient signal processing techniques that maximize Signal-to-Noise Ratio (SNR). PMCW radars employ a digital correlation technique along range, utilizing near-ideal auto-correlation functions that yield a distinctive 'thumbtack-like' range response, and significantly low-level sidelobes with the optimal sequence choice. By employing long spreading codes and capitalizing on the sharp auto-correlation function, PMCW radars excel in achieving precise range measurements, even in scenarios with significant variations in target scattering levels.
Additionally, PMCW radars demonstrate robust interference mitigation capabilities due to Code Division Multiple Access (CDMA). This allows multiple radar systems to operate in close proximity without causing significant interference to each other. The integration of communication and sensing functionalities is another notable advantage, enabling PMCW radars to support advanced applications that require both radar and communication capabilities.

The digital nature of PMCW radars makes it easy to modify the digital sequences, facilitating the implementation of advanced signal processing techniques. Proper sequence design can achieve extremely low sidelobes, further enhancing radar performance. Moreover, unlike FMCW radars, PMCW radars do not exhibit range-Doppler coupling, which simplifies signal processing and improves target detection accuracy. Note that range-Doppler coupling occurs when the measurements of a target's range and its velocity (Doppler shift) are interdependent, leading to uncertainties in determining either parameter.
Another significant advantage of PMCW radars compared to FMCW is that the maximum range does not depend on the bandwidth of the transmitted signal. This allows for greater flexibility in system design and optimization, making PMCW a versatile and powerful solution for automotive radar applications.

Future of PMCW in Automotive Radars

Due to its numerous advantages, PMCW technology is becoming the primary focus for automotive radars. The only challenge lies in the requirement for high-rate sampling and accurate Analog-to-Digital Converter (ADC)s. Unlike FMCW radars, which sample beat frequencies with relatively low bandwidth (5-20 Megahertz (MHz) typical for automotive applications), PMCW radars need to sample the full bandwidth of the transmitted signal, typically ranging from 1 to 2 Gigahertz (GHz).
However, recent advancements in chip integration have made it possible to adopt Gigasamples per second (Gsps) ADCs, addressing this challenge. Real-world performance measurements have demonstrated that PMCW radars exhibit superior performance compared to FMCW radars. This analysis indicates that PMCW radars have the potential to be successfully adopted in future automotive radar systems [1].

1.3 Radar and Communication Co-existence (RadCom)

Traditionally, both radar and communication systems have operated in separate frequency bands to avoid interference. However, the increasing demand for efficient spectrum utilization has brought significant attention to the concept of co-existence, where both radar and communication systems operate simultaneously within the same frequency band. This integrated approach aims to optimize the use of available spectrum resources while maintaining the performance of both radar and communication functions [14].
There are two primary approaches to achieving this integration. The first approach involves incorporating radar sensing capabilities into existing communication systems or components. This method leverages the infrastructure of communication systems to provide radar sensing functionalities, potentially reducing the need for additional hardware and improving overall system efficiency.
The second approach, which is **the focus of this work**, involves embedding communication functionalities directly into radar systems, a concept known as RadCom. This approach integrates communication capabilities within the radar hardware, allowing for seamless operation of both radar and communication functions.
RadCom systems leverage advanced signal processing techniques to enable the co-existence

of radar and communication functionalities. The key principles of RadCom systems include:

- **Waveform Design**: The design of waveforms that can simultaneously support radar and communication operations is crucial. These waveforms must be optimized to achieve a balance between radar detection performance and communication data rates. For instance, Orthogonal Frequency-Division Multiplexing (OFDM) waveforms can be used to achieve this balance by allocating different subcarriers for radar and communication purposes [14]. However, in this work, PMCW waveforms were used instead. PMCW waveforms are tailored to radar applications and provide several advantages over OFDM in this context. The key advantage and motivation for this choice is that PMCW are superior in their resilience to Doppler effects. This is attributed to their phase modulation scheme. In contrast, OFDM systems are significantly impacted by Doppler shifts. These shifts disrupt the orthogonality of subcarriers, leading to inter-carrier interference and consequent degradation in performance. Therefore, PMCW radars are a more suitable choice for applications requiring robust performance in dynamic environments.
- **Spectrum Sharing**: In case there are distinct radar and communication transmitters, efficient spectrum sharing mechanisms are essential to minimize interference between radar and communication signals. Techniques such as cognitive radio and dynamic spectrum access are employed to achieve this goal [14]. However, in this work, the radar and communication signals are inherently the same, meaning the signal inherently shares the same spectrum.
- **Signal Processing**: Advanced signal processing algorithms are used to separate and decode the received radar and communication signals. These algorithms must be robust to handle the challenges posed by the shared spectrum environment. For example, joint radar-communication signal processing techniques can be used to simultaneously extract radar and communication information from the received signals [14].

RadCom systems offer several distinct advantages:

- **Enhanced Spectrum Efficiency**: By sharing the same frequency band for both radar and communication, RadCom systems optimize the use of available spectrum resources. This reduces the need for separate frequency allocations and addresses the challenge of spectrum scarcity.
- **Cost and Space Savings**: Integrating communication capabilities into radar systems eliminates the need for separate communication hardware. This leads to cost savings and reduces the space required for hardware components, which is particularly beneficial in automotive applications.
- **Flexibility and Scalability**: RadCom systems can be easily adapted to support various communication standards and protocols. This provides flexibility and scalability for future developments, making RadCom a versatile solution for evolving automotive technologies.

Overall, the RadCom approach presents a promising solution for the efficient co-existence of radar and communication systems. RadCom systems pave the way for advanced automotive applications and improved spectrum utilization.

Nonetheless, RadCom systems present several challenges, including:

- **Complexity of Waveform Design**: Designing waveforms that can simultaneously support radar and communication operations is a complex task. Achieving an optimal balance between radar detection performance and communication data rates requires sophisticated design and optimization techniques.
- **Interference Management**: Managing interference between radar and communication signals is a primary challenge. Techniques such as interference cancellation and beamforming are employed to mitigate this issue. Interference cancellation techniques involve estimating and subtracting the interference signal from the received signal, while beamforming techniques use directional antennas to focus the transmission and reception of signals in specific directions. However, by using PMCW waveforms, interference mitigation is inherently achieved without the need for additional techniques as mentioned before. PMCW waveforms use orthogonal codes, which help in distinguishing between different signals and reducing interference.
- **Robustness of Signal Processing Algorithms**: The shared spectrum environment poses significant challenges for signal processing algorithms. These algorithms must be robust enough to handle the complexities of separating and decoding the received radar and communication signals without compromising performance.
- **Hardware Implementation**: Implementing RadCom systems in hardware requires careful consideration of the trade-offs between complexity, cost, and performance. Advances in hardware technology, such as Software-Defined Radios (SDRs), have facilitated the development of RadCom systems. SDRs provide a flexible platform for implementing RadCom systems by allowing the modulation, demodulation, and signal processing functions to be implemented in software. In this project, dedicated hardware components were used to achieve the desired functionality. This includes the use of high-performance Field Programmable Gate Array (FPGA) kits for tasks such as data acquisition, control, and interfacing with other components, custom-designed Radio Frequency (RF) front-ends for efficient transmission and reception, and the signal processing was performed on a Personal Computer (PC), which received the data in real-time from the FPGA kit. These components were chosen to meet the stringent requirements of both radar and communication functions.
- **Resource Allocation**: Efficient allocation of resources, such as power and bandwidth, is critical to ensure the optimal performance of both radar and communication functions. Resource allocation algorithms are designed to dynamically adjust the allocation based on the current operating conditions. For example, power allocation algorithms can be used to distribute the available power between radar and communication functions to maximize the overall system performance. Additionally, the

complexity and processing power needed for joint resource allocation in RadCom systems can be significant. Advanced algorithms must be employed to manage the allocation of available processing power between radar and communication tasks, ensuring that both systems meet their performance requirements.

High-level Radcom System Architecture

The block diagram in Figure 1.3 illustrates the principal procedures at the Transmitter (Tx) and Receiver (Rx) of a RadCom system based on a PMCW approach elaborated previously. At the Tx, the ranging and communication sequences are combined and up-converted with a LO and then sent to the RF stage. This signal propagates through the wireless channel and is received by the receiver's RF stage. The received signal then passes through the LO or downconversion and is then digitized by the ADC. Subsequently, two separate processors handle the received radar and communication processes. The radar processor extracts the ranging information, while the communication processor performs frequency and phase recovery, payload extraction, and bit decoding. These two processes are organized in a Time Division scheme.

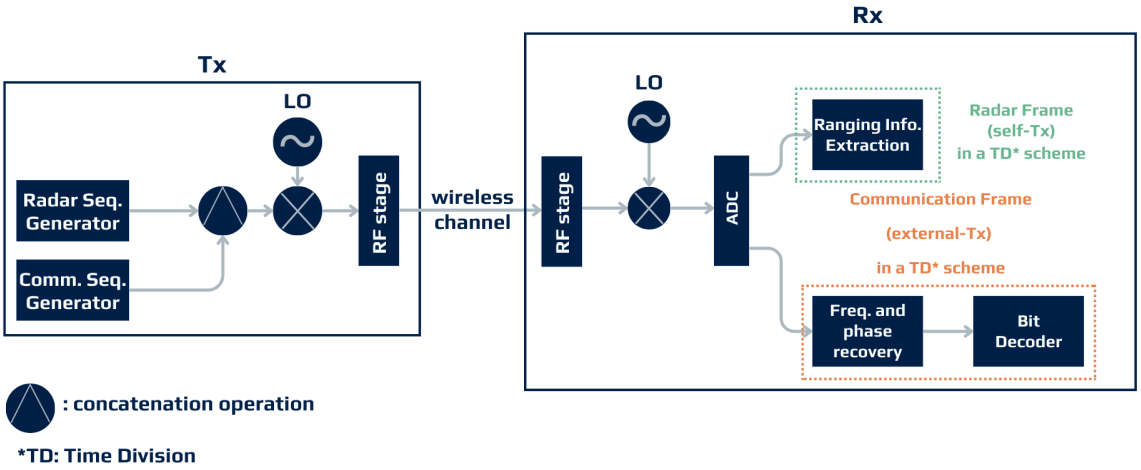


Figure 1.3: Block Diagram Illustrating the Principal Procedures at the Tx and the Rx of a RadCom System

Advantage of PMCW Radars in RadCom

The co-existence of radar and communication systems offers a promising solution to the challenge of spectrum scarcity. By leveraging advanced waveform design, spectrum sharing, and signal processing techniques, RadCom systems can achieve efficient spectrum utilization while maintaining the performance of both radar and communication functions. Ongoing research in this field continues to address the challenges and explore new opportunities for the development of RadCom systems.
In this context, PMCW radars stand out due to their unique capabilities. One of the most significant benefits is their ability to enable communication between a secondary radar on

the automobile and a primary radar (like a base station) or between automobiles. This communication enables data fusion, leading to improved functionalities and more accurate situational awareness. Integrating data from multiple radar sources enhances the system's overall performance, providing spectrum sharing and better detection, tracking, and decision-making capabilities.
In addition to data fusion, PMCW radars offer additional previously elaborated advantages such as interference mitigation, no range-Doppler coupling, enhanced resilience to interference, and higher maximum velocity. These benefits collectively contribute to the efficient co-existence of radar and communication functionalities, making PMCW radars a versatile and powerful solution for automotive radar applications.

1.4 Specific Radar System Prototype

Note that a comprehensive explanation of the hardware components, accompanied by an elaborate block diagram, is provided in Chapter 2.

The Prototype

Two binary-PMCW radar prototypes were built.

1. The initial one was originally built as an innovative solution with 1 Tx channel and 1 Rx channel. It was built to compare it with existing FMCW radar systems. That is why this prototype has been initially parameterized to copy as closely as possible the parameters of a typical FMCW radar discussed in detail in [1]. The transmitter of this prototype is constituted by the evaluation board of the ADMV7320 [4] 81–86 GHz band upconverter with typical saturation power P_{sat} = 26 deciBel-milliwatts (dBm). The receiver is constituted by the evaluation board of the ADMV7420 [5] 81–86 GHz low noise downconverter with baseband from Direct Current (DC), i.e. 0 hertz (Hz), to 2 GHz and a typical conversion gain of 10 deciBel (dB). The evaluation board with the ADF5610 [3] chip from Analog Devices generates a 14.04 GHz reference signal for both the upconverter and the downconverter which is then internally multiplied by a 6x factor to obtain an 84.24 GHz carrier for the modulation.

2. On the other hand, in the second prototype, the three chips: ADMV7320, ADMV7420, and ADF5610 were all integrated into a single custom-built RF board to be discussed in Chapter 2. Other than that, the primary components are essentially the same.

As introduced earlier, this radar system operates by transmitting a certain number of sequences of binary symbols modulating the phase of a carrier, called chips, with 0–π degree mapping and of duration T_{ch} each. An example of a transmitted signal modulating a carrier is shown before in Figure 1.2.
The transmitted sequence can be represented as:

$$s_{Tx}(t) = \sum_{n=0}^{N-1} g(t - nT_{ch}) cos(2\pi f_0 t + I_n \pi) \tag{1.1}$$

where $f_0 = 84.24$GHz is the carrier frequency, $I_n \in \{0,1\}$ is the sequence element, and $g(t)$ is a gate function of unit amplitude in the interval $[0, T_{ch}]$.

The received PMCW signal after being scattered by a target at a range R and with velocity $v = \frac{\lambda \times f_d}{2} = \frac{\lambda}{2 \times t_d}$ can be modeled as:

$$s_{Rx}(t) = A s_{Tx}(t - t_d) exp(j2\pi f_d t) \tag{1.2}$$

where f_d is the Doppler frequency-shift, t_d is the roundtrip delay time; i.e. duration between the transmitted and received pulses, R is the range given by: $R = \frac{c \times t_d}{2}$, and A is the complex amplitude that encapsulates all the effects of scattering, attenuations, losses, gains, and other propagation factors that the transmitted signal underwent before being received.
The Complex Correlation Data Matrix as shown in Figure 1.5 is constructed starting from the received In-phase (I) and Quadrature (Q) component samples. This matrix can be visualized as a 3-Dimensional (3D) matrix or two 2-Dimensional (2D) matrices, one for the I component and the other for the Q component. On the t-axis, we have the pulses, and for each pulse, there is a delay (range bin) axis.
For a train of pulses, which could be a continuous train as shown in Figure 1.4, the I and Q samples are received for each pulse. These samples are then correlated with the original transmitted sequence to obtain the correlation data, which has a length equal to the range bins for both I and Q components. This correlation data forms the Complex Data Matrix. Thus, the range response is directly derived by calculating the correlation between the received signal (s_{Rx}) and the transmitted signal (s_{Tx}).
Following this, for each range bin of delay (there are N_d delays), a Fast Fourier Transform (FFT) is applied to the columns, considering both the I and Q components of the entire pulse train at each range bin. This process results in the **range-Doppler** Matrix. Figure 1.6 shows an example of this matrix, where a target moving at approximately 100 m/s is positioned 400 meters away from the radar. By analyzing the different sequences, the Doppler-induced variations are extracted, allowing for the measurement of the targets' radial velocity.

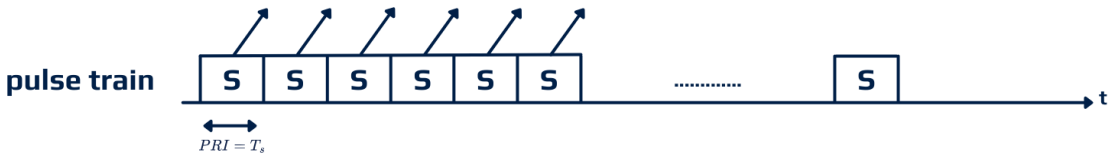


Figure 1.4: Example of a Continuous Transmission Pulse Train

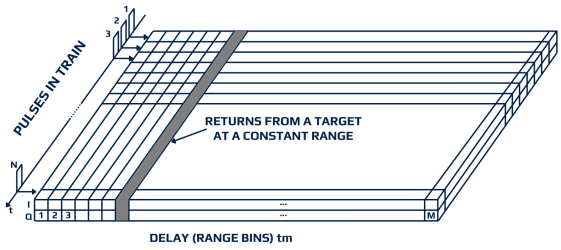


Figure 1.5: Complex Data Matrix Structure

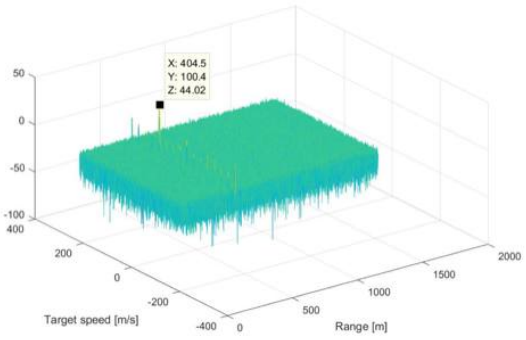


Figure 1.6: Range-Doppler Matrix Example

The range resolution of this system is equal to $R_{res} = \frac{T_{ch}c}{2} = \frac{c}{2B_{ch}}$ where the duration of the chip drives the bandwidth. Since PMCW radar relies on sampling a time signal, the maximum unambiguous range is limited, in case of continuous transmission of the same sequence, by the duration of the sequence (T_s). Therefore, the maximum unambiguous range, in case of continuous wave transmission (Figure 1.4), is $R_{max} = \frac{cT_s}{2}$ and it is independent of the bandwidth, providing a more flexible parameters optimization as mentioned earlier.
The previously mentioned advantage, derived from the signal nature of the PMCW system, that the range-Doppler estimation is not coupled is now substantiated by Equation 1.2.
Indeed, Figure 1.4 leads to notice a crucial trade-off in the design and parameterization of radar systems. In this figure's scenario, the Pulse Repition Interval (PRI) is set equal to the duration of the sequence T_s, resulting in the maximum Pulse Pulse Repetition Frequency (PRF). This configuration enhances the unambiguous velocity but simultaneously reduces the unambiguous range (R_{max}). In practical automotive radar systems, this trade-off must be carefully considered based on the specific requirements and expected functionalities. The choice between higher unambiguous velocity and greater unambiguous range depends on the operational context and priorities. However, with the PMCW approach, where the signal is digitized, these parameters can be dynamically adjusted. This flexibility allows for real-time optimization, enabling the system to prioritize one parameter over the other as needed.

Adopted Sequence

The sequences used in this radar system are **Golay sequences**. Golay sequences are known for their good autocorrelation properties, making them ideal for radar systems where accurate detection and range estimation are crucial. They are generated using specific algorithms and have been chosen for this radar system due to their ability to significantly reduce the interference caused by other radar systems operating in close proximity.
This type of code is formed by complementary pairs that satisfy the properties of having out-of-phase aperiodic auto-correlation coefficients' sum equal to zero [9].

Let

$$\underline{a} = \{a_i; \quad 0 \leq i \leq N-1\} = (a_0, a_1, a_2, ..., a_{N-1}); \quad a_i \in \{\pm 1\} \tag{1.3}$$

be a sequence of length N and its complementary pair be

$$\underline{b} = \{b_i; \quad 0 \leq i \leq N-1\} = (b_0, b_1, b_2, ..., b_{N-1}); \quad b_i \in \{\pm 1\} \tag{1.4}$$

The sequences are generated utilizing recursive construction. To generate the next code the previous code is concatenated to its complement. The previous code is concatenated to the inverse of the previous code to generate the complement code. An overview of the generation algorithm is shown in Figure 1.7.

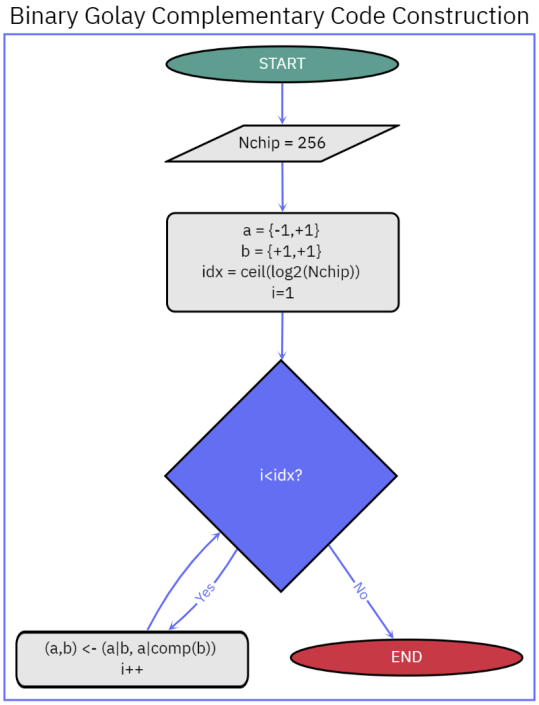


Figure 1.7: Golay Complementary Code Construction Flow Chart
N.B. '|' signifies the concatenation operator

Figure 1.8 illustrates a Golay complementary pair for $N = 256$, which is utilized in the system shown under ideal conditions, i.e. without noise.

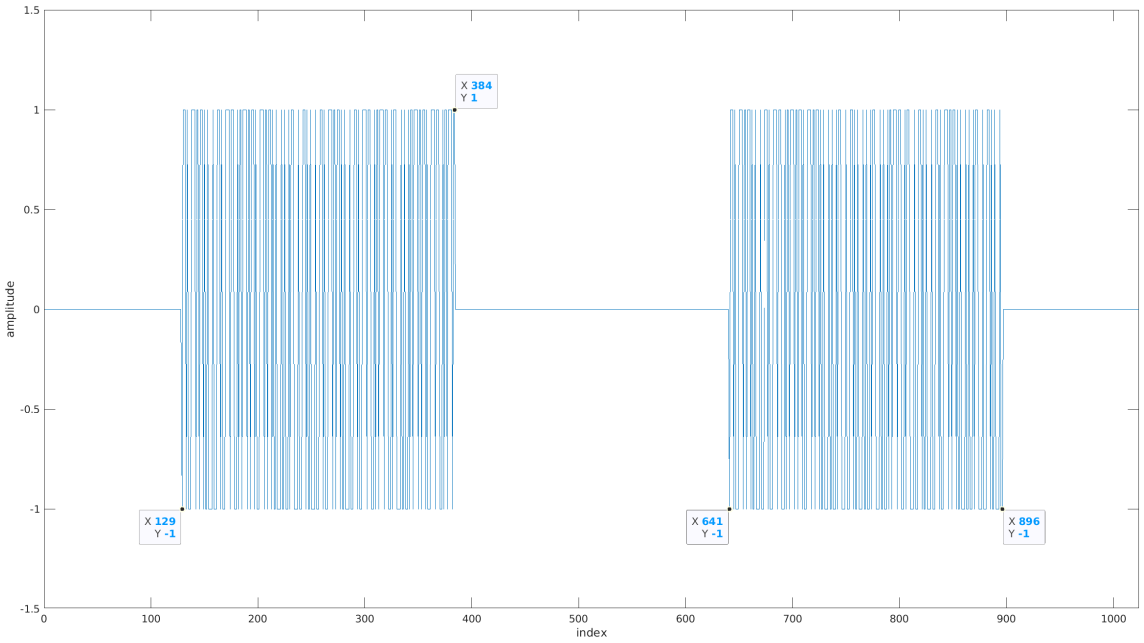


Figure 1.8: Golay Complementary Pair for $N = 256$ under Ideal Conditions

Define separately the two aperiodic auto-correlation functions as:

$$\rho_{a,b}(k) = \sum_{i=0}^{N-k-1} (a,b)_i \times (a,b)_{i+k}; \quad 0 \leq k \leq N-1 \tag{1.5}$$

For the Golay complementary pair we obtain,

$$\rho_a(k) + \rho_b(k) = 0; \quad k \neq 0 \tag{1.6}$$

This property ensures that the sum of the responses of the complementary pair, transmitted sequentially, cancels out the sidelobes and doubles the peaks. As a result, there is a $2\times$ improvement in SNR and zero sidelobes when the signal is in ideal conditions (no noise) as in Figure 1.8. This property is illustrated in Figure 1.10, which shows both the ideal correlation function and an example of the correlation function in real-life applications where noise is present. In the real case, there are negligible sidelobes due to the presence of noise. An example of the corresponding signal in real-life conditions for a slightly high SNR value of 10 dB is shown in Figure 1.9.

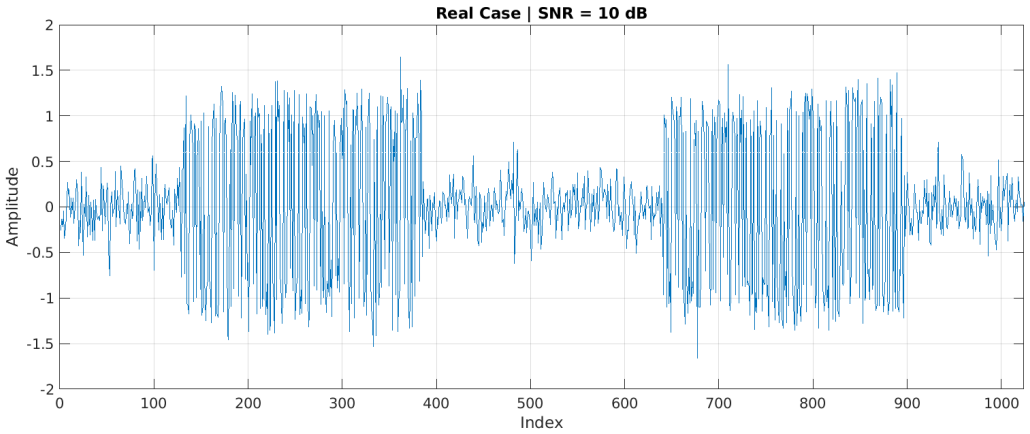


Figure 1.9: Real-life Signal Considering AWGN such that SNR = 10 dB

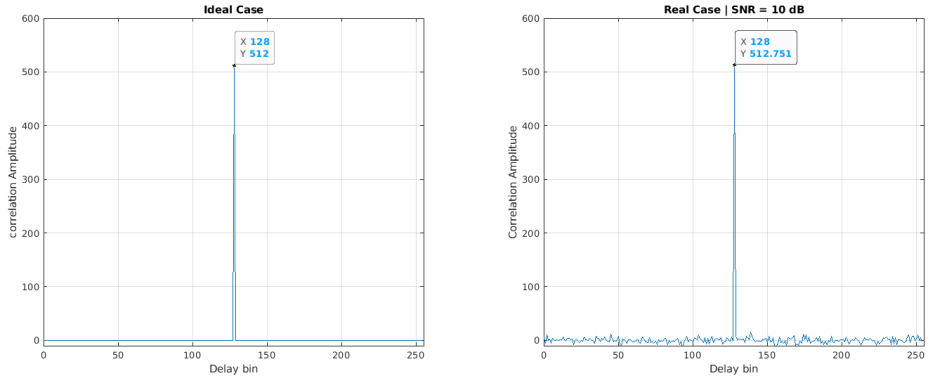


Figure 1.10: Correlation Function of both the Ideal signal shown in Figure 1.8 and an Example of the Real-life Signal shown in Figure 1.9

Notable, the peak of the correlation is at delay = 128 and with amplitude = 512. Both these numbers were anticipated because the signal has been delayed by 128 chips as could be inferred from Figure 1.8 while the amplitude value is due to the length of the complementary pair which is 512.

Radar Operations and Sequence Parameters

The binary-PMCW modulating sequences are generated by the FPGA hosted in a KC705 [11] development board as discussed in Chapter 2. The sequences are translated into differential signals and sent to the in-phase upconverter input. The downconverted differential, I and Q, received signals are sampled by an ADC chip hosted on the Data Aquisition (DAQ2) board connected to the same FPGA. The FPGA receives the I and Q raw data sampled by the ADC, and then arranges them in User Datagram Protocol (UDP) packets and sends them to a PC for processing and further analysis. This reception mechanism is illustrated at a high level in Figure 1.11. Comprehensive details on both hardware and firmware components are provided in their respective chapters.

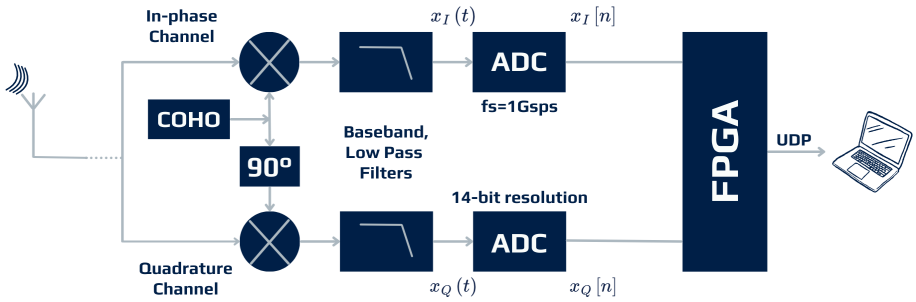


Figure 1.11: High-Level Overview of the Reception Mechanism

The binary-PMCW frame, depicted in Figure 1.12a, is composed of 128 complementary sequences (64 pairs). Each sequence has 256 chips with a duration of $T_{ch} = 4$ nanosecond (ns) and with a bandwidth of $B_{ch} = 250$ MHz. In this case, the range resolution depends on the bandwidth and is given by:

$$\delta R = \frac{T_{ch} \times c}{2} = \frac{c}{2 \times B_{ch}} = 0.6m \tag{1.7}$$

The range accuracy, R_{acc}, depends on the ADC sampling frequency (f_s). Since $f_s = 1$GHz, then each delay is $\tau = 1$ ns and $R_{acc} = \frac{\tau c}{2} = 0.15$m.

For each complementary sequence pair the correlation for a certain number (N_d) of delays (τ) between the transmitted sequence and the received I and Q signal is performed. The results are summed yielding a range response $R_{ab}(\tau)$

$$R_{ab}(\tau) = \sum_{i=1}^{N_d} a(t).a_{rx}(t + i\tau) + \sum_{i=1}^{N_d} b(t).b_{rx}(t + i\tau) \tag{1.8}$$

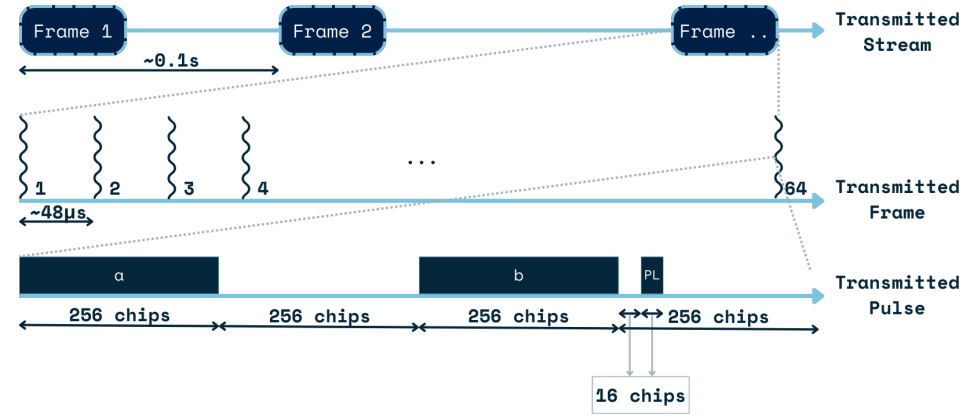
As will be elaborated on later, the a and b complementary sequences are transmitted with a PRI $\approx 48\mu s$ (PRF $\approx$ 20.83 kilohertz (kHz)). Thus there is around $44\mu s$ blank time interval between the transmission of each pulse, i.e. each complementary pair, equivalent to 11,000 zero chips. **Notably**, the frame repetition rate (f_r) is 100 millisecond (ms). Table 1.1 shows a summary of the main parameters of this system.

Symbol	Description	Value
B_{ch}	Bandwidth	250 MHz
T_{ch}	Chip Duration	4 ns
N	Number of chips in a sequence	256
N_p	Number of complementary pairs	64
f_s	Sampling frequency	1 GHz
f_r	Frame repetition rate	100 ms
δR	Range resolution	0.6 m
R_{acc}	Range accuracy	0.15 m
PRF	Pulse Repetition Frequency	20.83 kHz
δV	Velocity resolution	0.6 m/s
V_{max}	Max unambiguous velocity	19.2 m/s

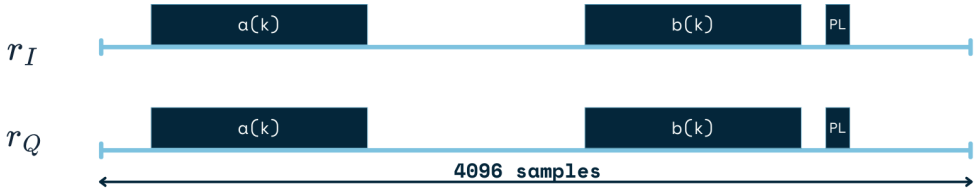
Table 1.1: Key PMCW Radar Prototype Parameters

Data Protocol and Structure

As briefly aforementioned, and as shown in Figure 1.12a, the transmitted pulse pairs are arranged in frames. Each frame consists of 64 pulse pairs. Each pulse is made up of 1,024 chips. The time duration for which 2 consecutive pulses within a frame are sent (PRI) is around $48\mu s$. Frames are transmitted every 100 ms.
As per the received pulse, there are the I and Q sampled signals received by the FPGA. Each pulse pair, over I and Q, is made up of 4,096 samples corresponding to 1,024 chips (one chip is 4x sampled) as illustrated in Figure 1.12b. Since the precision of the ADC chip is 14-bit, as discussed in 2.2.2, each I/Q sample is over 14 signed bits. The buffer storing data is structured so that at most it fits the data of a whole frame both on I and Q at a time.
The FPGA is programmed to send the received data in UDP packets of fixed length (1026 payload bytes). Each packet carries 256 samples, thus 16 packets are needed to send a whole pulse on I or Q.



(a) Transmitted Stream



(b) Sampled Received Pulse
N.B. PL:PayLoad

Figure 1.12: Data Structure

1.5 Communication System Technology

Since mm-wave signals encounter a complex propagation environment characterized by multipath, high scattering, and severe penetration losses, the mm-wave communication link suffers drastically from losses. This fact leads to having these links near Line-Of-Sight (LOS). Multipath occurs when transmitted signals reflect off various surfaces, such as buildings, vehicles, and other obstacles, before reaching the receiver. These multiple reflected paths can cause constructive and destructive interference, leading to signal fading, phase shifts, and time delays. In the context of this project, multipath propagation is particularly relevant due to the high frequency of mm-wave signals, which make them more susceptible to reflections and scattering. Consequently, these challenges must be addressed by the final developed system and should be considered for future work.
In this project, both the Tx and the Rx of the communication system employ the same antennas and hardware. The waveform has been designed following a 'radar-centric' approach. As mentioned, a PMCW approach is used to extract the target's phase and velocity information. The receiving functionality of the final unit operates in two modes according to a Time Division scheme:

1. Radar-only process mode: the radar unit operates normally as it was built to do

and as introduced earlier. The system employs a non-preemptive priority approach, where the ranging mode has higher priority over the communication mode to keep the radar functionalities intact.

2. Radar-communication process mode: the radar unit receives external signals and processes them to infer the received payload while not interrupting the ranging functionalities. However, it could be possible that the ranging data saved in the buffer are delayed before being sent to the laptop PC for processing. This possible delay is bounded by the duration of a whole Communication signal being received minus one clock cycle, thus by 4,092ns.

The FPGA is also responsible for generating the communication payload bits to implement the transmission functionality. A fixed-position fixed payload has been integrated within the transmitted pulses of a frame in the digital domain. Such that a meaningful payload is the one demodulated and inferred from a whole frame (64 pulses).
Straightforward, Binary Phase-Shift Keying (BPSK) modulation has been utilized. Although it could be possible to use higher modulation formats. That reflects on the relatively low throughput of this communication system, which is currently 10.24Kilobits per second (Kbps).
At the receiver of the PMCW radar unit, blind frequency compensation has been implemented adopting an FFT-approach [10]. Whereas phase synchronization and recovery have been implemented based on the correlation of the received I and Q signals with the transmitted radar signal excluding the payload (the complementary pair a and b as in Equations 1.3 and 1.4) for the N_d delays, the latter two processes are thoroughly explained in Chapter 4 in **'Carrier Frequency Offset Compensation'** and **'Phase Recovery and Synchronization'**.

1.6 Project Overview

The primary objective of this project is to integrate a communication system into an existing binary PMCW radar system. This radar system, which utilizes Golay sequences for BPSK modulation, has been designed with 1 Tx channel and 1 Rx channel. It operates in two receiving modes according to a Time Division scheme: Radar-only process mode and Radar-Communication process mode.
The communication system operates in the same mm-wave frequency, utilizing the same hardware as the radar system. Due to the propagation characteristics of such signals, the communication system operates near LOS. The integration of this communication system into the radar system aims to enhance the functionality and versatility of the radar system without disrupting its existing operations.
In addition to the integration of the communication system, the project also involves analyzing the performance of the integrated system in real-world conditions. The challenges and limitations of this integrated system will be identified, and solutions will be proposed to overcome these challenges and improve the system's performance.

Automotive Applications

The integration of communication capabilities into the existing binary PMCW radar system opens up a multitude of advanced automotive applications. This enhancement not only improves the functionality and versatility of the radar system but also significantly contributes to the development of more efficient and effective ADASs. Below are several key applications where this integrated system can be particularly beneficial:

Enhanced Collision Avoidance

One of the primary applications of the integrated radar-communication system is in enhanced collision avoidance. In a traditional radar-only system, the radar can detect obstacles and provide warnings to the driver. However, without communication capabilities, the system cannot share this critical information with other vehicles.

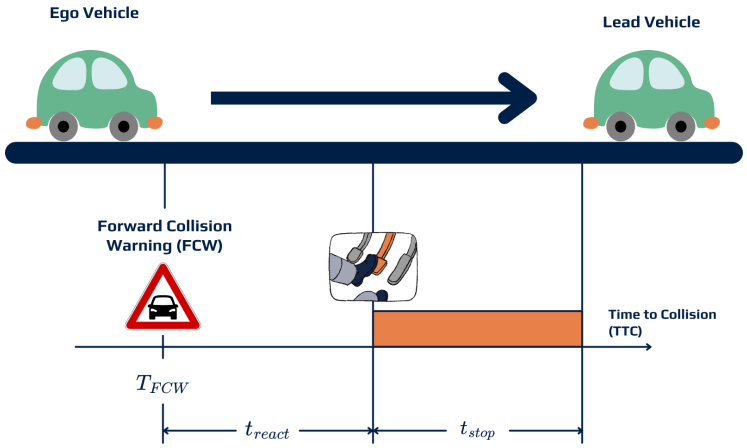


Figure 1.13: Forward Collision Handling Example

The example illustrated in Figure 1.13 could be handled according to two scenarios:

- **Without the System:** If a vehicle detects an obstacle or sudden deceleration ahead, it can only alert the driver. The reaction time depends solely on the driver's response (t_{react}), which may not be quick enough to prevent a collision, especially in high-speed scenarios.

- **With the System:** The integrated system enables real-time communication between vehicles and between vehicles and infrastructure. When a vehicle detects an obstacle or sudden deceleration, it instantly communicates this information to nearby vehicles. These vehicles can then warn much in advance (greater T_{FCW}), or even take preemptive actions, such as adjusting speed or changing lanes, to avoid collisions. This real-time data exchange significantly reduces reaction times, minimizing t_{react}, and enhances overall road safety.

Cooperative Adaptive Cruise Control (CACC)

The integration of communication capabilities facilitates CACC, an advanced form of Adaptive Cruise Control (ACC). In CACC, vehicles communicate with each other to maintain optimal spacing and speed, improving traffic flow and reducing congestion as shown in Figure 1.14.

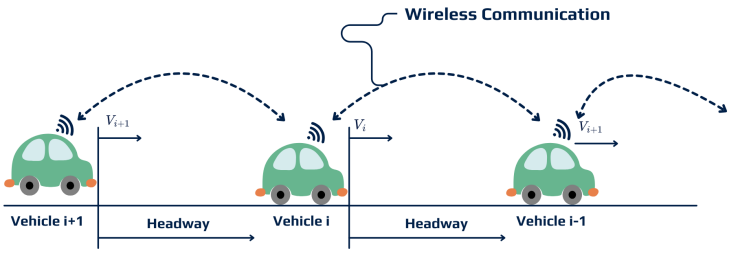


Figure 1.14: Example of a CACC Platoon

Without the System: Traditional ACC relies solely on radar to maintain a safe distance from the vehicle ahead. However, it cannot anticipate the actions of other vehicles further ahead, leading to abrupt braking and acceleration, which can cause traffic waves and increase fuel consumption.
With the System: The radar system provides precise range and velocity measurements, while the communication system ensures that all vehicles in the platoon are synchronized. Vehicles share information about their speed, acceleration, and braking intentions. This coordination allows for smoother acceleration and deceleration, reducing fuel consumption and emissions. Traffic flow is improved, and the risk of collisions is minimized.

Improved Lane Change Assistance

Lane change assistance systems benefit greatly from the integration of radar and communication technologies. The radar system detects vehicles in adjacent lanes, while the communication system exchanges information with those vehicles to confirm their intentions. For example, if a vehicle intends to change lanes, it can communicate this to nearby vehicles, which can then adjust their speed or position to facilitate a safe lane change. This dual capability ensures that lane changes are executed safely and efficiently, minimizing the risk of collisions such as in the example shown in Figure 1.15. Additionally, the system can provide real-time feedback to the driver, enhancing their situational awareness and decision-making.

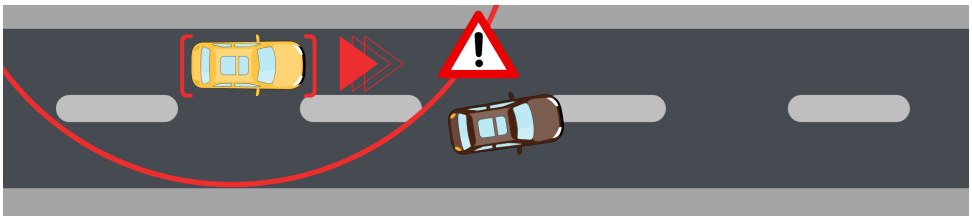


Figure 1.15: An Example Illustrating the Need for Lane Change Assistance

Intersection Management

Intersections are critical points where the risk of accidents is high. The integrated radar-communication system can significantly improve intersection management by enabling vehicles to communicate with each other as shown in Figure 1.16 and possibly with traffic signals or infrastructure in general.

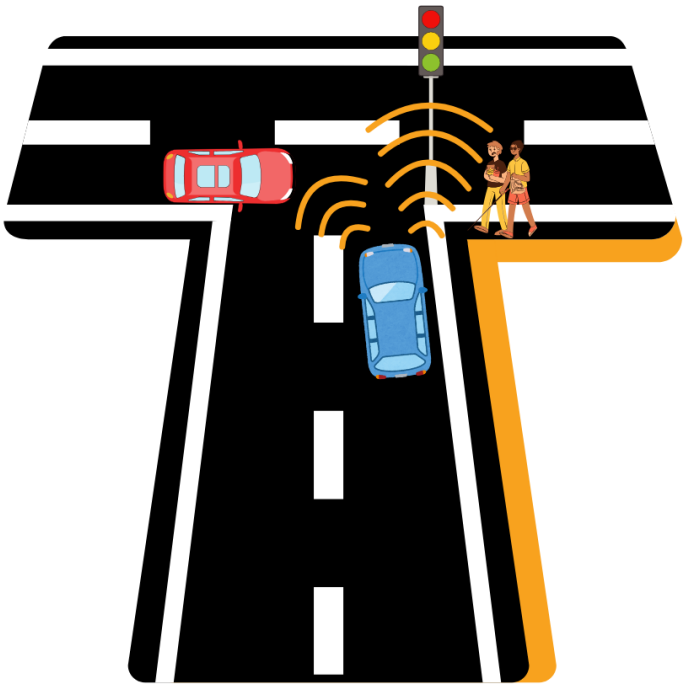


Figure 1.16: Intersection Traffic Example Scenario

Without the System: Drivers rely solely on traffic signals and their vehicles' own sensors to navigate intersections. This can lead to misjudgments, especially in complex intersections with multiple lanes and high traffic volume.
With the System: Vehicles approaching an intersection can receive information about the timing of traffic lights and the presence of other vehicles. For example, if a vehicle is about to run a red light, it can communicate this to other vehicles, allowing them to take preemptive actions. This communication allows for the coordination of vehicle movements,

reducing the likelihood of collisions and improving traffic flow.

Emergency Vehicle Coordination

In emergencies, the integrated system can facilitate the coordination of emergency vehicles with regular traffic. By communicating their presence and intended route, emergency vehicles can navigate through traffic more efficiently, reducing response times.
Without the System: Emergency vehicles rely on sirens and lights to alert other drivers. However, drivers may not always respond promptly or correctly, leading to delays or potential accidents.
With the System: The communication system provides real-time updates to all parties involved. For example, an ambulance can communicate its route to nearby vehicles, which can then clear the path. This coordination enhances the safety and effectiveness of emergency response operations.

Enhanced Situational Awareness

Overall, the integration of communication capabilities with the radar system enhances situational awareness for both drivers and autonomous vehicles. By providing a comprehensive view of the surrounding environment and enabling real-time data exchange, the system ensures that all road users are informed and can make better decisions.
Without the System: Drivers and autonomous vehicles rely solely on their own sensors to gather information about the environment. This can lead to blind spots and misjudgments, increasing the risk of accidents.
With the System: The integrated system provides a comprehensive view of the surrounding environment by combining radar data with information from other vehicles and infrastructure. For example, a vehicle can receive information about an obstacle around a corner that is not yet visible to its own sensors. This enhanced situational awareness contributes to safer and more efficient driving, ultimately leading to a reduction in accidents and improved traffic management.

The successful completion of this project will demonstrate the feasibility and benefits of integrating a communication system into a PMCW radar system, paving the way for the development of more advanced and versatile radar systems in the future. This integration is a crucial step towards the realization of fully connected and autonomous vehicles, contributing to the development of more efficient and effective ADASs.

Chapter 2

Hardware Description

The radar system, as depicted in Figure 2.1, comprises several key components, each contributing to the system's overall functionality.

2.1 Hardware Components

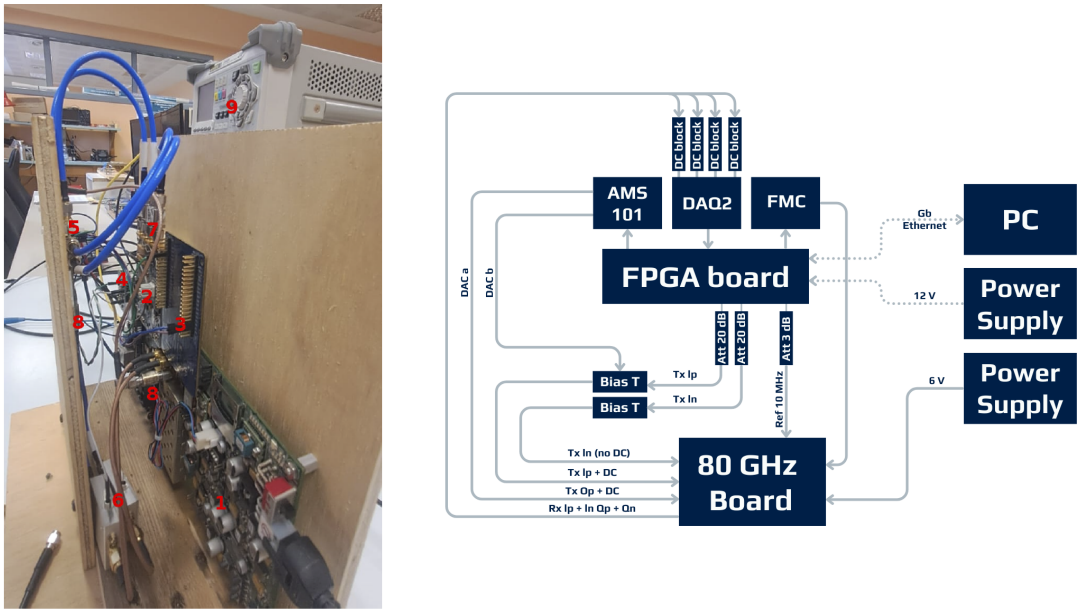


Figure 2.1: Hardware Configuration and Connectivity of the Radar System

The main components as shown in Figure 2.1 are:

1. A FPGA board (KC705 by Xilinx).
2. A DAQ2 Board (AD-FMCDAQ2-EBZ by Analog Design)

3. An FPGA Mezzanine Card (FMC) expansion board (CYUSB3ACC-005 by Cypress)
4. A dual Digital-to-Analog Converter (DAC) board (AMS101 by Xilinx).
5. An 81-86 GHz RF Transceiver board
6. Bias Tee
7. DC-blocks and attenuators
8. A power supply
9. A Linux-based laptop PC

2.2 Detailed Components' Description

In this section, a thorough description of the specifics of each component and how they contribute to the overall functionality of the radar system is presented.

2.2.1 FPGA Board

The system's central processing unit is the KC705 Evaluation Kit by Xilinx [11]. This kit is centered around the Kintex-7 FPGA, a highly flexible and programmable platform that forms the digital core of the radar system. The Kintex-7 FPGA is renowned for its high-performance capabilities, making it an ideal choice for integrating communication capabilities in PMCW automotive radars. The board is equipped with a multitude of peripherals, providing the flexibility to interface with various components of the system as shown in Figure 2.2. It's worth noting that this board is not just a piece of hardware, but a comprehensive kit that includes design tools, Intellectual Property cores, i.e. reusable "soft" blocks (IP cores), and pre-verified reference designs. This aids in rapid prototyping and development, significantly reducing the time from concept to implementation.

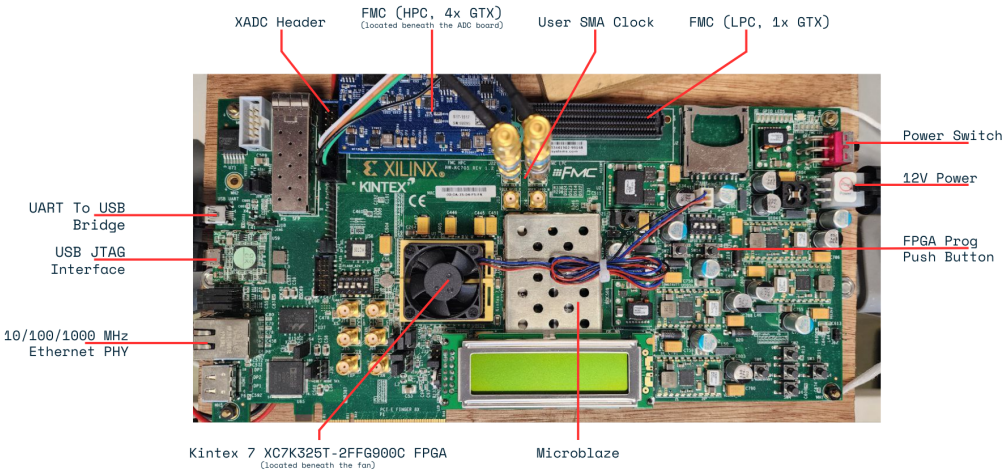


Figure 2.2: Annotated FPGA Board Showing Key Components

2.2.2 DAQ2 Board

The AD-FMCDAQ2-EBZ board by Analog Design [2] shown in Figure 2.3 serves as the primary data acquisition component which is connected to the FMC High Pin Count (HPC) connector of the FPGA board through JEDEC Standards (JESD). This board is equipped with a dual 1 Gsps ADC chip (AD9680) which has 4 channels with 1 GSps and 14-bit precision.

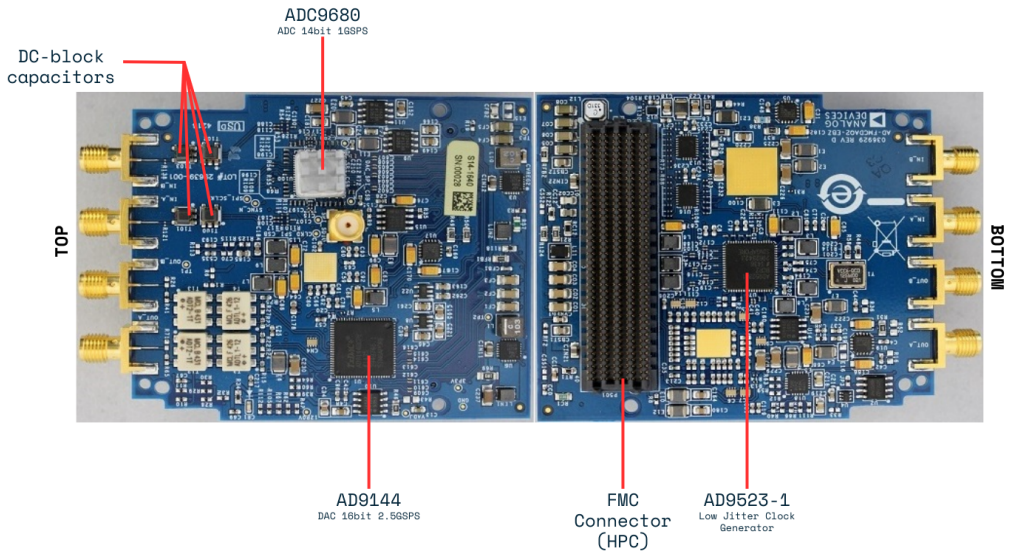


Figure 2.3: Annotated DAQ2 Board Showing Key Components

The ADC signals are used as double-ended to enhance the signal integrity and noise performance. The interface between the ADC and the FPGA is facilitated by a JESD204B serial bus, ensuring high-speed data transfer as depicted by the block diagram in Figure 2.4. One key modification that has been made to the board is the shorting of the DC-block capacitor. External DC blocks are used instead to allow matching precisely the frequency of the captured signals to our requirements. This change is reflected in Figure 2.5.

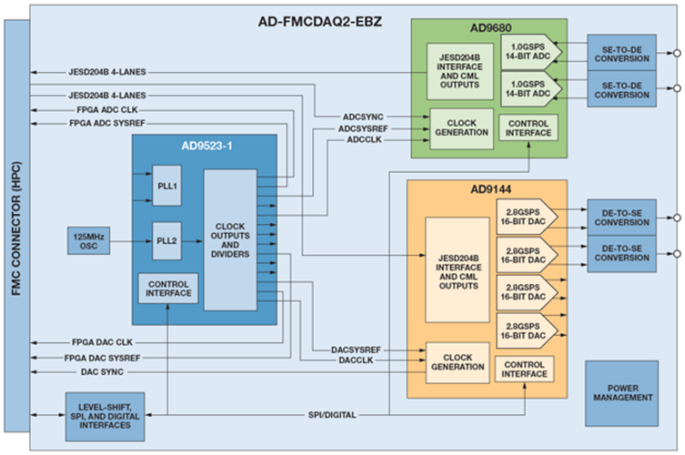


Figure 2.4: DAQ2 Board Block Diagram [2]

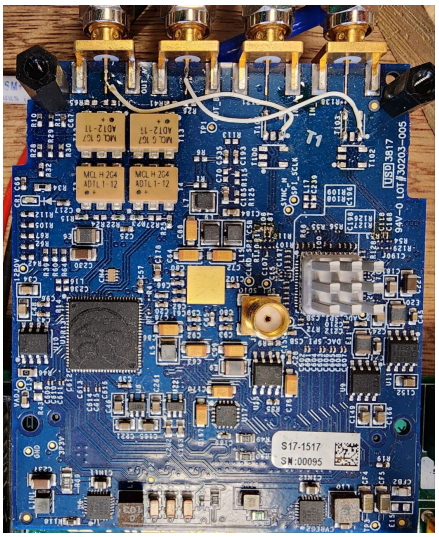


Figure 2.5: Modified DAQ2 Board - Front Side

2.2.3 RF Transceiver Board

The 81-86 GHz RF board is a custom-designed component (by the team that designed this PMCW radar prototype) that plays a crucial role in the system. It comprises several key elements:

- **E-band Upconverter (ADMV7320)** [4]: This component, provided by Analog Devices, operates in the 81-86 GHz frequency range. It features an on-chip waveguide launcher, which facilitates the efficient transmission of signals.

- **E-band Downconverter (ADMV7420)** [5]: Also provided by Analog Devices, this downconverter operates in the same frequency range as the upconverter and also features an on-chip waveguide launcher. It is responsible for converting high-frequency input signals into lower-frequency signals that are easier to process.

- **PLL Chip (ADF4372)** [3]: This Phase-Locked Loop (PLL) chip, provided by Analog Devices, generates the LO signal to feed the up/down converters.

The Tx and Rx launchers are connected to two different horn antennas. Figure 2.6a shows the transmitter antenna while Figure 2.6b shows that of the receiver.

(a) Tx Horn Antenna (b) Rx Horn Antenna

Figure 2.6: System's Antennas

The Tx and Rx channel antennas are standard gain horns with 10.5dB and 20dB gains respectively.

The RF board is powered by a power supply different than that used to supply the FPGA board, ensuring a consistent and stable power source. It is cooled by a fan mounted on the chassis, which helps to maintain an optimal operating temperature.

2.2.4 Complementary Components

Several additional and indispensable components were integrated to ensure the system operates as expected and achieves its intended functionality. Each of these components serves a unique purpose and their collective operation is crucial for the system's performance. These components and the other ones mentioned previously work in harmony to ensure the system performs at its peak. The most critical among these are:

FMC Expansion Board

The CYUSB3ACC-005 board, provided by Cypress, enables the addition of the Serial Peripheral Interface (SPI) to program the 81-86 GHz PLL chip. This board is connected to the FPGA via the FMC Low Pin Count (LPC) connector, granting access to extra FPGA pins.

AMS101 Board

The AMS101 [12] board, a product of Xilinx, is equipped with a dual DAC chip. This chip is programmed by the FPGA. The voltage output from this DAC is utilized to compensate for the LO leakage at the I and Q inputs of the 81-86 GHz transmitter (LO nulling). To achieve a suitable output voltage range, modifications were made to the output series resistor, changing its value from 5000 Ohm to 820 Ohm. This adjustment allows for more precise control over the output voltage, thereby enhancing the system's performance.

Bias Tee

The system incorporates two bias tees that are connected to the 'I' channels (positive and negative) of the Tx signal of the 81-86 GHz board. These bias tees play a crucial role in adding a DC component, generated by the AMS101 board, to the signal, which is essential for LO nulling.
The LO nulling technique eliminates the unwanted LO signal that can leak into the RF output.
The bias tees and their functionality are per the specifications provided in the ADMV7320 [4] datasheet.

DC-Blocks

DC-blocks are essential components in the system that are strategically placed in front of the SubMiniature version A (SMA) connectors of the DAQ2 board. Their primary function is to decouple signals, effectively blocking any DC component of a signal from passing through. This ensures that only the desired frequencies are transmitted, enhancing the overall performance and reliability of the system.

Attenuators

By setting the signal levels into the correct ranges, attenuators prevent signal overload and ensure that the system operates within its optimal range.
In this setup, there are three main attenuators. Two 20dB attenuators are placed between the FPGA board and the two Bias Tees. A 3dB attenuator is positioned between the FPGA board and the RF Transceiver board on the reference signal connection. This reference signal is fed to the RF board, which is internally multiplied to generate the desired carrier frequency.

Power Supply

The system requires two distinct power rails to supply the needs of different boards:

- A 12V 2.5A external power rail is used for the KC705 board. This ensures that the board receives the necessary power for its operation, contributing to the overall performance and reliability of the system.
- A 6V 2.2A power rail is used for the 81-86 GHz RF board. This power rail is carefully chosen to match the power requirements of the RF board, ensuring its optimal operation.

PC with Linux

A PC with Ubuntu 18.04 serves as the primary interface for user interaction and data processing in the system. It interfaces with the radar system over a 1 Gigabits per second (Gbps) Ethernet connection, providing a robust and high-speed link for data transfer.

Chapter 3

Firmware Description

3.1 Initial State

The starting status of the PMCW automotive radar firmware system was composed of the FPGA Very high-speed integrated circuit Hardware Description Language (VHDL) code. The latter is the system's backbone. It is composed of several sub-blocks, each with a specific function. The main blocks' overall key functions and behavior flow are depicted in the block diagram of Figure 3.1.

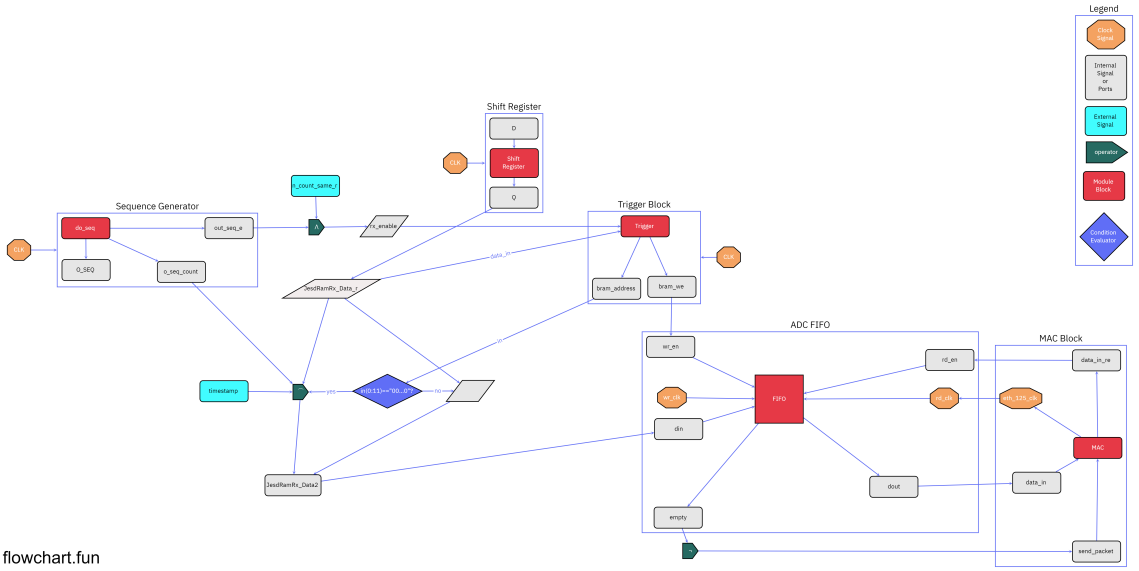


Figure 3.1: Block Diagram Showing the Basic Behaviour of the 'Firmware at its Initial State

Regarding the above block diagram, note that:

1. '$\wedge$' signifies the logical 'and' operator, '$\frown$' signifies the concatenation operator, and '$\neg$' signifies the logical 'not' operator.

2. All the clock signals named 'CLK' are synchronized to a clock signal '`JesdRamRx_Clk`' mentioned below operating at 250MHz. Conversely, the '`eth_125_clk`' signal serves as a Double Data Rate (DDR) clock signal, functioning at a frequency of 125 MHz.

3.1.1 Jesd204 Transceiver Interface

Interfaced with the high-speed serial lane that connected the ADC and DAQ2. The Jesd204 transceiver Interface to DAQ2 board block is a crucial component of the radar system. It primarily comprises several sub-components, each serving a specific function in the overall system.

- **Jesd204b_Rx_Interface:** It is a key component of the radar system. It manages data reception from the high-speed serial lane that connects the ADC and DAQ2. This interface includes several sub-components that handle tasks such as demapping the transport layer, checking the ramp of the received data, managing the reset and clock signals for the GT transceivers, supporting the transmission of data, providing the status of the transmission, and mapping the transport layer for transmission. Each sub-component is crucial in ensuring that data is accurately transmitted and received.
- **Jesd204b_Rx_TransportLayerDemapperRamp:** It is responsible for demapping the transport layer of the received data. It operates in a clocked environment, extracting samples from the input data word for each channel when valid data is received. The demapped samples and error signals are then assigned to the output signals in the subsequent clock cycle. A '`ready_out`' signal is generated to indicate the availability of valid demapped data. This block ensures accurate processing of the received data in the radar system.
- **Jesd204b_Rx_RampChecker:** It checks the ramp of the received data by comparing it with the expected ramp sequence. The block operates in a clocked environment and maintains registers for each channel to store the received samples and error signals. It also maintains counters for each channel to count the number of errors. The block generates a pass signal, which is asserted if all received samples match the expected samples and no error signals are asserted. It also generates a '`check_valid`' signal, which is asserted when a valid check has been performed. It also provides status signals indicating the synchronization status, synchronization loss, and full status. These signals are useful for monitoring the operation of the block and diagnosing any issues.
- **Jesd204_phy_0:** Also known as JESD PHY (4.0) in Vivado, is a critical component in the radar system. It represents the physical layer of the JESD204 interface. This block manages the transmission and reception of data through the GT transceivers. It handles the reset and clock signals for the transceivers, ensuring that the data transmission and reception are synchronized with the system clock.
- **Jesd204b_Tx_Support:** It manages the transmission of data through the GT transceivers. It includes the "`Jesd204b_Tx_Interface`" component and operates in

a clocked environment. This block interacts with four GT transceivers (gt0, gt1, gt2, gt3), each having signals for carrying the received data, indicating the character risk, the disparity error, and whether the received data matches the expected patterns or sequences defined in the JESD204 standard. These signals ensure accurate processing and transmission of data in the radar system.

- **jesd204b_tx_Status:** This module in the radar system monitors the status of data transmission through the GT transceivers. It uses a clocked environment that uses '`tx_sync_event`' and '`tx_sync_loss`' registers to track synchronization events and losses respectively. The 32-bit '`jesd204_tx_conf`' input configures the JESD204 transmitter, while the '`jesd204_tx_status`' output provides the transmitter's status. This module ensures accurate monitoring and control of data transmission, contributing to the overall reliability of the radar system.
- **Jesd204b_Tx_TransportLayerMapper:** This module is a key component of the JESD204 interface that maps the transport layer for data transmission. It operates in a clocked environment and uses several registers and signals to manage the mapping process. The module takes in 8 16-bit samples from a single channel and concatenates them into a 128-bit '`tx_tdata`' signal for transmission. The '`tx_tready`' signal indicates when the transmitter is ready, and the '`ready_out`' signal is asserted when the transmitter is ready or when the fill buffer is not empty. This module ensures accurate mapping of the transport layer for data transmission

Each of these components plays a crucial role in the operation of the radar system. They work together to ensure that data is accurately transmitted and received.

3.1.2 AXI Controller

The VHDL code defines an **Advanced eXtensible Interface (AXI) Interface Controller** for the FPGA project. This controller interfaces with various system components, including a processor, a system monitor, and a JESD204B interface. The system monitor component monitors several system aspects, including voltage levels, temperature, and system clock and reset signals. Another key component is the 'FPGAController', which controls various aspects of the system including the AXI register interface and the Jesd204B interfaces. The code also includes a process that handles the read and write operations of the AXI interface. This process is sensitive to the rising edge of the '`s_AXI_AClk`' signal, which is the clock signal for the AXI slave interface. Depending on the current state of the read and write operations, the process updates various signals related to the AXI interface and the Jesd204B interfaces and transitions to the next state. This comprehensive design allows efficient control and monitoring of the FPGA project. This block is crucial for interfacing with other blocks and programming the MicroBlaze processor through a C-code. It provides a standardized protocol for communication between the FPGA and peripheral devices, enabling high-speed data transfer and low-latency communication. It supports multiple data channels, allowing simultaneous read and write operations. Additionally, it allows the MicroBlaze processor to access and control the FPGA's internal registers and peripherals through memory-mapped Input/Output (I/O).

3.1.3 Sequence Generator

It is a key component. Its primary function is to generate the binary phase modulating sequence to be transmitted.

- **Ports:** The Sequence Generator block has several input and output ports. Key input ports include '`I_CLK_250MHz`' (the clock signal input) mapped to '`JesdRamRx_Clk`' (explained in 3.1.5), and '`s00_axi_aclk`' (the AXI interface clock signal). Key output ports include '`O_SEQ`' (the generated sequence to be transmitted), '`out_seq_e`' (the sequence enable signal), and '`o_seq_count`'.
- **Functionality:** The Sequence Generator block generates binary sequences based on the input parameters and the internal logic defined in the VHDL code. The '`do_seq`' subblock is responsible for the actual generation of the sequences. It uses the clock signal and the clock-enable signal to control the timing of the sequence generation. The sequence to be transmitted is then output through the '`o_seq`' port. The '`out_seq_e`' port outputs a signal that indicates whether the sequence generation is currently enabled or not. The '`out_seq_count`' port outputs a count of the number of sequences that have been generated.
- **Interaction with other blocks:** The signals generated by the Sequence Generator block are used by other blocks in the system. For instance, the Trigger block uses the '`out_seq_e`' signal to control the data flow and operations of the system. The AXI Controller block also uses some signals to interface with various system components. The AXI Controller block provides the necessary interface for the Sequence Generator block to communicate with the rest of the system.

3.1.4 Trigger Block

This is a crucial component in the FPGA-based radar system. It manages the data flow and controls the operations based on the received signals implementing the logic for triggering data capture from the ADC and storing them into a First In, First Out (FIFO). Its input data is the output data ('`JesdRamRx_Data_2`': 128 bits/clock cycle) of the shift register block.

- **Ports:** This block has several input and output ports. These ports serve as the interface between the Trigger Block and other blocks in the system. For instance, the '`data_in`' input port receives '`JesdRamRx_Data_r`'. The '`rx_enable`' input port receives a condition signal to enable or disable the receiver. The output ports, '`bram_address`' and '`bram_we`', are used to interact with the Block Random Access Memory (BRAM) FIFO block, mapping the latter two signals to '`JesdRamRx_Addr`' and '`JesdRamRx_We`' defined in the top module of the whole VHDL project respectively.
- **Internal Signals:** The block uses several internal signals for its operations. These signals control the internal behavior of the block and are not visible to other blocks.

For example, the '`rx_enable_ttt`' and the '`rx_enable_tttt`' signals are used to delay the '`rx_enable`' signal by 3 and 4 clock cycles respectively. The '`in_address`' and '`in_bram_we`' signals control the BRAM operations as they are mapped to the output ports '`bram_address`' and '`bram_we`' respectively. One more important internal signal is '`dac_n_counter`' which is initialized to 0.

- **Functionality:** The block operates synchronously with the rising edge of the clock. On each clock cycle, it updates the delayed '`rx_enable`' signals, the address counter signal '`in_address`', and the '`in_bram_we`' signal. These updates control the flow of data and the operations of the block. When the '`rx_enable_ttt`' signal changes from 0 to 1 where '`rx_enable_tttt`' is 0, the signal '`dac_n_counter`' is assigned to $\texttt{FF}_{16} = 255_{10}$, thus signaling to empty the ADC FIFO to start a new sequence capture. Where the '`rx_enable`' signal takes as input the condition "`n_count_same_r` $\wedge$ `out_seq_e`", where '`out_seq_e`' is the enabler signal output of the block "sequence_generator" as discussed in 3.1.3. And, '`n_count_same_r`' is a one-clock cycle delayed signal of '`n_count_same`', and the latter is a signal that ensures that the current sequence transmitted is different from the one that has been transmitted in the previous clock cycle. So triggering the capture happens only once at the beginning of the transmission of each sequence.
  The block also controls the BRAM write enable input signal and the address signal through the '`bram_we`' and '`bram_address`' ports. When the last 12 bits of '`JesdRamRx_Addr`' are zero, it indicates that the system has successfully captured samples corresponding to 1024 chips, representing one complete sequence. At this point, additional information about the timestamp and sequence counter is appended at the beginning of each sequence, resulting in the signal '`JesdRamRx_Data2`' defined in the top module. The '`JesdRamRx_We`' signal, which enables the FIFO buffer block to start storing data, is set to 1 as long as '`in_bram_we`' is 1. This condition holds true when '`bram_address`', thus '`JesdRamRx_Addr`', is less than 1024 (1000000000_2). This behavior dictates when data should be written to the BRAM, influencing how the block interacts with the BRAM and other blocks that also interface with it. **Note** that '`JesdRamRx_Addr`' is what is referred to as 'in' in Figure 3.1.

  In summary, once the trigger is activated due to the initiation of a new sequence transmission, it remains active for 1024 clock cycles, allowing the entire sequence to be captured and stored in the buffer.

3.1.5 ADC FIFO

The ADC FIFO block is a crucial component in this system. It is one of the IP cores provided by the Vivado software, which is configured with 4 synchronization stages. It serves as a temporary storage for the data coming from the ADC before it is written to the BRAM. The FIFO operates on a first-in, first-out basis, meaning that the data that comes in first is the first to be output. The write width of the FIFO is 128 bits (represented by the input port '`din`'), the write depth is 65536, and the read depth is 128 bits (represented by the output port '`dout`').

- **Ports:** The ADC FIFO block has several input and output ports. The '`din`' input port receives the '`JesdRamRx_Data2`' signal from the top module. The '`wr_en`' input port receives the '`JesdRamRx_We`' signal from the top module, which enables or disables the write operation, so logically it receives the signal from the output port of the Trigger block '`bram_we`'. The '`rd_en`' input port receives the '`EthTx_re`' signal mapped to the output port of the MAC module '`data_in_re`', which enables or disables the read operation. The '`dout`' output port sends the data to the Ethernet Transmitter input port in the Media Access Control (MAC) module '`data_in`'. The '`empty`' output port sends the '`fifo_empty`' signal, indicating whether the FIFO is empty. The ADC FIFO block operates with two distinct clock signals, '`wr_clk`' and '`rd_clk`', mapped to '`JesdRamRx_Clk`' signal in the top module and the signal generated by '`eth_125_clk`', the output clock signal of the MAC block, respectively.
- **Functionality:** The ADC FIFO block operates synchronously with the rising edge of the clocks (the '`wr_clk`' signal governs the rate at which data is written into the FIFO, while the '`rd_clk`' signal controls the rate at which data is read from the FIFO). On each corresponding clock cycle, it checks the '`wr_en`' and '`rd_en`' signals. If '`wr_en`' is high and the FIFO is not full, it writes the data on the '`din`' port to the FIFO. If '`rd_en`' is high and the FIFO is not empty, it reads data from the FIFO and outputs it on the '`dout`' port. The FIFO also updates the '`empty`' signal based on its current state.
- **Interaction with Other Blocks:** The ADC FIFO block interacts with the Trigger Block and the MAC block. The Trigger Block controls when data is written to and read from the FIFO. The MAC block reads the data received and stored in the FIFO. The operation of the ADC FIFO block is crucial for the correct timing and synchronization of the system, as it buffers the data between the ADC and the BRAM.

3.1.6 MAC Module

The MAC module implements Ethernet communication, streaming UDP packets carrying the data from the FIFO to the Physical layer (PHY). This module is crucial for ensuring efficient data transmission over the network to the PC for processing.

- **Ports:** The MAC module features several input and output ports. Key input ports, excluding those related to internet standards or default connection management, include '`send_packet`' (signal to initiate packet transmission to the UDP port) and '`data_in`' (data input from the FIFO). In the same context, key output ports include '`Eth_Rx_clk`' (Ethernet receive clock), '`data_in_re`' (read enable signal for the FIFO), and '`eth_125_clk`'.
- **Sub-blocks:** The MAC module comprises several sub-blocks, each serving a specific function. Those include:
  - **byte_data:** This sub-block is pivotal in the MAC's module development in this project. It sets the UDP destination port address on an internal signal

('udp_dst_port') thus defining the packet flow and it initiates the transmission of this flow based on the signal at the output of the port 'send_packet'. It also organizes data for UDP packet transmission over the Ethernet interface, handling the construction of Ethernet frames, including the MAC, Internet Protocol (IP), and UDP headers.

- **add_crc32:** This sub-block adds a CRC32 checksum to the Ethernet frames to ensure data integrity during transmission.
- **add_preamble:** This sub-block adds a preamble to the Ethernet frames, which is required for synchronization at the receiver.
- **rgmii_tx:** This sub-block handles the transmission of Ethernet frames using the RGMII standard.
- **re_data_logic:** This sub-block oversees the data flow between the FIFO and the Ethernet interface, ensuring correct data formatting and transmission.

- **Functionality:** The MAC module operates synchronously with the rising edge of the clock signals. It handles Ethernet frame transmission, ensuring correct data formatting and network transmission. The module uses the Reduced Gigabit Media Independent Interface (RGMII) standard to streamline communication between the MAC sublayer and the PHY.
  This block supports a 1 Gbps Ethernet connection, with a UDP destination port number of 4096 (0x1000). The ADC operates at a speed of 1 Gsps, and the output clock of the PHY is 250MHz DDR, ensuring high-speed data transmission and reception.

- **Interaction with Other Blocks:** The MAC module primarily interacts with the ADC FIFO block, reading data from the FIFO and transmitting it over the Ethernet interface.

The MAC module is vital to the radar system, enabling efficient and reliable Ethernet communication.

3.1.7 Complementary Blocks

RAM-based Shift Register

The Shift Register block, named "c_shift_ram_0", is a component in the FPGA-based radar system. It is another one of the IP cores provided by the Vivado software configured as a Random-Access Memory (RAM)-based shift register with a fixed length. The shift register operates on a first-in, first-out basis. The width of the shift register is 128 bits and its depth is 64, meaning it can delay 64 entries of 128 bits each.

- **Ports:** The Shift Register block has three ports: 'D', 'CLK', and 'Q'. These ports serve as the interface between the Shift Register block and other blocks in the system. The 'D' input port receives the 'JesdRamRx_Data' signal the output of the **Jesd204b_Tx_Rx_Ip** module, which is the output of the **Jesd204b_Rx_RampChecker**

subblock. The 'CLK' input port receives the 'JesdRamRx_Clk' signal, which is the write clock of the FIFO as explained in 3.1.5. The 'Q' output port sends the 'JesdRamRx_Data_r' signal, which is used later in the top module and other blocks.

- **Functionality:** The Shift Register block operates synchronously with the rising edge of the clock. On each clock cycle, it shifts the data on the 'D' port into the register. The data is then available on the 'Q' output port. The Shift Register block plays a crucial role in the system's data flow, as it provides a delay element that can be used to ensure that the ADC can start sampling before the actual data arrives such that we can ensure that no samples are missed.
  So, because the depth was 64, there were 64 delays. In the time domain, it translates to $64 \times 4 = 256$ (ns) or 256 samples before the first sample, which is way more than necessary to ensure correct data reception.

Notably, the output of the Shift Register block, 'JesdRamRx_Data_r', is used in a process that is triggered on the rising edge of 'JesdRamRx_Clk'. If the address 'JesdRamRx_Addr' is all zeros, 'JesdRamRx_Data2' is assigned a value that includes a timestamp, 'out_seq_count', and a portion of 'JesdRamRx_Data_r'. Otherwise, 'JesdRamRx_Data2' is assigned the value of 'JesdRamRx_Data_r'.

Processor System Block Design and Clock Wizard

It includes a soft-IP microprocessor (uBlaze), memory, and I/O peripherals (SPI, GPIO, IIC) as well as generated various clocks of the system from an external reference.

3.2 Upgraded State

The firmware behavior has been upgraded and modified according to the needs of the final implemented demo system. By progressively adjusting the VHDL code of the system at the starting point, the final firmware behavior has been obtained that allows further capabilities. The main and essential modifications are done on several blocks and are highlighted below.

3.2.1 Sequence Generator

The sequence generator has been modified to read from a 1024-bit register fixed payload data in chunks (blocks) of 16 bits and integrates each chunk into the transmitted signal. Then after each 64 sent pulses (a single frame), which is equivalent to 1,024 transmitted payload bits, it restarts and then starts transmitting the payload all over again in the new frame.

Tests and Workflow Process

To achieve the aforementioned final behavior, there were several steps taken, out of them the most important transitions are:

1. A straightforward but indispensable change has been done on a register dictating the number of sequences to be transmitted in a frame. The new number is 64 instead of 65.

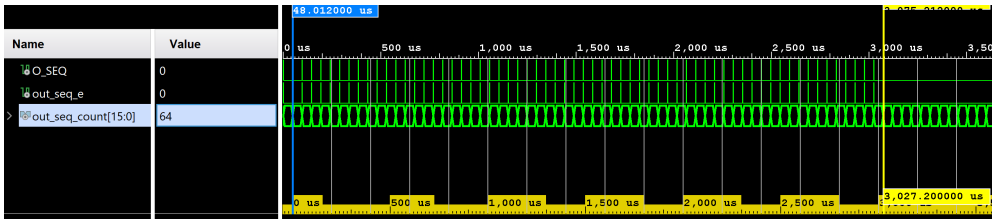
2. Generate the desired binary payload to be transmitted and saved in an AXI. For this study, a payload has been generated encoding the following string:

   ```
   RadarTest:UniqueId1234, Date:2024-04-16, Time:08:32:06, Msg:Successful transmission test! Extending msg to reach 128 characters.
   ```

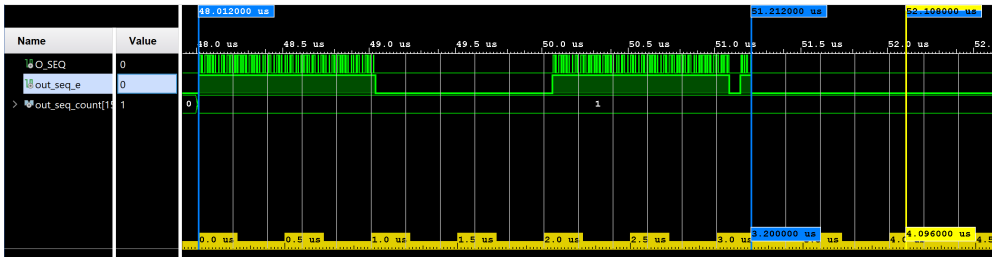
   This is encoded as 1,024bits, divided into 64 chunks of 16 bits each where the first, second, and last chunks are:

   ```
   0101001001100001 0110010001100001 ..... ..... ..... ..... ..... 0111001100101110
   ```

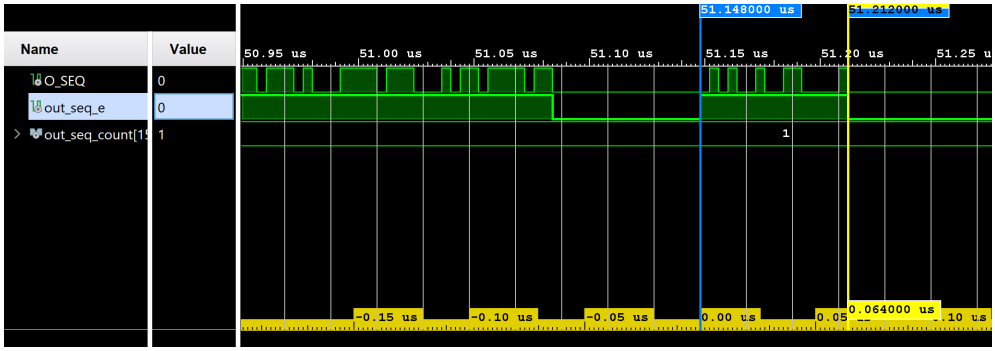
3. Test the new modifications iteratively through the simulation tool provided by Vivado. Figure 3.2 example of what was a confirmation of the correct behavior of the modified "Sequence Generator" block. This is summarized in two observations. One is that now there are 64 pulses in a frame as can be deduced from Figure 3.2a. The latter figure shows the pulses ('`O_SEQ`') in a frame. By counting the pulses or by looking at the value of the counter ('`out_seq_count`') at the last pulse, it is evident that there are 64 transmitted pulses per frame in the updated system behavior. Moreover, in the same figure, the time duration shown in yellow in the bottom right corner (3,027.2μs) aligns with the expected duration of 64 pulses excluding the silence period behind the payload of the final pulse, i.e. 0.896μs. The other important observation is made by looking at Figure 3.2b showing a zoomed-in picture of the first transmitted pulse, the 'b' sequence is followed by a payload. The left and right blue markers mark the beginning of the pulse and the end of the payload respectively. The yellow marker is at the end of the transmitted pulse thus showing a duration with respect to the first blue marker equal to 4,096μs. Zooming furthermore onto the payload of the first pulse, Figure 3.2c confirms that the payload is indeed 16 bits (chips) confirmed by the duration of the payload (0.064μs = 64ns = 16chips$*$4ns/chip). One last check is to check the content of the payload, during testing, all the bits of the payload have been checked for several frames to confirm the appropriate behavior. However, in this context and to keep it short, only the first, second, and last pulse's payloads of a frame will be shown to be compared against the bits shown earlier. Figures 3.2c, 3.2d, and 3.2e serve this purpose and they align with the integrated payload.



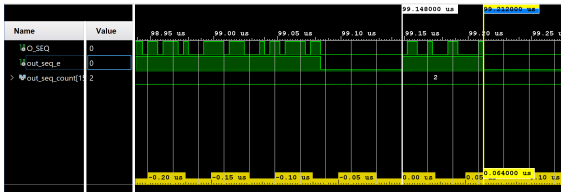
(a) Transmitted Frame



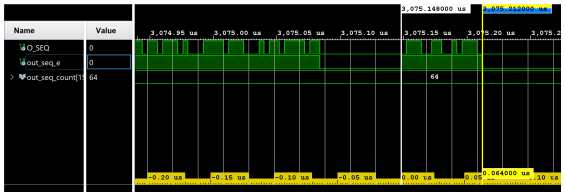
(b) Transmitted Pulse



(c) First Pulse's Payload



(d) Second Pulse's Payload



(e) Last (64) Pulse's Payload

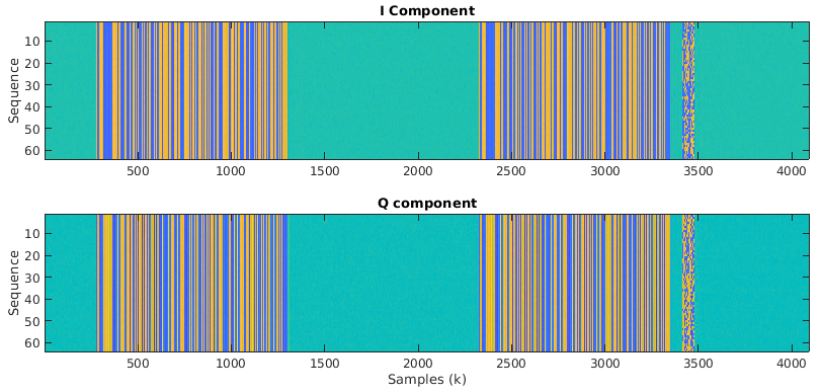
Figure 3.2: Simulated Results after Modifying the Sequence Generator Block

4. Further test the modifications using a single radar unit with an 'in-lab' transmission-reception approach. This test basically means that the radar is solely operating in the **R**anging mode.
   The firmware has been updated to reflect the new changes and some 'in-lab' tests were carried out to verify the correct behavior where an illustration of the setup is shown in Figure 3.3. As at this point of the work, the updated C-client "radar_84_ghz_R", explained in 4.2.1, was not yet built, and due to some bugs that have been solved later, Wireshark has been used to capture and save all the UDP packets arriving at

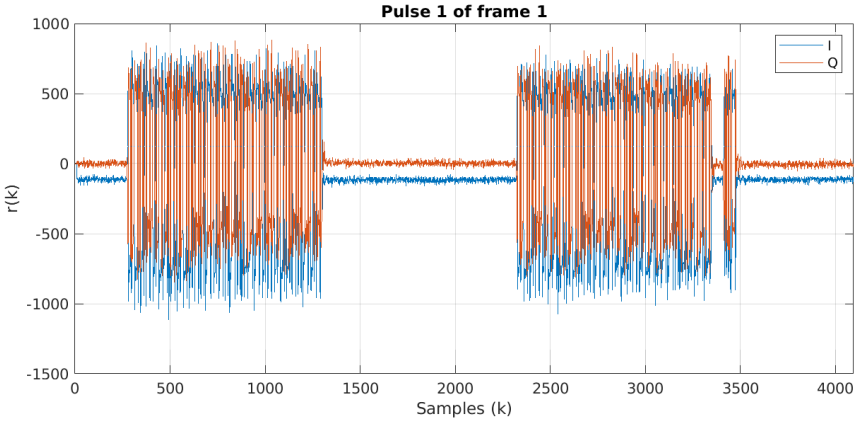
the UDP destination port 4096. Then using the Matlab script "pcap_R" the data has been loaded, processed, displayed, and validated.

Figure 3.3: LAB Setup Illustration Used to Test the Upgraded Sequence Generator Block

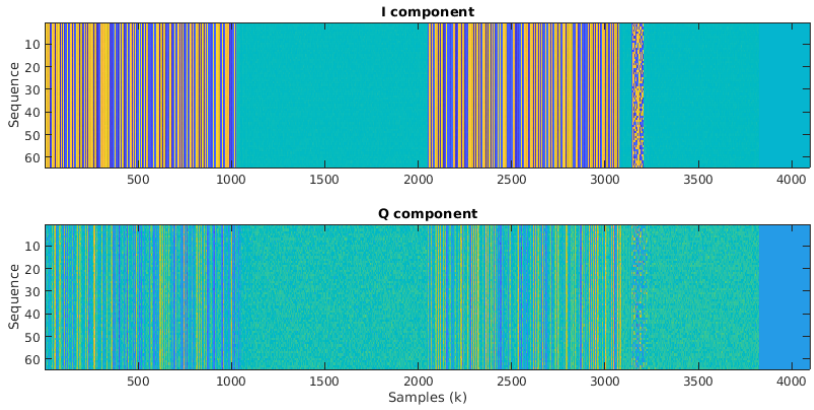
Figure 3.4b shows now a payload found behind the 'b' code and there are 64 received pulses in a frame. This indicates a preliminary positive behavior. However, it is crucial to inspect the payload of each pulse and thus of a whole frame to decide if the payload has been correctly transmitted. For that, the frame's payload shown after the 'b' code in Figure 3.4a, has been extracted, decoded, and compared to the the initial transmitted message. The decoded message is shown in Figure 3.4d. By careful visual comparison, one could confirm the correct transmission of the payload. However, for the sake of double-checking a simple check has been done on the binary vectors (transmitted and received) in Matlab to confirm as shown in the latter figure. **Note** that in order to correctly decode the payload, it was crucial to compensate for the phase rotation the signal underwent. This rotation is visible from the presence of the Q components in high amplitudes as shown in Figure 3.4a, i.e. high amplitude means bright and strong colors on the map of the figure. That's why an additional phase recovery step was added to "pcap_R", this process has been thoroughly explained in **"Phase Recovery and Synchronization"**. Such that the payload has been decoded at the output of this step which is shown in Figure 3.4c. Where it is clear in the latter figure how the Q component became almost negligible and the I component is dominantly the high component.



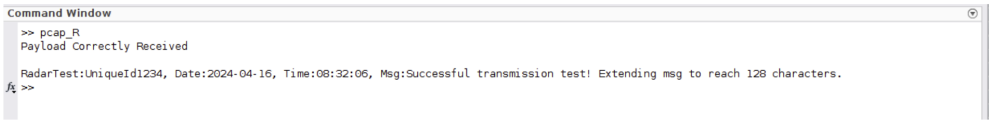
(a) Transmitted Frame



(b) Transmitted Pulse



(c) Transmitted Frame after Phase Recovery



(d) MATLAB Command Window Snapshot Showing the Decoded Payload of a Frame

Figure 3.4: Captured Results after Modifying the Sequence Generator Block

Once those essential steps are done, this concludes the features to be implemented on the transmitter side.

3.2.2 Trigger Block

As can be inferred from what has been presented in 3.1.4, this block is an essential part of the receiver and is to be upgraded. This block has been modified to not only function as it was already doing but also allow for the simultaneous capturing of external signals (what we call communication signals) through a threshold triggering mechanism.

Tests and Workflow Process

In this block as well, a step-by-step approach has been used but in this case, not only the simulation tool has been used in the first stage of tests but also the Integrated Logic Analyzer (ILA) provided as well by Vivado has been utilized in the second stage of tests. Finally, for the third stage of tests, an 'in-lab' transmission-reception approach has been used using two separate radar units, where this setup is shown in Figure 3.5.

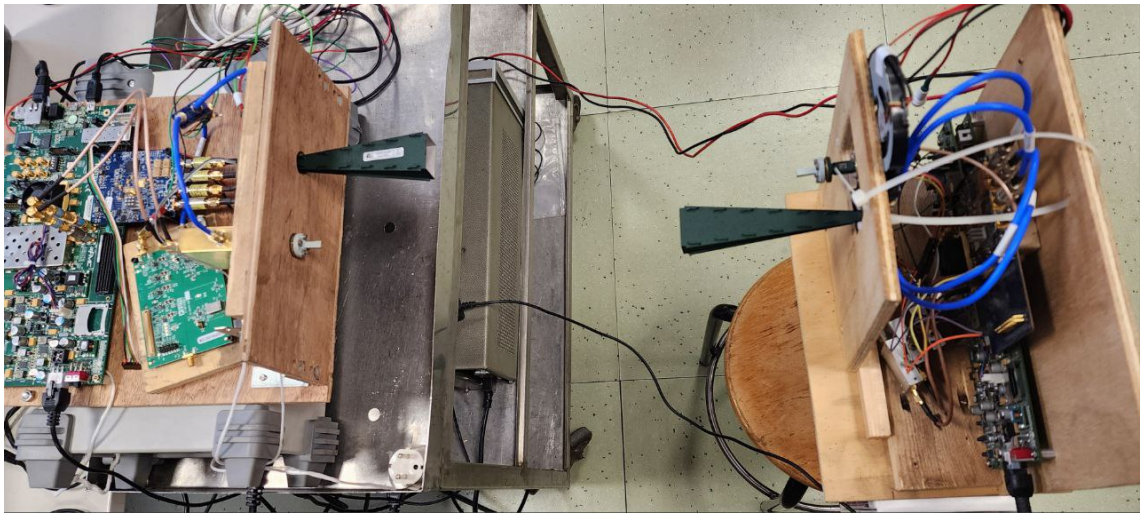


Figure 3.5: Experimental Setup of the two Radar Units (Left Unit is the Rx)

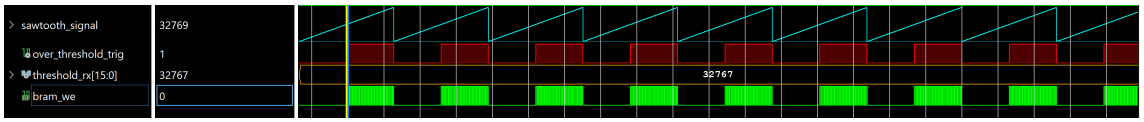
Considering a scenario where the radar (left unit) is only receiving an external signal (from the right unit), so only operating in the joint **C**-mode, i.e. its Tx is off while the other radar unit is transmitting, the main stages and steps taken are:

1. An absolute value process has been built in VHDL code to allow to get the absolute value of the input data.

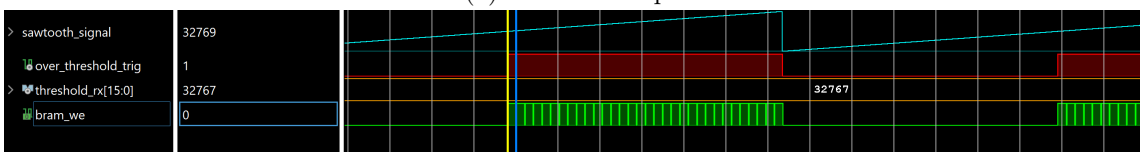
2. A basic data bus comparison process has been built to check whether any of the eight 14-bit samples (4 I and 4 Q samples) is greater than a threshold. The latter is a value saved as a signal in this block. Its default value is $200_{16} = 512_{10}$. However, later, an additional feature has been added to change this threshold in real-time as

per the user needs as discussed in 4.2.3.
The basic idea is that when at least one of the eight samples is greater than the threshold, this block should trigger capturing the joint-**C**ommunication frame by setting '`bram_we`' write enable signal to 1 for 1,024 clock cycles, thus capturing 1,024 chips.

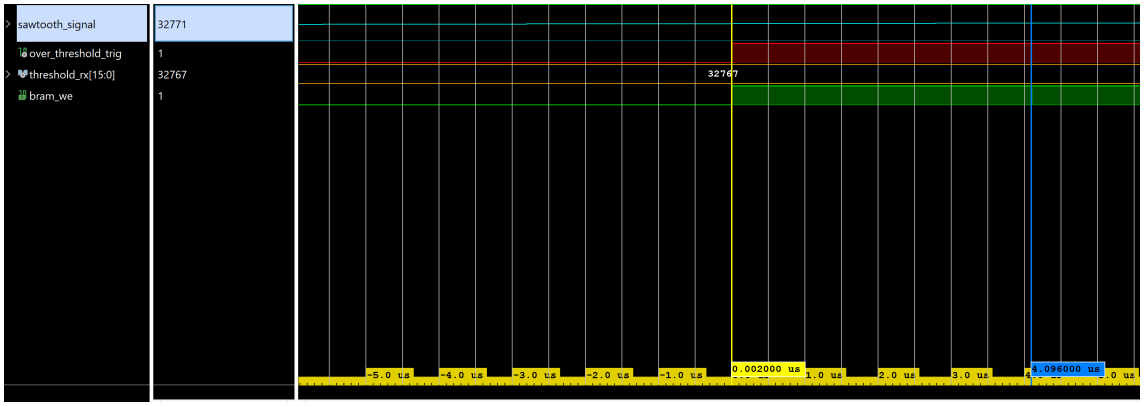
3. To test the progress at this point, a new testing block has been coded in VHDL to generate a positive 'sawtooth', i.e. ramp, shaped signal. Then the simulation tool was used to verify that the triggering mechanism was working as expected. Figure 3.6a shows the simulation output of several ramps. Zooming in to show a single ramp as in Figure 3.6b, it is evident that the trigger is properly functioning as the signal '`over_threshold_trig`' is 1 as long as the ramp's value is greater than the threshold. Moreover, the write-enable signal ('`bram_we`') is 1 when the value of the signal is 32769 which is greater than the set testing threshold (32767). The fact that '`bram_we`' is activated at a signal value of 32769 instead of 32768 as one would expect is due to the clock-synchronous behavior of the FPGA that updates signals at the next rising edge of the clock after the instruction is issued. Moreover, zooming onto a single duration between the rising and the falling of the '`bram_we`' signal as shown in Figure 3.6c, the duration has been noticed to be 4,096 μs (shown by the bottom of the blue marker) which corresponds to 1,024 clock cycles. Thus the '`bram_we`' has been active for 1,024 clock cycles. Then as the ramp signal is continuously increasing until its maximum value we could see how the '`bram_we`' is set once again to 1 after 1 clock cycle and the same behavior repeats.



(a) Several Ramps



(b) Single Ramp (the values shown are those marked by the yellow marker)

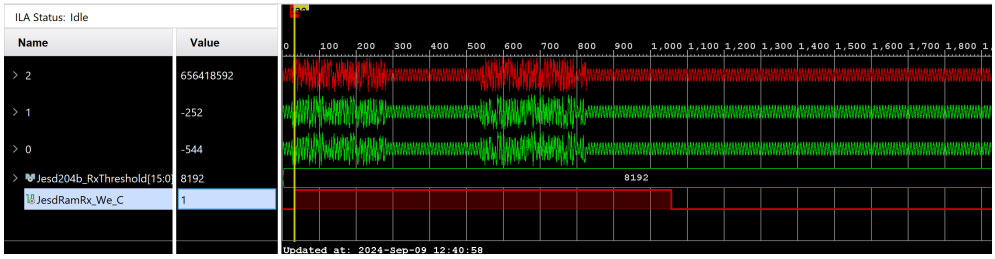


(c) Zoom-in for a Single Active '`bram_we`' Duration

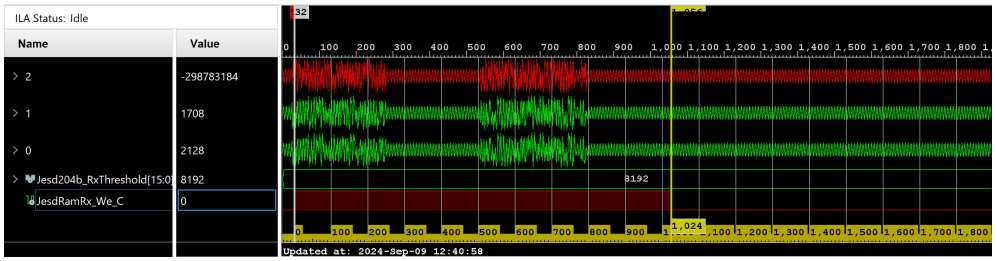
Figure 3.6: Simulation Results Validating the Sawtooth Signal Triggering Mechanism

Once this step has been finalized, the proper triggering functionality has been checked and validated, signaling the readiness to proceed.

4. As said, once the trigger functionalities were tested in simulation on the artificial signal, the firmware behavior was updated to reflect the new VHDL code changes. The ILA tool has been used to further confirm it works before going on the actual data capture from the PHY. The transmitter of the receiver unit is off and the transmitted signal (by the other unit) is again the Golay Complementary Pair with the payload integrated into it. The outputs shown in Figure 3.7 shows the three buses taken from the input data to the trigger block ('`JesdRamRx_Data`') where each bus represents one of the I or Q samples. The threshold value (8192) is also shown, and the output write enable signal ('`JesdRamRx_We_C`'). By observing the behavior shown, it is evident that the write-enable signal is triggered once at least one of the samples is greater than the threshold which is marked in red in Figure 3.7a. Moreover, looking at Figure 3.7b it is once again confirmed that once the trigger is active it remains active for 1024 clock cycles as marked in yellow in the mentioned figure.



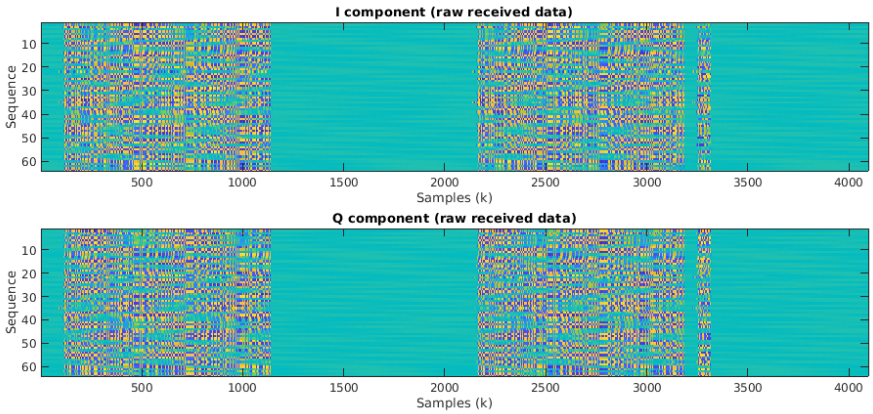
(a) Position of the Trigger



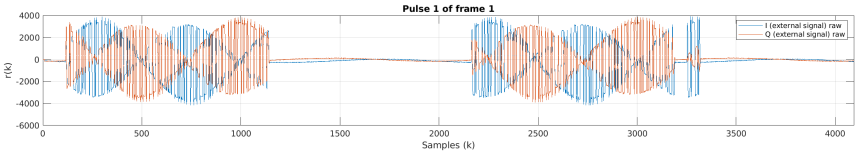
(b) Showing the Duration of Active Trigger

Figure 3.7: ILA Outputs

5. A final stage of testing is to capture on the PC the received data and inspect them. These data have been captured and saved also using Wireshark and the UDP destination port number is still 4096 at this point. A Matlab script "pcap_C", similar to "pcap_R", has been written and used to load and display data. Some results are shown in Figures 3.8a and 3.8b.
   As can be noticed from Figure 3.8a, the received data are not aligned. That is to say, the complementary pair (a,b) are not aligned within a frame between the 64 sequences. In fact, this is expected as the Tx and Rx are not coherent as each one is on a different radar unit. Moreover, looking at Figure 3.8b, it is obvious that the signals are modulating a carrier.



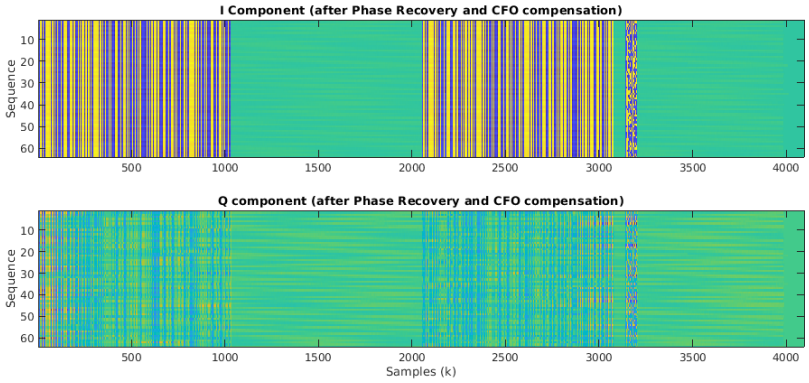
(a) Transmitted Frame



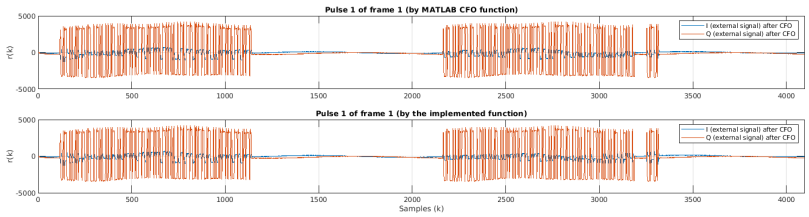
(b) Transmitted Pulse

Figure 3.8: Captured Raw Results after Modifying the Trigger Block

6. Due to the two mentioned observations, it has been crucial at this point to introduce both Carrier Frequency Offset (CFO) compensation and phase synchronization which are elaborated thoroughly in the sections **"Carrier Frequency Offset Compensation"** and **"Phase Recovery and Synchronization"**. For that, new modifications were implemented on "`pcap_C`" to include the necessary algorithms that will enable correctly receiving, demodulating, and inferring the payload. In fact, Figure 3.9 validates the implementation of the mentioned algorithms. Where the former shows the (a,b) complementary pair aligned for a whole frame, while the latter shows the signals not modulating a carrier anymore and compares the implemented algorithm to a function already found in MATLAB from the Communication toolbox.



(a) Transmitted Frame after CFO Compensation and Phase Recovery



(b) Transmitted Pulse after CFO Compensation

Figure 3.9: Captured Processed Results after Modifying the Trigger Block

As a final step in this script, it was crucial to add a part to demodulate and decode the payload, discussed in **"Timing Recovery and Payload Decoding"**, of several received frames to verify the correct reception of data as shown in Figure 3.10.

```
Command Window
  Payload of Frame 1 Correctly Received:
  RadarTest:UniqueId1234, Date:2024-04-16, Time:08:32:06, Msg:Successful transmission test! Extending msg to reach 128 characters.

  Payload of Frame 2 Correctly Received:
  RadarTest:UniqueId1234, Date:2024-04-16, Time:08:32:06, Msg:Successful transmission test! Extending msg to reach 128 characters.

  Payload of Frame 3 Correctly Received:
  RadarTest:UniqueId1234, Date:2024-04-16, Time:08:32:06, Msg:Successful transmission test! Extending msg to reach 128 characters.

  Payload of Frame 4 Correctly Received:
  RadarTest:UniqueId1234, Date:2024-04-16, Time:08:32:06, Msg:Successful transmission test! Extending msg to reach 128 characters.

fx >>
```

Figure 3.10: MATLAB Command Window Snapshot Showing the Decoded Payload of Several Frames

After the previously elaborate 3 stages of testing and verification, it could be confirmed that the standalone '**C**ommunication' receiver is functioning properly.
The crucial next step is to allow the co-existence of these two 'logical' receivers, i.e. the original **R**anging receiver to capture the **R**anging frame and the joint **C**ommunication receiver to capture its corresponding frame in a Time Division scheme.

For that, again, the main highlights of the work are:

1. Two additional output ports are added '`bram_address_C`' and '`bram_we_C`' serving the same purpose as '`bram_address`' and '`bram_we`' discussed in 3.1.4, where the latter two are now renamed '`bram_address_R`' and '`bram_we_R`'. Obviously, the last letter of the output ports corresponds to which mode they are used in, i.e. **R**anging or **C**ommunication.

2. This block has been programmed to always prioritize in time domain the capture of the **R**anging frame. For that the write enable signal of this mode represented by '`bram_we_R`' has been used as a blind signal to temporarily 'blind' the communication receiver.

Figure 3.11, shows a logical illustration of how these two logical receivers co-exist highlighting the crucial communicated control signals. The blind signal force stops the integrated **C**-trigger from functioning.

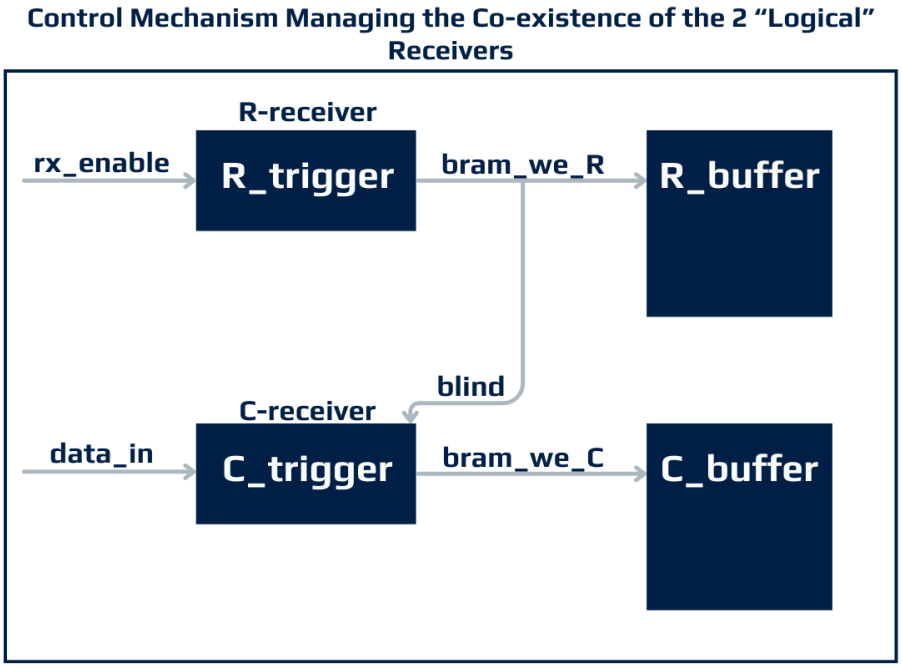


Figure 3.11: Logical Interpretation of the Control Flow behind the updated Trigger Block

The ideal case which is indeed dominantly the most probable one, is that different mode frames, **R**-frame and **C**-frame, never arrive at the same time, this will cause no delay to any of the received frames. The dominance is due to the low transmission duration and extremely high silence ones. To elaborate, in a frame, the radar transmits and expects its echo for 0.3072 (ms) while the silence period between frames is $100 - 0.3072 = 99.6928$ (ms). Thus the % ratio between the two periods is $\frac{0.3072}{99.6928} \times 100 \approx 0.3\%$. Considering this extremely low probability, we distinguish two additional cases where frames from different modes arrive within the same timeline:

1. **A C-pulse arrives while the R-receiver is on**, i.e. ranging data is already being received while some other radar transmits to this radar unit. This will cause signal corruption but will not trigger the '**C**-receiver'.
2. **An R-pulse arrives while the C-receiver is on**, i.e. external communication data is being received and stored when the same radar unit receives its echoes. This again will cause signal corruption and will trigger the '**R**-receiver' to capture data and store them in the corresponding FIFO but will delay the packet transfer to the PHY until the first availability of the ethernet link.

Note that, even though the previous two cases could happen, they still have an extremely low probability of occurring.
The proper final functionality and behavior of the updated Trigger block as well as other blocks has been elaborated in Chapter 5.

3.2.3 ADC FIFO

This block has been modified to better utilize the memory, i.e. higher efficiency. The need for higher utilization efficiency rose since the final system requires two different buffers of the same size one for the **R**-frames (**R**-buffer) and one for the joint-**C**-frame (**C**-buffer). It is important to note that the write-enable signal of the **R**-buffer is still the same as the original with a new naming convention while that of the **C**-buffer is the signal at the output of the new output port of the trigger block '`bram_we_C`'. However, the functionalities of the **R**-buffer are intact and replicated for the **C**-buffer.
Due to the existence of two large-size buffers, the BRAM usage (92%) exceeds the one recommended by Vivado (80%) which in turn makes it harder for the FPGA to do the "timing-closure" successfully which puts the system in the risk of having some failing routes and unexpected behavior. Thus some optimizations were done to ensure the system will not fail at any stage. One of these optimizations is exploiting the fact that the ADC is of 14-bit precision, thus the 8 samples per clock cycle (4 I samples and 4 Q samples) need exactly 112 bits (14bit/sample). The shape of the buffer has been altered to be:

- Write Depth: 65536 (not changed)
- Write Width: 112 bits (was 128)
- Read Depth: 112 bits (was 128)

These changes forced consequent changes in the shape of handled inner data leaving the terminal data input/output of the system unaltered. Several alterations were made to ensure that the final behavior aligns with the original one looking at the overall system input/output.

3.2.4 MAC

This module has been modified to implement new logic that accommodates other firmware changes and facilitates the transport of received communication data.
Key additions include:

- **Ports:** An additional input port '`send_packet_C`' has been added, functioning similarly to '`send_packet`' mentioned in the starting state description of this module where the latter has been renamed to '`send_packet_R`'. As the notation suggests, the port ending with '`R`' carries the signal responsible for initiating the packet transmission carrying the Ranging data, while the one ending with '`C`' does the same for communication data. Following the same analogy, '`data_in_re_R`' and '`data_in_re_C`' output ports were defined which carry the read enable signals fed to the **R**-buffer and **C**-buffer. Finally, one output port, '`in_port`', has been added. It marks which buffer the module chose to read from, and sets the buffer's data bus to read from as well as the UDP destination port number.

- The core modifications/upgrades were done to the MAC top module and the sub-block **"byte_data"**. These include adding/modifying signals or ports and implementing various logic operations and commands.

- **Functionality:** The final behavior of this module is summarized by the 'logical' block diagram shown in Figure 3.12.

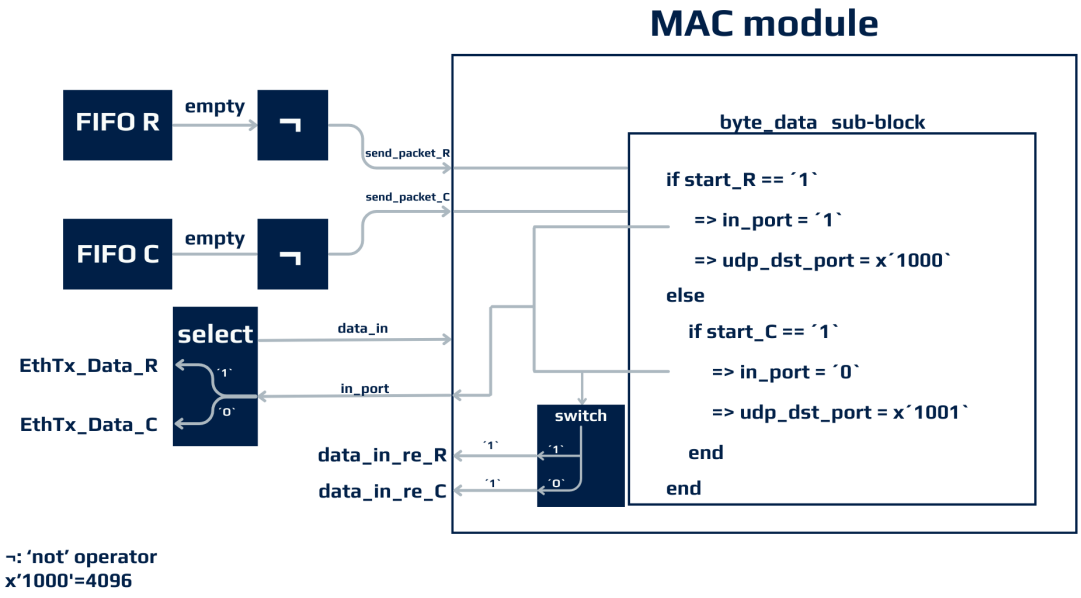


Figure 3.12: Logical Interpretation of the Control Flow behind the updated MAC Block

In the updated functionality, the MAC module interfaces with the two FIFO buffers, **R**-buffer and **C**-buffer, to get the '`empty`' signal from each. Primarily, if the **R**-buffer is not empty, the MAC module starts reading from it to form and send UDP packets. Otherwise, if it is empty, it checks the **C**-buffer to perform the same operation. Depending on which buffer is being read from, an internal signal ('`in_port`'), also mapped also the output port with the same name, is set to '1' or '0' and thus the UDP destination port is set to 4096 or 4097 when reading from the **R**-buffer or from

the **C**-buffer, respectively.
Given that the maximum writing speed into the system is proportional to the sampling rate (1Gsps, resulting in a writing bit rate of 28Gbps with a 14-bit ADC) and the maximum output rate is limited by the Ethernet connection speed (1Gbps), it is crucial to start reading as soon as either buffer is not empty. However, reading from the **R**-buffer is always prioritized, as ensured by the logical flow implemented in the "byte_data" sub-block, as shown in Figure 3.12.

3.2.5 RAM-based Shift Register

For the same reason the shape of the FIFO buffers was altered, the depth of this block has been changed to 32 (was 64). This change helped in keeping the usage of BRAM as minimal as possible while still getting enough delay for correct data reception.

Chapter 4

Software Description

4.1 Starting State

The starting status of the PMCW automotive radar software system was composed of three main components: the C-client application, two Matlab scripts, and the FPGA uBlaze code. Each of these components played a crucial role in the operation of the radar system.

4.1.1 C-Client Application

The client software was written in C and consisted of two applications: "**radar_84_ghz**" and "**png_draw**".
An example output of this client is shown in Figure 4.1a.

radar_84_ghz

This application is a complex piece of software. It's designed to receive and process data from an FPGA board through an ethernet connection on a preset UDP port with address 4096 and can save this data onto a disk, depending on the command-line parameters provided. Here's a detailed breakdown of its main functionalities:

1. **Header Files**: The application includes several standard C and system-specific header files, which provide the necessary libraries and functions for network communication, mathematical operations, and image processing.

2. **writeImage Function**: This function writes out the Portable Network Graphics (PNG) image file.

3. **Constants and Arrays**: The application defines several constants and arrays. For example, '`DELAY`' and '`N_PULSE`' are defined as constants (512 and 65 respectively), representing the number of delay bins and the number of pulses per frame respectively. Arrays like '`p_debug_map`' and '`p_debug_map2`' are used to store the Doppler Map and Correlation Matrix Map respectively. The '`pulse_data_i`' and

'`pulse_data_q`' arrays are used to store the 'I' and 'Q' data from the sampled received ranging signal.

4. **Main Function Arguments**: The main function of the program takes two command-line arguments: the starting delay to be used in the correlation calculation and a mask to determine whether to save data and which data to save.

5. **Socket Programming**: Socket programming is a method of communication between two computers using a network protocol. In this application, UDP is used. The application creates a socket, which is an endpoint for receiving data across a computer network. It sets the socket options to optimize the performance, such as increasing the receive buffer size to 4MB and disabling the checksum for the socket. The socket is then bound to a specific IP address and port number. In this case, it binds to any available network interface and port number 4096. This means the socket will receive data sent to this port number on any network interface of the machine.

6. **Fast Fourier Transform** : The application uses the Keep It Simple, Stupid (KISS) FFT library to perform Fast Fourier Transforms. This is used to convert the time-domain radar signals into the frequency domain for further processing.

7. **Data Processing**: The application processes the received data as has been introduced in the Introduction in 1.4, performing operations such as correlation calculation according to the range response as established in equation 1.8 and consequently calculate the FFT along the columns to generate and save both the Cross Correlation and the range-Doppler map images in real-time with the `writeImage` function.

png_draw

This application continuously shows a PNG image which could be the range-Doppler image and/or the absolute value of the correlation both generated and updated in real-time by the processing application (**"radar_84_ghz"**). The default width and height of the images shown are 512 and 64 respectively. In fact, this application serves as an image-viewing tool and is responsible for what is presented in Figure 4.1a.

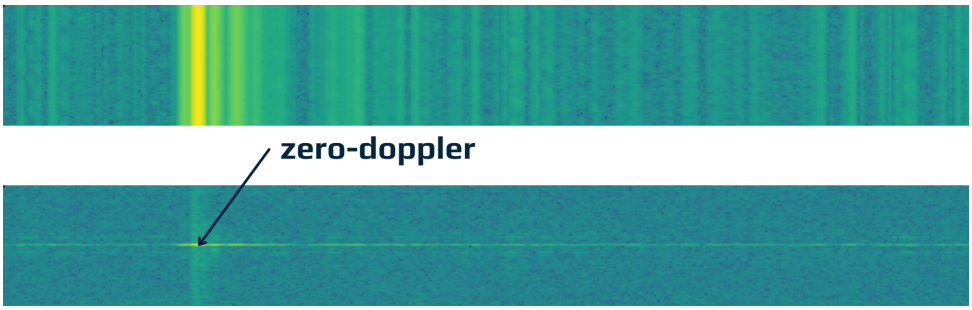
4.1.2 Matlab Scripts

There were two scripts named "**compare_data_ab.m**" and "**get_code_ab.m**".

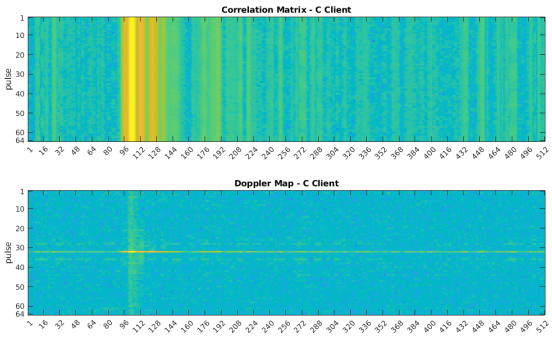
- **compare_data_ab.m** loads the I/Q raw data and the images previously saved to disk from the C application, performs the same processing, and finally, compares the results to those of the data processed by the C application.
  This script is mostly useful for showing the different phases of processing and for debugging purposes.

- **get_code_ab.m** is the script that generates the Golay complementary codes according to the mechanism elaborated in the introduction.

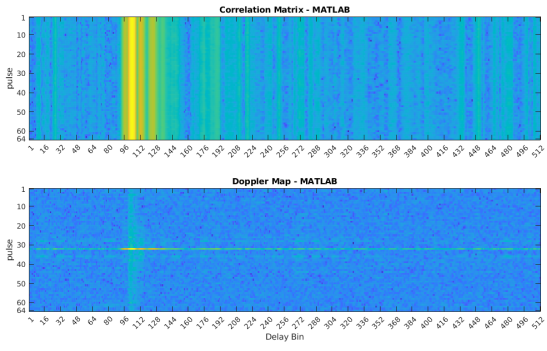
Again, some example outputs are shown in Figures 4.1b and 4.1c.



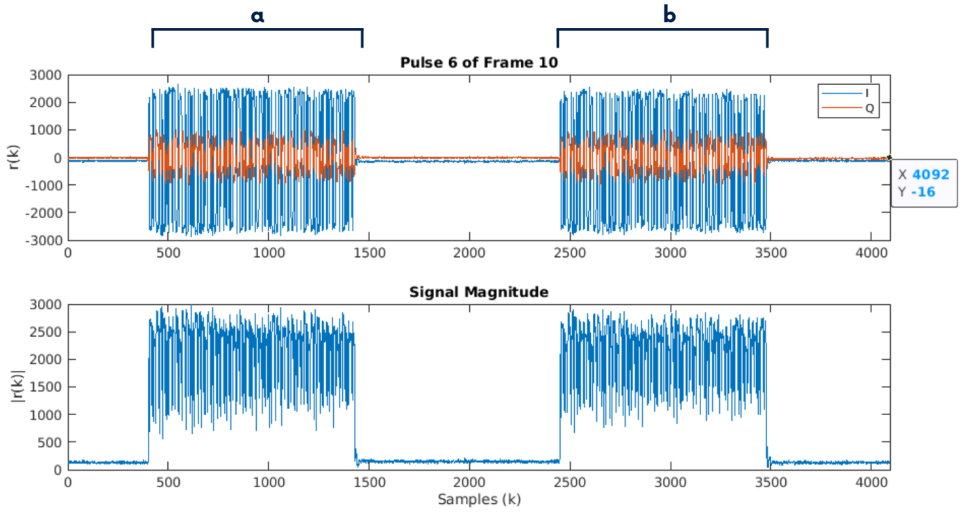
(a) Correlation Matrix (above) and Doppler Map (below) as Shown by the C-client



(b) C-client-Calculated Outputs Loaded by Matlab



(c) Calculated by Matlab



(d) Typical Reflected Ranging Signal

Figure 4.1: Original Behavior Example Outputs

The two plots in Figure 4.1a are generated and updated in real-time by the C-client running on the Linux PC. In contrast, Figures 4.1b and 4.1c are produced during the post-processing steps using MATLAB. Despite the system initially transmitting 65 pulses

per frame, the maps for a single frame only display $64 = 2^6$ pulses. This discrepancy arises because the FFT is applied column-wise, and computations are faster when the lengths of the FFTs are powers of 2. This efficiency is attributed to the computational complexity of the FFT algorithm, which is $O(n \log_2 n)$ instead of $O(n^2)$ when the length (n) is a power of 2 [8]. Lastly, Figure 4.1d presents the received ranging pulse at the system's starting point. It's worth noting that the number of samples is 4,092, not 4,096 as one might expect due to the data protocols explained in the introduction. This is because the first 4 samples of each pulse, which are reserved for metadata, are removed prior to plotting to enhance the visualization.

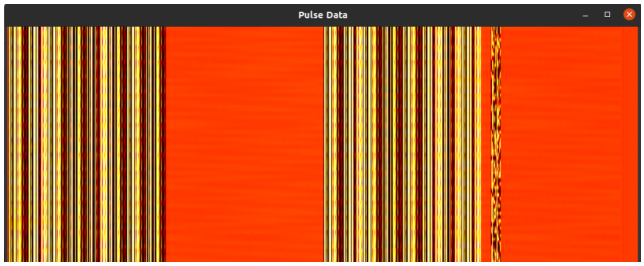
4.1.3 FPGA uBlaze Code

The uBlaze software was written in C "bare metal", i.e., no Operating System (OS) used. The software initialized the board's peripherals, then waited for commands on Universal Asynchronous Receiver / Transmitter (UART) or Ethernet and handled the commands with a switch/case statement. It includes an important script which is **"set_dac"** as well as several other interfaces with different blocks of the system through the AXI interface. The latter is like an application that sends a command via a UDP packet to the FPGA board, configuring the DAC output values of the AMS101 board to compensate for the LO leakage (LO nulling).

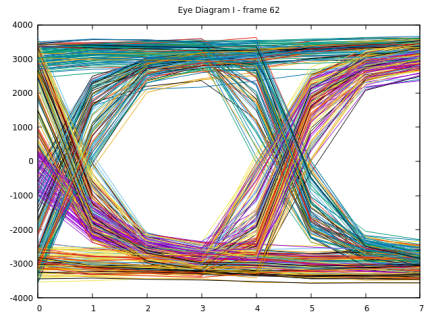
4.2 Upgraded State

4.2.1 C-Client Application

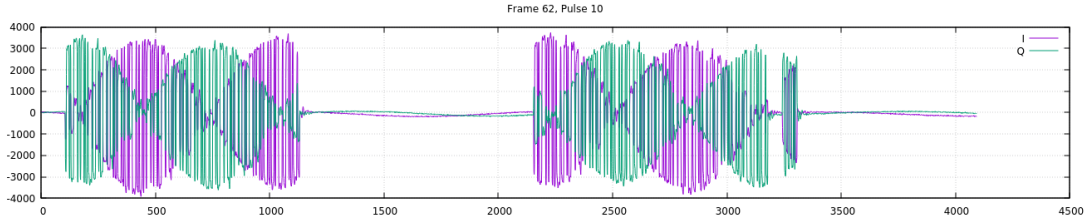
For the final model, two different C-clients were used. Namely, "**radar_84_ghz_R**" and "**radar_84_ghz_C**" for the ranging frame data and the communication frame data respectively. The former is a modified version of "radar_84_ghz" while the latter is a new client written from scratch. These two clients could be run in parallel where the main distinction in the flow of the two streams of data is in the UDP destination port addresses, they are 4096 and 4097 respectively.
In addition to those, several C-applications have been written to visualize different aspects of the received communication signal. Those include the I/Q signals, the overall received frames, and the eye diagram of the frames' payload samples (frame-by-frame). Examples of such output are shown in the subplots of Figure 4.2.



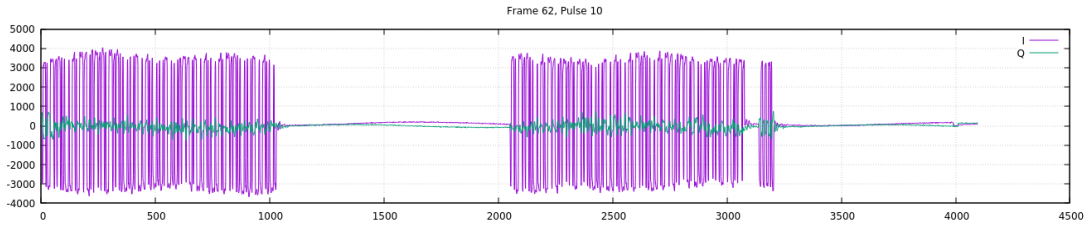
(a) Frame Map Plot



(b) Frame's Payload Eye Diagram



(c) Real-Time Signal Plot Before any Processing



(d) Real-Time Signal Plot after Processing

Figure 4.2: New C-Client Applications Example Outputs

Final Behaviour

- "**radar_84_ghz_R**" is generally the same as "**radar_84_ghz**" explained in the starting point section of this discussion.
  However, the number of pulses or sequences per frame has been set to 64 instead of 65 as it was previously. That is to reflect on the changes done in the FPGA code. Moreover, now the frame condition is not only based on the sub-pulse number anymore but also on the timestamps where it is anticipated that the timestamp difference between two successive Ranging frames has to be in the order of 10^7, more precisely around 23844000 where this timestamp is just a conventional number that increases at the Tx according to the source clock.

- "**radar_84_ghz_C**" has been built from scratch. It is responsible for receiving the data stream or flow arriving at the UDP port with address 4097, then processing this raw data which includes CFO compensation, phase synchronization, and finally payload demodulation and decoding. It also compares the decoded payload

to the already known transmitted payload. Consequently, counters tracking both the number of correctly received frames and mis-received frames are shown on the terminal.

Behaviour Workflow

Some crucial backbone algorithms allow the successful reception and visualization of the communication data.
The overall behavior regarding both **CFO** and then **Phase Recovery** is summarized in the flow chart of Figure 4.3.

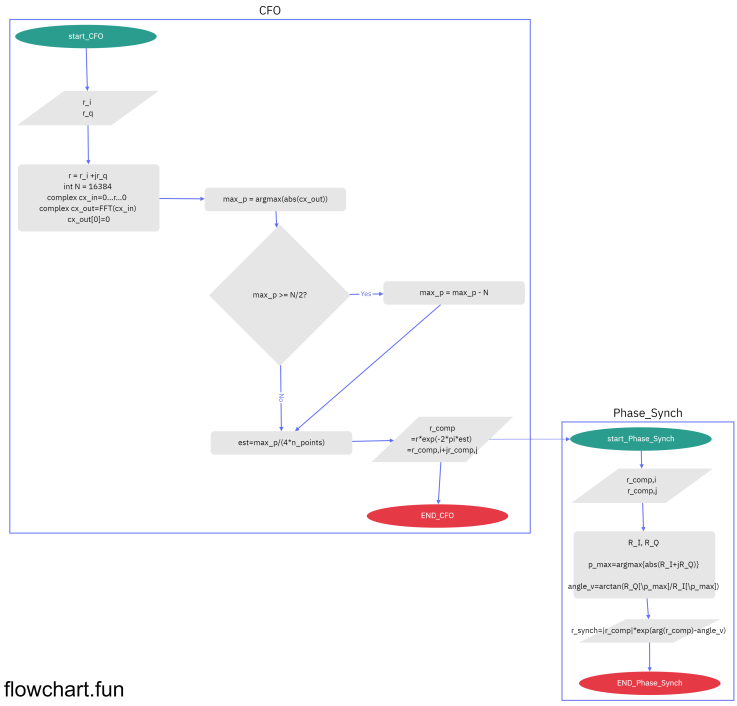


Figure 4.3: Flowchart Summarizing the Implemented CFO Compensation and Phase Recovery Algorithms

Carrier Frequency Offset Compensation A non-data-aided feedforward CFO estimator has been built based on an FFT approach. The principles behind this estimator have been summarized in [7] and extensively discussed in [10].
For what concerns this study, the estimator is given by:

$$\Delta\hat{f} = \frac{f_s}{N \cdot m} \arg\max_f \left| \sum_{k=0}^{N-1} r^m(k) e^{-j2\pi kt/N} \right|, \quad \left(-\frac{R_{\text{sym}}}{2} \leq f \leq \frac{R_{\text{sym}}}{2} \right) \tag{4.1}$$

where

$$N = 2^{\lceil \log_2\left(\frac{f_s}{f_{\text{res}}}\right) \rceil} \text{is the number of samples, i.e. FFT length} \tag{4.2}$$

Here, $m = 4$ is the order of the signal, $f_s = 10^9$ samples per second (sps) is the sampling rate, $f_{\text{res}} = 10^5$ is the desired frequency resolution which sets the FFT length used, f is the we search for ranging between $-\frac{R_{\text{sym}}}{2}$ and $\frac{R_{\text{sym}}}{2}$, and $R_{\text{sym}} = \frac{f_s}{4} = 250 MegaBaud$(MBd) is the symbol rate.
Basically, what is implemented is:

1. Construct the complex received discrete (sampled) signal as:

$$r(k) = r_I(k) + j \cdot r_Q(k) \tag{4.3}$$

   where r_I and r_Q are the I and Q received signal respectively.

2. Construct the input complex vector, cx_{in}, of size N to undergo the FFT by taking the fourth order power of the 4096 complex samples of r and zero-padding it. This structure is illustrated in Figure 4.4.

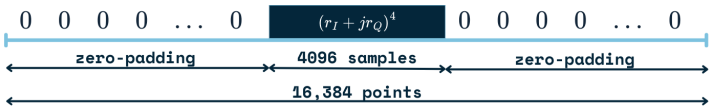


Figure 4.4: FFT Input Data Structure

3. Neglect the DC component, i.e. first element, and find the position of the element with a maximum magnitude of 'cx_{out}', the complex FFT output, denoted 'max_p' in the flow chart.

4. Adjust 'max_p' according to its position to distinguish whether it is in the negative section of the FFT (as shown in Figure 4.5) or the positive section.

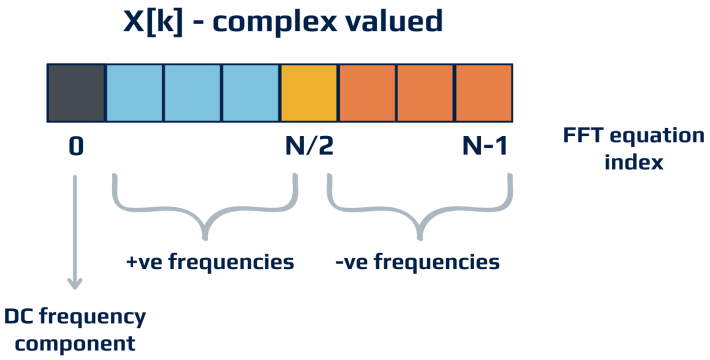


Figure 4.5: FFT Output Structure

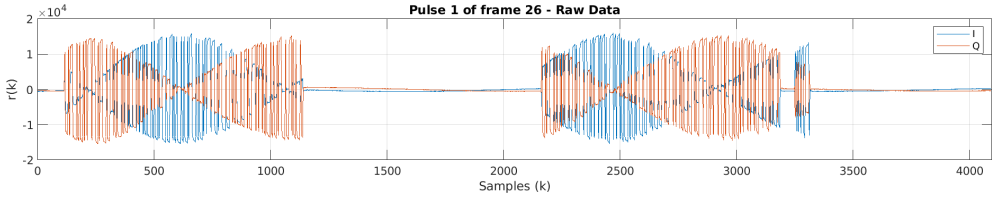
5. Calculate the frequency offset estimation (est $= \Delta\hat{f}$) as est $= \frac{\text{max\_p}}{N \times m} = \frac{\text{max\_p}}{16{,}384 \times 4}$.

This process is performed only for the first received pulse of each frame for computational efficiency. The calculated value of 'est' is then used for all the 64 sampled pulses of the

frame to compensate for the carrier offset following the formula:

$$r_{comp}(k) = r(k) \times exp(-2\pi \cdot \text{est} \cdot k) \tag{4.4}$$

It is important to highlight that Figures 4.6a and 4.6b provide illustrative examples of the received raw pulse, prior to any processing, and the pulses after the CFO compensation performed by the "radar_84_ghz_C" client in real-time, respectively. Another significant point to note is that these data sets were loaded and visualized using Matlab without undergoing any additional processing.



(a) Pulse - Raw Data

(b) Pulse after CFO Compensation

Figure 4.6: Received Communication Data

It is evident from Figure 4.6b that the signal is not modulating a carrier anymore however it is still phase rotated as both I and Q components are present in significant levels.

Phase Recovery and Synchronization The input to this step should be the signal obtained at the output of the CFO compensation, denoted as r_{comp}, i.e. signal shown in Figure 4.6b. The algorithm for the phase recovery is based on a correlation approach exploiting the fact that the transmitted signal is known a-priori. This signal is the Golay complementary pair as shown in Figures 1.8 and 1.12a. The first step of this correlation approach is similar to what has been presented in equation 1.8.
In this case, $a(t)$ is the code, and $b(t)$ is its complement. The first 1,024 samples of r_{comp} are denoted by a_{rx}, and the third set of 1,024 samples (i.e., samples 2,048 to 3,071 using zero-based indexing) are denoted by b_{rx}.
The correlation calculation is performed once separately for the I and Q components of r_{comp}, where

$$r_{comp}(k) = r_{comp,I}(k) + j \cdot r_{comp,Q}(k) \tag{4.5}$$

The steps are as follows:

1. Compute the 2 correlation functions R_I and R_Q using $r_{comp,I}(k)$ and $r_{comp,Q}(k)$ respectively.

2. Construct the complex correlation function as:

$$R = R\_I + j \cdot R\_Q \tag{4.6}$$

3. Identify the position of the maximum magnitude element of R, denoted as p_max.

4. Compensate, according to equation 4.7, based on the argument of the complex number in R with index p_max, denoted as 'angle_v' (angle_v = arg (R[p_max])) in the flowchart shown in Figure 4.3.

This process should be repeated for every received pulse to allow the correct payload demodulation from the final synchronized signal given as:

$$\begin{aligned} r_{synch}(k) &= r_{synch,I}(k) + j \cdot r_{synch,Q}(k) \\ &= |r_{comp}(k)| \times exp(arg(r_{comp}(k)) - \text{angle\_v}) \end{aligned} \tag{4.7}$$

Figures 4.6b and 4.7 illustrate a pulse before and after this phase synchronization step. Importantly, these data sets were loaded and visualized using Matlab without the need for any additional processing as well. It's worth noting that the phase synchronization process, performed in real-time by the "radar_84_ghz_C" client, significantly alters the state of the pulses. This is evident from the fact that the Q component of the signal, i.e. $r_{synch,Q}(k)$, aligns closely with the noise level. Consequently, nearly all the signal power is concentrated along the I component, the high component.

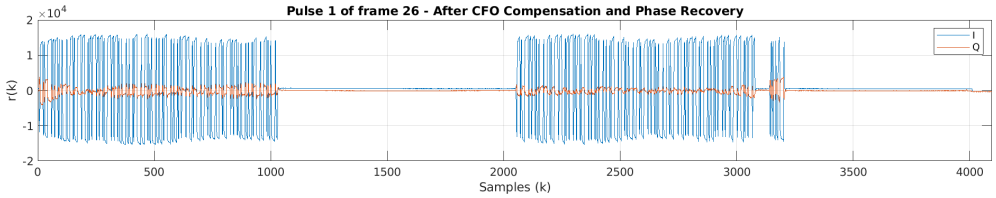
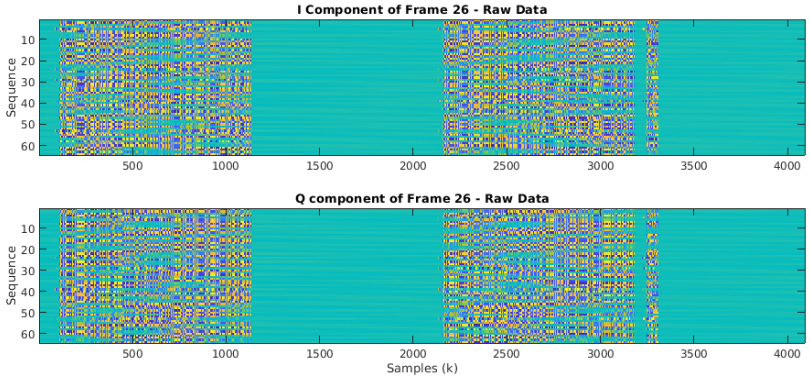
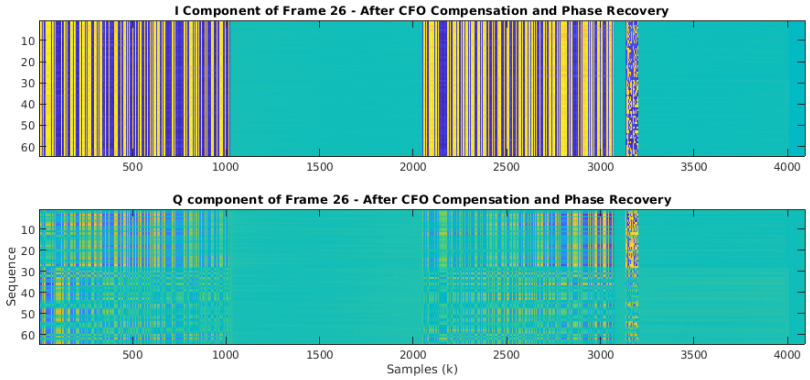


Figure 4.7: Received communication Pulse after CFO Compensation and Phase Recovery

Furthermore, Figures 4.8a and 4.8b display the 64 pulses of a frame. It becomes apparent that the sequences align well only after the phase synchronization process. This alignment is expected only on the ranging part of the signal as the a and b sequences of each of the 64 pulses are identical, leading to their alignment as depicted in Figure 4.8b. Indeed, it's important to note that while the a and b sequences of each pulse align after phase synchronization, the payload part of each pulse is not expected to align. This is because each pulse typically carries a unique payload. Therefore, variations in the alignment of the payload part across different pulses are normal and expected. This distinction is crucial for the correct interpretation of the data.



(a) Raw Data



(b) Data after CFO Compensation and Phase Recovery

Figure 4.8: Captured Communication Data Organized in Frames (64 pulses)

Timing Recovery and Payload Decoding In this stage, the payload samples are extracted from each pulse on the I component (High component), which was previously denoted as $r_{synch,I}(k)$. For a frame, there are 64 payload samples per pulse, representing 16 chips. This results in a total of 4096 payload samples, which represent the 1024 payload chips. Given that the payload is fixed and its position is well known as illustrated in Figure 1.12a, this extraction process is straightforward.
The modulation format used is BPSK. A threshold approach is applied to every second sample to downsample the payload samples. This is based on the assumption that each second sample is the strongest one. If a sample is greater than zero, it is mapped to a bit equal to 1, otherwise, it is mapped to 0.
Finally, the 1024 bits of payload are decoded to retrieve the original transmitted string. Each character in the string is represented by 8 bits. This process ensures that the original data is accurately and efficiently recovered from the received signal.
In a real-time scenario, the C-client, **"radar_84_ghz_C"**, is designed to continuously compare the decoded payload with the known transmitted payload for each frame. This

comparison process is crucial for assessing the accuracy and reliability of the data transmission and recovery processes.
Two counters are displayed on the terminal allowing the user to clearly and immediately understand the system's performance. One counter tracks the number of frames received correctly, while the other counter keeps track of those that were not.

Data Visualization As briefly aforementioned, data visualization functionalities have been seamlessly integrated into the system. This integration facilitates real-time, efficient visualization of the data. The user can visualize various aspects of the data. This includes the raw I and Q received sampled signals $r_I(k)$ and $r_Q(k)$, as depicted in Figure 4.2c. Users can also view the final synchronized signals $r_{synch,I}(k)$ and $r_{synch,Q}(k)$, as shown in Figure 4.2d. Additionally, the entire received frame can be visualized as shown in Figure 4.2a, along with the eye diagram corresponding to a frame's payload samples (4096 samples), as in Figure 4.2b. The user can choose what to plot by setting the appropriate flags.
To enable these functionalities, several libraries, including Gnuplot and OpenCV, were utilized.

4.2.2 Matlab Scripts

The scripts mentioned in the description of the starting state of the software of this system have been adjusted to cope with the newly implemented features or modifications while keeping the same general functionality.
Other than the scripts **"pcap_R"** and **"pcap_C"** which were devised during the firmware upgrade step of the system and were not used later, a new script has been written "**read_fromC**" which loads the data saved by the C-client application and displays them by setting the directory of the saved data and some flags inside. Additionally, it could do the whole processing all over again to compare the performance between the output of the C-client and its calculated results. This latter feature came very handy in debugging steps while later after everything has been set up it is not as much needed that's why it could be turned on or off with a flag set by the user.

4.2.3 FPGA uBlaze Code

Several modifications/additions were applied to this code to cope with the updated firmware behavior. The number of sequences per frame has been modified to be 64 instead of 65 on the AXI bus programming the corresponding FPGA peripheral. The default threshold has been set to be $200_{16} = 512$. The Tx of the prototype system has been set to be off by default. Additionally, three user-interfacing commands have been added to allow more flexibility in the user interaction with the system. These commands are communicated through the UART port interface and they are:

1. **v**: This command turns the Tx on which displays a message confirming that the Tx is on as well as the number of sequences transmitted per frame.

2. **t**: This command turns the Tx off. It also displays a message confirming that the Tx is off.

3. **o0xt**: This command sets the threshold of the **C**-Receiver, where 't' is the threshold value in hexadecimal. This command is crucial to allow the user to modify the threshold in real-time to cater the specific needs.

Chapter 5

Tests and Validation

After upgrading the various components of the system and iteratively checking that each newly implemented feature functions properly. It is essential to test the overall system performance in an 'in-lab' scenario as close as possible to its intended real-world application with a setup similar to that shown in Figure 3.5. The primary focus of this chapter is to validate the integrated communication capabilities of the upgraded system, rather than examining each smaller upgrade, which has been detailed in the previous chapters.

5.1 Validation Criteria

The validation of the system's functionality relies on three key metrics, which must be observed simultaneously:

1. **The proper radar functionality.** This is verified by ensuring that ranging data is received correctly, without errors on the C-client application "**radar_84_ghz_R**", and that an informative range-Doppler map is generated. An informative map indicates that the radar can detect target position and speed in real-time, accurately reflecting this on the range-Doppler map.

2. **Correct demodulation of the received payload.** This is confirmed by observing the counters displayed on the C-client app "**radar_84_ghz_C**". As previously mentioned, two displayed counters are used to track the number of correctly and incorrectly received frames. The application compares the demodulated payload with the a-priori known transmitted payload to verify correctness.

3. **A wide-open eye diagram for the payload samples** is a critical indicator of communication system performance. The wider the eye opening, the more reliable and robust the signal, directly reflecting the system's ability to handle data transmission with minimal errors. This metric is crucial, as a completely closed eye diagram makes correct payload demodulation nearly impossible. Even when demodulation

appears successful, the eye diagram provides deeper insight into the overall signal integrity and system efficiency, making it an essential measure of performance beyond just error rates.

5.2 Tests

Both the Ranging and Communication process modes should operate simultaneously and correctly. To ensure clarity, the test results are presented in two distinct sections, though it is important to emphasize that **these tests were performed concurrently**.

5.2.1 Ranging Process Mode

The radar's ranging capabilities were tested by moving a corner reflector at several known distances and analyzing the range-Doppler map, as well as a zero Doppler cut (2D range profile). These tests were conducted to verify that the radar can accurately detect targets at different ranges and display the results in real-time.

- The radar system was placed in a lab environment with a moving target.
- The C-client application "**radar_84_ghz_R**" was used to visualize the range-Doppler map and monitor the detection of the target.
- Figure 5.1 shows the range-Doppler map showing a target moving from 2 to around 4meter (m) away from the radar, where the target is clearly detected. The Doppler axis also indicates the static nature of the target.
- In addition to the range-Doppler map, a zero-Doppler cut (2D Range Profile) of the data was performed, as shown in Figure 5.2, illustrating the range profile of the target after it briefly moved.

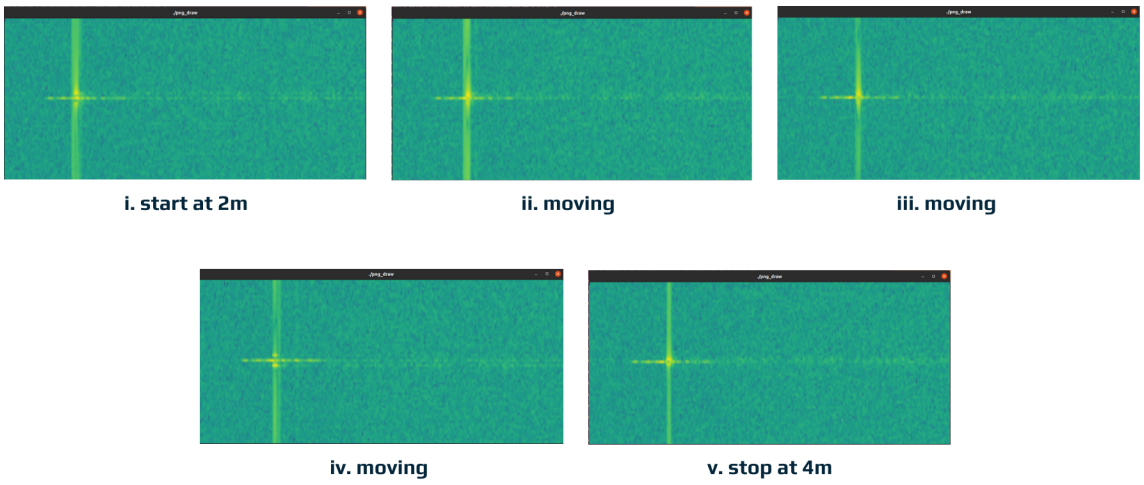


Figure 5.1: Real-time Range-Doppler map showing successful detection of a moving target

The clear representation of the target's range and static nature with respect to the zero Doppler axis indicates correct radar functionality.

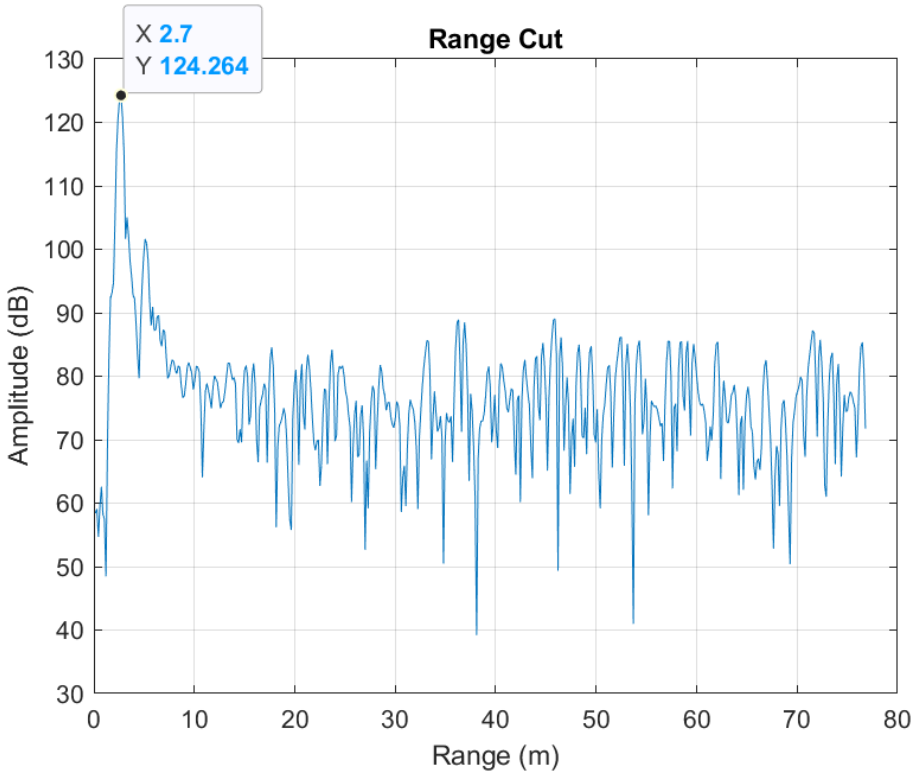


Figure 5.2: Zero-Doppler cut showing the detection of a corner reflector at around 3 meters

The strong peak indicates a clear reflection and confirms accurate distance measurement.

5.2.2 Communication Process Mode

In parallel to the radar ranging tests, the communication mode was tested to evaluate how distance affects signal integrity and payload demodulation. Tests were conducted at varying distances (1m, 2m, 3m, 4m) to observe how the system performs as the signal weakens over distance.

- The radar system was configured to transmit a known payload, and the C-client application "**radar_84_ghz_C**" displayed the counters for correctly and incorrectly received frames.
- Eye diagrams were captured at each distance to assess the quality of the received signal and detect any closing of the eye as the distance increased.
- At around 1 meter, as shown in Figure 5.3a, the eye diagram is wide open, indicating excellent signal quality, with no bit errors, confirming successful communication.

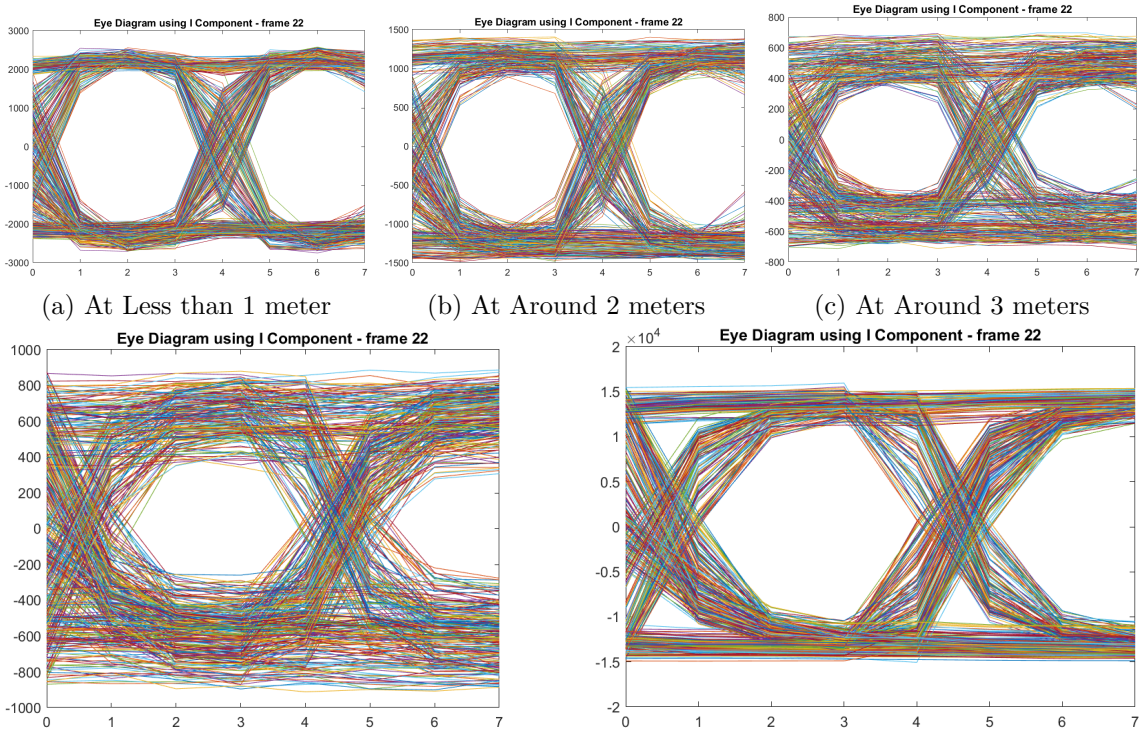
This has been used as an ideal reference to the later cases where the receiving radar unit will be moved at different distances away from the transmitting radar unit.

- At 2 meters (Figure 5.3b), the eye diagram shows minor degradation, but remains open, with all frames demodulated correctly confirming successful communication at this distance.

- At 3 meters (Figure 5.3c), the eye shows noticeable closing. Still, communication remains stable with no bit errors, reflecting the increasing distance's effect on signal quality, but communication remains robust with null demodulation errors.

- At 4 meters (Figure 5.3d), the eye diagram is more closed, indicating signal degradation but the system still manages to demodulate the payload with zero error rates.

- After doubling the transmitter power at the same 4-meter distance (Figure 5.3e), the eye diagram opens significantly wider, further improving signal quality while maintaining the zero error rate.



(a) At Less than 1 meter (b) At Around 2 meters (c) At Around 3 meters

(d) At Around 4 meters (e) At Around 4 meters

Figure 5.3: Eye Diagrams of the Communication Payload Samples Plotted for a Single Frame at Different Distances Separating the two Radar Units

5.3 Conclusion

The testing and validation of the upgraded system demonstrated its ability to handle both radar sensing and communication tasks concurrently without interference. The radar system successfully detected targets in real-time, as shown in both the range-Doppler map and zero-Doppler cut, confirming accurate target detection and distance measurement.
The communication tests, particularly through eye diagram analysis at various distances, showcased how the system's performance degrades with increasing range. At short distances (1m, 2m), the system performed exceptionally well, with wide-open eye diagrams and no errors. As the distance increased to 3m and 4m, the eye diagrams revealed signal degradation, though the system continued to demodulate the payload successfully also with no error rates.
Overall, these results confirm that the integrated radar-communication system meets the required performance metrics, functioning reliably under both radar and communication modes. This validation marks a significant success in advancing radar technology with integrated communication capabilities.

Chapter 6

Conclusions and Next Steps

The successful integration of communication capabilities into an existing PMCW radar system represents a significant leap forward in radar and communication technologies, particularly in the context of automotive applications. This thesis focused on creating a robust radar system capable of simultaneously performing high-resolution sensing and real-time data transmission without compromising either function. By addressing the complex challenges of integrating communication within a radar framework, this work demonstrates that such co-existence is not only feasible but can also offer tangible benefits to future vehicular systems.

Throughout the project, the design and implementation of advanced firmware and software were paramount. Key contributions include the implementation of critical algorithms for CFO compensation and phase recovery, which ensure reliable communication performance even under challenging conditions. These algorithms, paired with high-level real-time signal processing techniques, enable the system to maintain signal integrity and deliver accurate communication payload demodulation, as demonstrated through extensive laboratory testing. The system's ability to produce wide-open eye diagrams further underscores the effectiveness of the integration.

The real-world implications of this work are substantial. The fusion of radar sensing with communication opens up numerous opportunities for applications such as Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication and ADAS. These technologies are critical to the future of autonomous driving, where vehicles must seamlessly communicate with each other and with infrastructure to ensure road safety, reduce traffic congestion, and improve overall transportation efficiency. The ability of the developed system to operate in both radar and communication modes simultaneously, with high reliability, addresses a fundamental need in the automotive industry—integrating sensors with communication to create smarter, safer, and more connected vehicles.

Moreover, the successful validation of the system in a controlled lab environment, using real-time range-Doppler maps and demodulation metrics, shows that this integrated solution is practical and can be deployed in real-world applications in later stages. The testing process, which evaluated performance under different conditions and distances, further confirmed that the system could potentially meet stringent automotive standards. The findings indicate that future radar systems, built on the foundation laid in this thesis,

will be able to achieve higher functionality and versatility without increasing hardware complexity.

Future Work: A potential extension of this project involves the development of a multi-transmitter, multi-receiver radar system, utilizing orthogonal codes to differentiate between simultaneously transmitting units. This upgraded version would allow the system to enhance its spatial resolution and target detection accuracy. The use of orthogonal coding enables simultaneous transmissions without interference, significantly improving the system's capability to detect multiple targets and operate in more complex environments. Such a system would allow for more robust target tracking, greater range resolution, and enhanced clutter mitigation. Currently, a prototype with these characteristics is under development, and its success could mark a substantial advancement over the current system, pushing radar-communication integration toward even more demanding applications, such as fully autonomous vehicles operating in dense traffic scenarios.

In conclusion, this thesis not only advances the technical integration of radar and communication systems but also provides a blueprint for the next generation of automotive radar systems. The work contributes to the ongoing development of connected, autonomous vehicles, paving the way for safer and more efficient transportation systems. The successful implementation of such a system underscores the importance of continued innovation in radar technology, particularly as the demand for more intelligent, interconnected vehicles grows. This work represents a crucial step toward realizing the full potential of radar-communication systems in real-world automotive scenarios.

Bibliography

[1] M. Caffa, F. Biletta, and R. Maggiora. Binary-phase vs. frequency modulated radar measured performances for automotive applications. https://doi.org/10.3390/s23115271, 2023. Accessed: 2024-4-11.

[2] Analog Design. Ad-fmcdaq2-ebz board. https://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/eval-ad-fmcdaq2-ebz.html. Accessed: 2024-06-05.

[3] Analog Design. Adf4372 microwave wideband synthesizer with integrated vco. https://www.analog.com/en/products/adf4372.html. Accessed: 2024-06-10.

[4] Analog Design. Admv7320 e-band upconverter sip, 81 ghz to 86 ghz. https://www.analog.com/en/products/admv7320.html. Accessed: 2024-06-10.

[5] Analog Design. Admv7420 e-band low noise downconverter sip, 81 ghz to 86 ghz. https://www.analog.com/en/products/admv7420.html. Accessed: 2024-06-10.

[6] Jürgen Hasch, Eray Topak, Raik Schnabel, Thomas Zwick, Robert Weigel, and Christian Waldschmidt. Millimeter-wave technology for automotive radar sensors in the 77 ghz frequency band. *IEEE Transactions on Microwave Theory and Techniques*, 60(3):845–860, 2012. Accessed: 2024-04-25.

[7] MathWorks. comm.coarsefrequencycompensator system object. https://it.mathworks.com/help/comm/ref/comm.coarsefrequencycompensator-system-object.html, 2024. Accessed: 2024-05-15.

[8] Cleve Moler and MathWorks Steve Eddins. Faster finite fourier transforms in matlab. https://www.mathworks.com/company/technical-articles/faster-finite-fourier-transforms-matlab.html, 2001. Accessed: 2024-04-25.

[9] Matthew G Parker, Kenneth G Paterson, and Chintha Tellambura. Golay complementary sequences, 2003. Accessed: 2024-04-18.

[10] Y. Wang, K. Shi, and E. Serpedin. Non-data-aided feedforward carrier frequency offset estimators for qam constellations: A nonlinear least-squares approach. https://doi.org/10.1155/S1110865704403175, 2004. Accessed: 2024-05-15.

[11] Xilinx. Amd kintex7 fpga kc705 evaluation kit. https://www.xilinx.com/products/boards-and-kits/ek-k7-kc705-g.html. Accessed: 2024-06-05.

[12] Xilinx. Ams101 evaluation card. https://www.xilinx.com/products/boards-and-kits/hw-ams101-g.html. Accessed: 2024-06-05.

[13] Zhengqi Xu, Ke Wu, Xiaoming Liu, Jing Jin, and Jianjun Zhou. A 24 ghz mimo pmcw automotive radar with doppler compensation. In *2021 IEEE MTT-S International*

Wireless Symposium (IWS), pages 1–3, 2021. Accessed: 2024-04-25.

[14] L. Zheng, M. Lops, Y. C. Eldar, and X. Wang. Radar and communication coexistence: An overview. https://arxiv.org/pdf/1902.08676, 2019. Accessed: 2024-4-25.