applied sciences

Article

Software-Defined Platform for Global Navigation Satellite System Antenna Array Development and Testing

Diogo Gomes [1], Diogo Baptista [1], Hugo Dinis [2,3,*], Paulo M. Mendes [2,3] and Sérgio Lopes [4]

[1] Bosch Car Multimedia Portugal, S.A., Rua Max Grundig, n.° 35, 4705-820 Braga, Portugal
[2] Center for MicroElectromechanical Systems (CMEMS-Uminho), University of Minho, 4800-058 Guimarães, Portugal
[3] LABBELS–Associate Laboratory, 4710-057 Braga, Portugal
[4] Centro Algoritmi/LASI, University of Minho, 4804-533 Guimarães, Portugal
* Correspondence: hdinis@dei.uminho.pt

Abstract: With the increasing demand for accurate and robust positioning solutions, the use of GNSS antenna arrays has gained significant attention. However, their development and testing are frequently constrained by the inflexibility of traditional hardware platforms, often requiring extensive reconfiguration throughout the development cycle. This paper presents a platform based on a system on chip to develop a highly flexible software-controlled system that is capable of directly sampling up to 16 antenna elements. Multibeam digital beamforming is implemented using the available FPGA resources and the resulting signal is reproduced by the integrated DAC and can be connected to any conventional single antenna GNSS receiver. This paper presents the architecture of the platform, detailing its components and capabilities. Our experimental results demonstrate that the system can phase shift every channel with errors of less than 0.5° and can reconfigure 4 simultaneous beams of a 16-antenna array at speeds of 1.2 kHz, and 20 beams at around 400 Hz. The average delay introduced by each channel of the system is around 381 ns with a maximum deviation of 1.05 ns. The delay was also measured for the implementation using 4 beams, which achieves a slightly bigger average delay of 384.6 ns while keeping the variation to 5 to 6 ns. This system is intended to be used as the backbone for the development of antenna arrays and beamforming algorithms. Given its flexibility, it is not necessary to develop new hardware between development iterations or even for different systems, as only the software layer needs to be modified. Consequently, it is possible to expedite the development stage before producing dedicated solutions for industrial applications.

Keywords: GNSS; SoC; FPGA; digital beamforming; antenna array

check for updates

Citation: Gomes, D.; Baptista, D.; Dinis, H.; Mendes, P.M.; Lopes, S. Software-Defined Platform for Global Navigation Satellite System Antenna Array Development and Testing. *Appl. Sci.* **2024**, *14*, 9621. https://doi.org/10.3390/app14219621

Academic Editors: Andrea Prati and Piotr Gas

Received: 29 May 2024
Revised: 2 October 2024
Accepted: 4 October 2024
Published: 22 October 2024

1. Introduction

GNSSs are considered a crucial element in enabling higher levels of autonomy [1], but the current limitations associated with the connection integrity of these systems impair its importance in the decision-making systems of autonomous vehicles (AVs) [2]. The limitations can be multipath and non-line-of-sight signals, highly dynamic environments, and susceptibility for intentional or accidental interference, such as jamming and spoofing [3–5]. These limit the overall performance of current driving automation and must be overcome to guarantee the success of AVs. Systems with higher degrees of automated driving will eventually require instantaneous, centimeter accurate positioning [6].

Software-defined antennas (SDAs) are being explored as platforms to support next-generation wireless capability [7–9]. The benefit of such antennas can be leveraged to improve the quality of GNSS services in severely degraded environments [10]. Smart adaptive antennas that track GNSS satellites have been shown to help alleviate both situations [11]. However, their development presents certain challenges. SDAs are composed of the antenna array and the control unit, both requiring extensive development and testing. Antenna characterization is usually performed by antenna pattern measurements in an anechoic chamber [9,12], requiring a platform to control the phases introduced to each channel. On the control unit side of the system, complex control algorithms must be developed to maximize the benefits of SDAs for GNSSs [13]. These algorithms handle interference mitigation and signal enhancement, among other tasks. Their development requires extensive testing in anechoic chambers as well as "real world" tests [14]. More factors like array size, phase, and delay calibration between elements, and overall cost, which increases with the number of elements, impose technical challenges for the deployment of such solutions in the mass market sector. Therefore, commercially available systems usually utilize 2–3 elements and employ only interference mitigation rejection [15,16].

When developing antenna arrays and beamforming algorithms, we can resort to a multitude of methods. With simulations, we can quickly and inexpensively validate the proposals. However, real-world validation is extremely valuable. Alternatively, or together with simulations, we can develop a dedicated system to test the proposal. With these dedicated systems, optimal performance can be achieved. Nevertheless, it is an expensive and time-consuming process, as each system is custom-made. Resorting to radio frequency system on chips (RFSoCs) can alleviate this. RFSoCs integrate hardened high-speed data converters with already established FPGA and microprocessor architectures [17]. This unlocks highly versatile and adaptable systems for quick development and real-world testing with much lower costs.

This paper presents a receiver-independent implementation of a software-defined platform with the use of RFSoC technology, aiming to facilitate its integration with existing automotive receivers. This system is intended to be used as the backbone for the development of antenna arrays and beamforming algorithms. Given its flexibility, it is not necessary to develop new hardware between development iterations or even for different systems, as only the software layer needs to be modified. Consequently, it is possible to expedite the development stage before producing dedicated solutions for industrial applications. Previous works we have found in our research have not explored this. The proposal of an antenna array for GNSS applications is not in the scope of this paper. The main contribution of this paper is the development of a platform for standalone solutions that can be integrated into existing receivers as if it were a traditional single antenna, i.e., the output of the array and FPGA is a single signal fed to the receiver.

This paper continues with a summary of related works in the following section. In Section 3, we present the reported system and its requirements. In Section 4, we present the system's architecture, and in Section 5, we present the calibration results and performance metrics. Finally, in Section 6, the conclusions and future work are discussed.

2. Related Work

The recent adoption of RFSoC technology for 5G massive MIMO systems has made them a strong contender for the future of digitally reconfigurable beamforming systems. The research presented in [18] introduces a new digital beamformer for better wireless communication, especially in 5G networks. It uses a smart method to handle interference issues, demonstrating impressive results with minimal impact on signal quality. The approach reduces the need for complex and costly hardware, making it more efficient for multiple-user scenarios. This study also highlights how this technology can improve the performance of millimeter-wave frequencies in 5G communication systems, addressing challenges like signal loss and bulky hardware. Overall, it presents a cost-effective and energy-efficient solution for enhanced wireless communication.

The study in [19] investigated multi-tile synchronization (MTS) and calibration for beamforming in wireless LTE communication, emphasizing optimal phase shifts among transmitters within an RFSoC context. The research demonstrated effective synchronization and calibration methods through digital beamforming for consistent phase shifts in varied communication scenarios.

Several works report on FPGA and algorithm performance optimization of massive adaptive beamforming for 5G mobile networks [20,21]. In these works, the focus is the reduction of the consumed resources and power while improving the throughput of the system. They demonstrate the feasibility, the flexibility, and the capability of FPGA system for digital beamforming, which validates our claim to use these systems to construct our development platform for beamforming algorithms and antenna arrays.

The literature on beamforming for vehicular GNSS applications predominantly focuses on contemporary SoCs, particularly those manufactured by Xilinx, Inc. (San Jose, CA, USA). These SoCs are favored for their efficient development process and high-performance capabilities, attributed to the integration of FPGA technology with a central processing unit (CPU).

In [14], the researchers implemented a four-antenna single-output single beam system capable of functioning with any receiver. They opted to use external commercially off the shelf (COTS) ADCs that communicate with the SoC in order to reduce development time; however, this introduces unnecessary delays due to extra overhead performed by communications. As mentioned by the authors, the majority of the system was constructed in FPGA due to its superior throughput, except for the weight calculation task, which was assigned to the algorithm executing in the CPU. The approach appears to be rational given the existing technological capabilities within the automobile industry. However, it may become insufficient as autonomous vehicle technology advances, leading to higher demands for signal accuracy and resilience.

The project carried out by the authors in [11,22,23] improves in the integration of the components by using the novel RFSoCs provided by Xilinx, which includes GHz capable ADCs on the die itself, therefore reducing the overall size of the system. This is a system that, according to the authors, is capable of being integrated in an unmanned aerial vehicle due to its small footprint. The system integrates the receiver into the die, using a multiple-output beamformer closely integrated with a custom-made receiver to improve overall performance and reduce overall package size. This solution differs from our implementation as the integration of the receiver is not our goal.

The literature contains several works that aim to develop antenna array technology for GNSS applications, such as jammer multipath suppression [24], and the design of compact [25] and low-cost [26] arrays. Similarly, works focusing on beamforming algorithms are reported [27–29]. The development of these arrays and the experimental validation of the algorithms could be significantly accelerated, and the results could be better validated with the use of a platform such as the one we propose in this paper. A similar approach is employed in the development of GNSS receivers, where FPGA-based platforms are employed to develop advanced receivers [30–32].

3. System Development

3.1. Problem Overview

The SDA system is depicted in Figure 1, and it demonstrates how the blocks of the system interact. An RF frontend composed of low-noise amplifiers and bandpass filters is connected to a hypothetical 16-element antenna array. This block is responsible for amplifying the signal and eliminating the noise outside the band of interest. This is different to conventional systems, which perform analog down-conversion before sampling the signal, and therefore requiring less components overall.

The SDA controller oversees signal acquisition via integrated ADCs and subsequent down conversion. The baseband signals are then fed to the beamforming controller, where crucial digital beamforming operations, such as complex weight multiplication and summation of sampled signals, take place. This involves combining array channel samples in a manner that promotes constructive interference for signals at specific directions, while inducing destructive interference for others and, therefore, forming beams directed at the intended GNSS satellites. The GNSS receiver provides the controller with the satellite positions to which the beams should point after signal decoding.

Following these operations, the computed signal undergoes conversion back to its analog form at its carrier frequency, facilitated by a DAC. The analog signal is then transmitted to a GNSS receiver for position calculation.

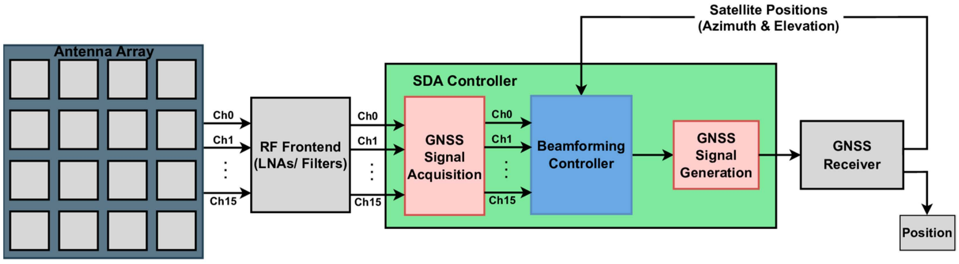

Figure 1. Block diagram of the SDA system.

3.2. System Requirements

The system processes signals from the upper L-band from all constellations, as demonstrated by the S_{21}, or transmission loss, measurement results presented in Figure 2, where each of the 16 channels operates from 1.45 GHz to 1.75 GHz. The choice of this band was due to project requirements aiming for the development of antenna arrays for the L1/E1 band. However, the FPGA platform can operate with other frequency bands, and even perform dual-band operation, with new software and calibration. At its output it produces an analog signal like a regular antenna to facilitate seamless integration with commercial receivers.

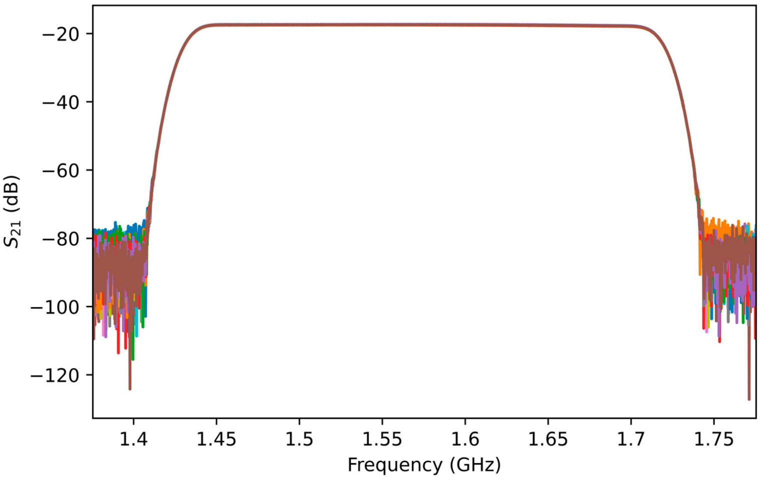

Figure 2. Measured S_{21} parameters of the FPGA's 16 channels; a demonstration that the board's current configuration can process signals in the 1.45–1.7 GHz range.

By ensuring negligible impact on positioning, time delays are carefully managed. The control unit, integrated with a receiver, supplies the positioning solution to an autonomous vehicle's decision unit, meeting Society of Automotive Engineers' standards with a total delay of not less than 100 ms [33]. Efforts are made to minimize signal delays introduced by the control unit, particularly crucial when the vehicle is in motion at highway speeds.

The feedback interface from the receiver utilizes the NMEA 0183 specification [34], the most common among commercial receivers. This interface facilitates effective communication between the control unit and the receiver.

A noteworthy feature is the system's ability to support multiple beam implementations, allowing for the testing and implementation of various beamforming algorithms without significant reconfiguration. The system is equipped for live testing of diverse beamforming algorithms without extensive reconfiguration.

Designed with scalability in mind, the system accommodates varying numbers of antennas and beams, ensuring adaptability to systems with a larger number of antennas and beams. Additionally, the implementation is crafted for reusability, making it applicable in platforms with a feasible cost for integration into production vehicles.

4. System Architecture

The output from specific single-output beamforming architectures can be seamlessly transformed back to its analog state using a DAC. Given that commercial receivers are predominantly configured to interface with the more common fixed reception pattern antennas (FRPAs), they inherently expect a singular input. Single-output multiple-beam beamformers, however, require the satellite position to calculate the direction of arrival (DOA) of the signals.

The computation of the complex weight multiplications in beamforming can be performed in parallel by the FPGA, minimizing time delay between the input and output. This is critical to achieve the signal synchronization native to beamforming systems.

The development platform chosen for this project is the Xilinx ZCU216 evaluation board, which uses the XCZU49DR RFSoC. This platform contains 16 ADCs, allowing the direct sampling of up to 16 antenna element arrays.

The diagram of the system, presented in Figure 3, is divided into three main blocks. The hardened data converters, represented in green, capture (ADC) and reproduce (DAC) L1/E1 frequency signals. The captured signals are transmitted by an AXI4-Stream containing their I/Q representation. The beamforming IP core, represented by the blue block in the center of the diagram, performs the complex weight multiplication and summing operation of the sampled signals. The single product of this core is then converted to an analog signal to replicate a traditional single antenna output that can be fed to a GNSS receiver. The hardened data converters were implemented using the Data Converter IP core from Xilinx [17]. Its configuration is explored in the next subsection. The beamforming IP core is a custom-made IP that receives the ADCs data from the Data Converter IP and outputs the beamforming solution back to one DAC of the Data Converter IP. Further explanation on the beamforming IP can be found below.

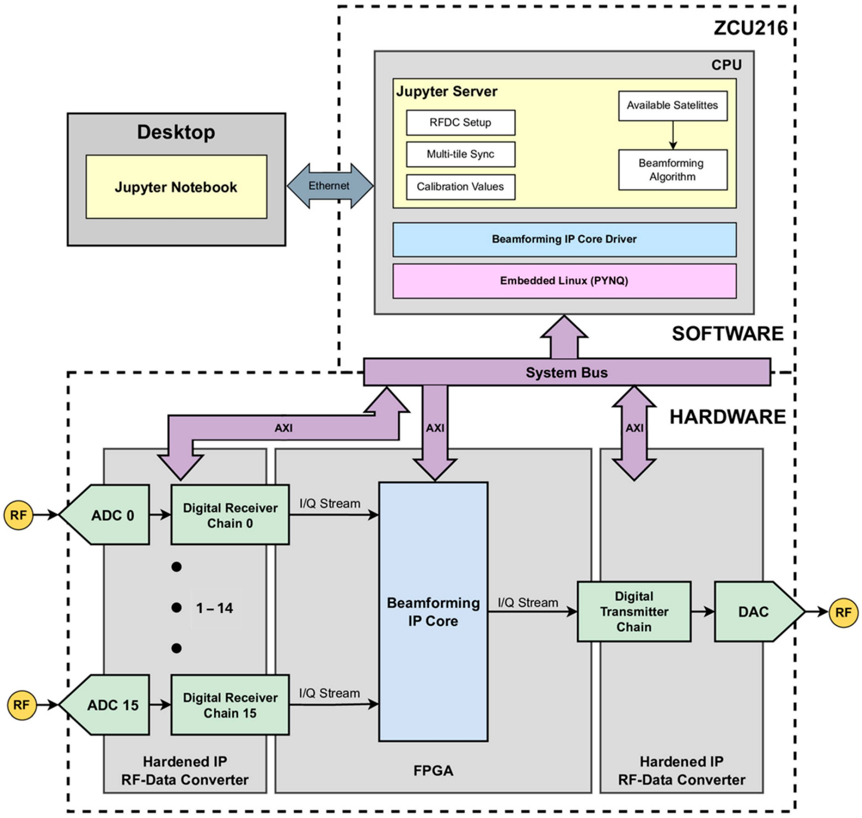

Figure 3. Block diagram of the developed system.

This hardware is controlled by the processor side using the system bus which relies on the AXI4-Lite protocol. The processing system can be programmed live by an Ethernet connection to a desktop to achieve fast testing of beamforming algorithms.

4.1. RF-Data Converters

The system was developed just for L1/E1 frequency signals centered at 1.57542 GHz. The XCZU49DR SoC contains 16 14-bit ADCs and DACs, capable of sampling at a maximum frequency of 2.5 GHz and 9.85 GHz, respectively. This means that the sampled signal sits in the second Nyquist zone in relation to the maximum sampling frequency, causing aliasing. Nevertheless, this characteristic can be used to directly sample the signal without IF demodulation. This allows the signal to be folded to the first Nyquist zone while keeping the information. It is to be noted that the aliased signal is flipped left to right when folding. However, this can be compensated by the internal digital mixer present in the data path.

The sampling sequence is described in Figures 4 and 5, where the L1/E1 signal (centered at 1.57542 GHz) is undersampled at 2.4 GHz, shifting to 0.82458 GHz, as represented in Figure 3. Then, the resulting signal is demodulated to baseband resorting to a −0.82458 GHz signal, as shown in Figure 4.

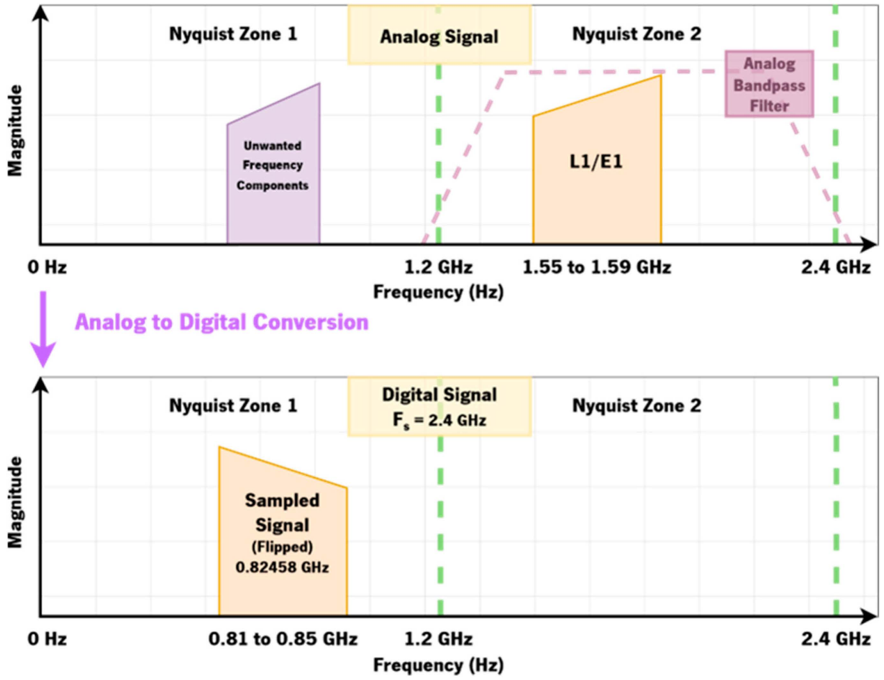

Figure 4. Second Nyquist zone sampling of the L1/E1 band.

Some additional considerations were taken for the configuration of the data converter. The desire for the maximum sampling frequency possible to improve performance and consequently reduce delay in the FPGA fabric while keeping in mind the complexity, resource usage of the beamforming IP core, has resulted in the following:

- Sampling at 2.4 GHz.
- Convert the signal to baseband IQ and flip it using the complex mixer at a frequency of −824.58 MHz.
- Decimate by 8, resulting in a rate of 300 MHz.

After beamforming, the digital to analog conversion is performed by the DAC. In order to negate signal delay in the FPGA fabric, a similar approach was taken to that of the ADCs, not requiring any additional logic, contrary to if different rates or word lengths were used.

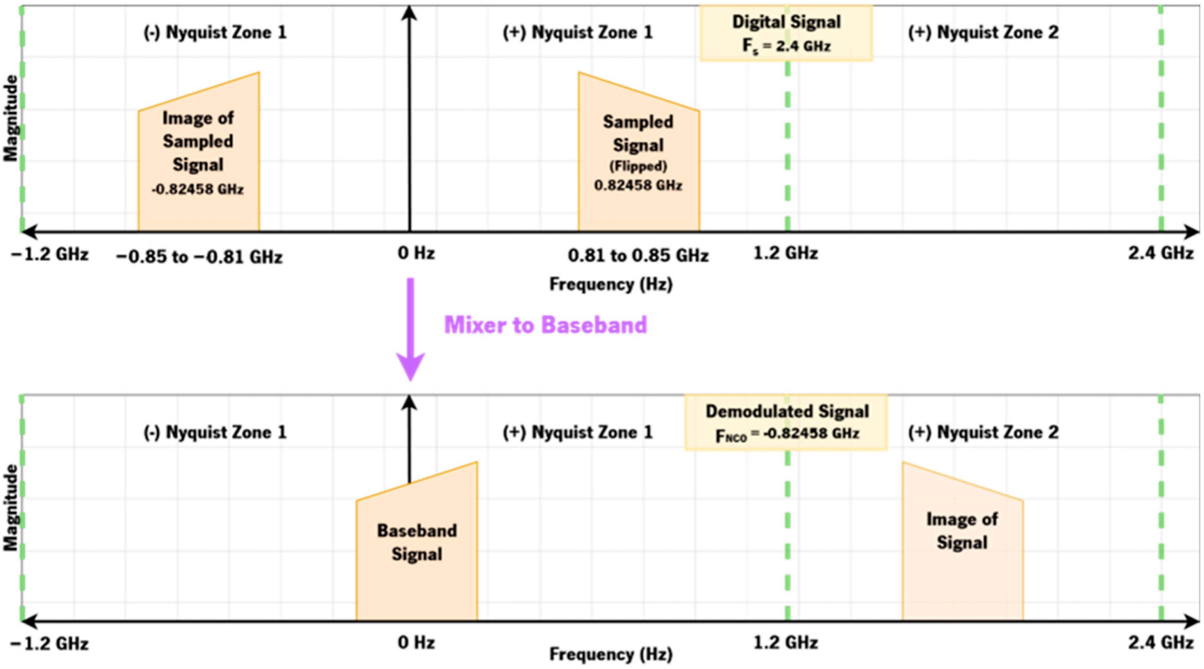

Figure 5. Mixing the signal to baseband.

The data path preceding the DAC, interpolates the beamformed samples by 8 and mixes the baseband complex signal with the original carrier frequency of 1.57542 GHz. This is then converted to analog at a frequency of 2.4 GHz.

4.2. Beamformer

The core operations of digital beamforming are the complex weight multiplication and summation of the sampled signals. This implies the simultaneous complex weight multiplication of every sampled channel when trying to reduce signal delay between input and output of the system. This can be computed using Equation (1):

$$b(t) = \sum_{k=0}^{k=N} W_k(t) \cdot X_k(t) \tag{1}$$

where $X_k(t)$ represents the complex sampled signal of channel k at a given time t. This is multiplied by the complex weight destined to the same channel $W_k(t)$. The output of the beamformer b(t), is the product of the summation of these N multiplications.

Figure 6 presents the beamforming IP core developed with Vitis High-Level Synthesis (HLS) [35] and added to the block design in Vivado. This block can generate multiple beams in a parallel manner. Multiple beams can be generated simultaneously just by summing every beam pointing to different satellites. This was implemented within the FPGA fabric present in the SoC. The complex samples are transmitted from the RF-ADCs to the IP core using the AXI4-Stream protocol adopted by Xilinx. The complex weights are transmitted by the processing system using the AXI4-Lite protocol. Both the in-phase and quadrature weight components are stored back-to-back in register spaces, where they can be updated independently. In this way, the system can adapt to many use cases. The multiplication and summation of the samples uses the dedicated DSP48E2 blocks for faster processing.

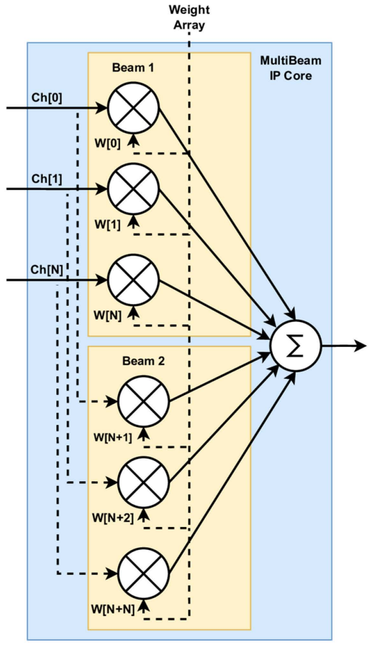

Figure 6. Implementation of the multibeam IP core.

4.3. Beam Control Algorithm

The computation of the required complex weights according to satellite positioning is performed by the processing system, albeit Software. This enables a highly flexible implementation while maintaining the minimal delay requirements.

At this stage of development, the control algorithm does not perform amplitude modulation; it only controls the signals' phases. Given the azimuth and elevation of any satellite, the algorithm calculates the phase shift for every channel to align the main beam of the antenna array towards the specified direction. Furthermore, due to the nature of the implementation, new algorithms can be tested and implemented without changing any hardware components.

The algorithm is derived from the array factor of planar arrays with M elements equally spaced in the x direction, N elements equally spaced in the y direction, and equal amplitude across all elements. By ensuring these conditions, the phase between element-to-element that ensures the main beam points in the $[\phi_0, \theta_0]$ direction becomes the following [36]:

$$\beta_x = -\frac{2\pi d_x}{\lambda} \sin\theta_0 \cos\phi_0, \tag{2}$$

$$\beta_y = -\frac{2\pi d_y}{\lambda} \sin\theta_0 \cos\phi_0, \tag{3}$$

where β_x and β_y are the phase between elements in the x and y direction that ensure the desired main beam direction, d_x and d_y are the spacing between elements in the x and y direction, and λ is the wavelength of the incoming signal. These phases must then be used to evaluate the total phase in each element ($\beta_{xy} = (x - 1)\, \beta_x + (y - 1)\, \beta_y$) and then converted to a complex phase delay ($W_k = W_{xy} = e^{-j\beta}{}_{xy}$) before updating the IP core that handles the multiplication of the incoming elements' signal with the complex delay.

5. Results

The system was tested for both its functionality and performance. The following sections present the different tests performed and the corresponding results, namely, the hardware resource usage of the FPGA, software computational speed, phase control, and delay induced in the signal.

5.1. Hardware Resource Usage

After implementing the hardware design with the Vivado tools, a test was performed to assess the impact of the beam numbers in the utilization of FPGA resources. The test was performed for 5 different beam numbers, from 1 to 20 beams.

Table 1 contains the utilization results achieved by the different implementations. Only the look-up tables (LUTs), flip-flops (FFs), and DSP48E2 results are shown since they are the ones that vary in relation to the beam number. Both the absolute number of components and the percentage in relation to the available resources are shown. The implementation is occupying a small area of FPGA and there is a linear increment of resources used in relation to the number of beams used. DSP48E2 usage is the more crucial since there are fewer available DSPs overall. This will lead to routing problems if a larger number of beams is needed due to the disparity of utilization. It is important to note, however, that these values are for a 300 MHz frequency, meaning that if a larger implementation is needed, the reduction of clock rate will consequently reduce the number of resources used.

Table 1. Hardware resource usage according to the number of beams used.

	LUT	FF	DSP48E2
Available	425,280	850,560	4272
1 Beam	13,501 (3.17%)	12,680 (1.49%)	64 (1.5%)
4 Beams	21,724 (5.11%)	25,182 (5.11%)	512 (11.99%)
8 Beams	32,990 (7.76%)	41,008 (4.82%)	512 (11.99%)
12 Beams	43,046 (10.12%)	56,304 (6.62%)	768 (17.98%)
20 Beams	63,908 (15.03%)	87,569 (10.30%)	1280 (29.96%)

5.2. Software Performance

The speed at which the processor computes the phases for each element is a crucial aspect of the system that directly impacts its performance. The highly dynamic environment introduced by a vehicle necessitates fast beam control.

The arm processor present in the RFSoC contains four cores, which enables the capability of parallel calculation of the different beam phases, therefore increasing performance.

To test the processing capabilities, a loop was made where all phases were calculated for all 16 elements over 50 times in a quick succession while constantly changing the $[\phi_0, \theta_0]$ of said beams, forcing the change of the complex delays of the IP core.

Table 2 contains the average time and corresponding frequency achieved when calculating the channel phases. Our system can perform beamforming of 20 beams at over 400 Hz (2.5 ms), which is significantly more than what is required to use current GNSS receivers. However, it is important to consider that these results are largely due to the simple control algorithm present and may worsen when more complex algorithms are used.

Table 2. Average computation time and frequency according to the number of beams processed.

	Single Threaded	Multithreaded
1 Beam	0.42 ms (2407 Hz)	-
4 Beams	1.34 ms (746 Hz)	0.81 ms (1231 Hz)
8 Beams	2.67 ms (374 Hz)	1.21 ms (822 Hz)
12 Beams	4.04 ms (247 Hz)	1.49 ms (670 Hz)
20 Beams	6.69 ms (149 Hz)	2.44 ms (409 Hz)

5.3. Phase Control Validation

The Keysight E5071C vector network analyzer (VNA) was used to validate the phase shifts introduced by the system. This was determined by measuring the S_{21} of each channel. Port 1 of the VNA was connected to each ADC through uncalibrated cables and baluns, and Port 2 was connected to the DAC.

Three situations were taken into consideration when validating the functionality of the system. Firstly, without introducing any phase shifts the phases of each channel were measured. This was performed to assess the phases introduced by each channel when left uncalibrated. Secondly, the previous results were used to calibrate the system by aligning all channel's phases with respect to ADC 0, and new measurements were taken to check if the calibration worked as expected. These results are demonstrated in Table 3. After that, the channel phases were introduced for a beam with θ = 45° and ϕ = 90° and the phases were measured again to assess the difference between the computed phases and the measured ones. All the results presented are for the central frequency of L1, 1.57542 GHz.

Table 3. Phase compensation results.

	Uncompensated	**Compensated**
ADC 0	-	-
ADC 1	21.62°	−0.04°
ADC 2	1.32°	0.08°
ADC 3	20.46°	0.16°
ADC 4	−0.49°	0.11°
ADC 5	19.14°	0.14°
ADC 6	1.62°	0.26°
ADC 7	18.85°	0.07°
ADC 8	1.49°	−0.06°
ADC 9	21.79°	0.29°
ADC 10	2.99°	0.18°
ADC 11	22.79°	0.16°
ADC 12	3.75°	−0.53°
ADC 13	21.84°	−0.22°
ADC 14	1.63°	0.13°
ADC 15	22.05°	0.09°

Table 4 shows the results for the calibration step of the process. The RFSoC is capable of synchronizing every channel using Xilinx's "Multi-tile Synchronization". This delays every sample by a fixed amount of time and is repeatable even during power cycles. However, due to the uncalibrated cables and baluns, the measured phases are not equal but can be easily compensated. This was achieved by using the measurements to approximate the phase of each channel to the one of the ADC 0, resulting in repeatable differences of less than a degree.

Table 4. Calculated and measured phases for θ = 45° and ϕ = 90°.

	Calculated	**Measured**	**Difference**
ADC 0	−101.82°	−101.73	−0.09°
ADC 1	−203.65°	−203.57°	−0.08°
ADC 2	−305.47°	−305.40°	−0.07°
ADC 3	−407.29°	−407.51°	+0.22°
ADC 4	−101.82°	−102.06°	+0.24°
ADC 5	−203.65°	−203.58°	−0.07°
ADC 6	−305.47°	−305.55°	+0.08°
ADC 7	−407.29°	−407.41°	+0.12°
ADC 8	−101.82°	−101.65°	−0.17°
ADC 9	−203.65°	−203.73°	+0.08°
ADC 10	−305.47°	−305.54°	+0.07°
ADC 11	−407.29°	−407.23°	−0.06°
ADC 12	−101.82°	−101.46°	−0.36°
ADC 13	−203.65°	−203.57°	−0.08°
ADC 14	−305.47°	−305.50°	+0.03°
ADC 15	−407.29°	−407.25°	−0.04°

In Table 4, the results of beam generation are presented. The "Calculated" column contains the outputs of the beam control algorithm when generating a main beam with θ = 45° and ϕ = 90°. When measured, the resulting phases are very close to the calculated counterparts, producing at most a difference of −0.36°.

5.4. Delay Measurements

Considering that the system will be placed in the signal path between the antennas and a GNSS receiver, it is important to measure the delay introduced, given that it may negatively affect the positioning.

The group delay of the system was extracted using the previous S_{21} phase measurements. Figure 7 presents the delay results obtained for the 16 channels across the E1/L1 frequency range. From the results, it is possible to verify that while not perfectly constant, the system achieves very little variations across all channels.

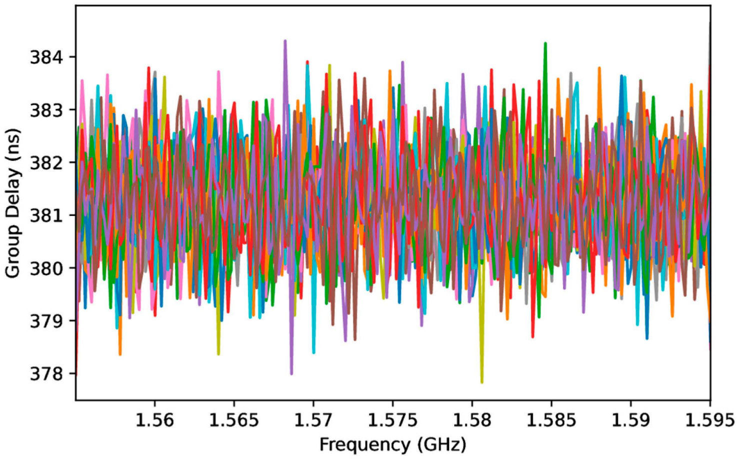

Figure 7. Group delay results for L1/E1 frequency bandwidth.

Table 5 summarizes the mean and standard deviation of the group delay for the 16 channels for 1 beam. All channels have an average value of 381 ns with a maximum deviation of 1.05 ns. The delay was also calculated for the implementation using four beams, which achieves a slightly bigger average delay of 384.6 ns while keeping the variation to 5 to 6 ns. This is due to the more complex implementation of the beamforming IP core which results in a slightly bigger signal path. The influence of this delay on the positioning is yet to be validated through open-sky tests. For static scenarios, this delay should not be an issue, but in the case of dynamic scenarios in which the car can move with velocities up to 120 km/h, this means the signal reaches the receiver ~12.7 µm apart from the time it was available at the output of the antenna. Further testing needs to take place to understand the implications of this delay between the antennas' output and the receiver.

Table 5. Group delay mean and standard deviation for each channel for 1 beam.

	Mean (ns)	Std (ns)	ADC	Mean (ns)	Std (ns)
ADC 0	381.23	0.83	ADC 8	381.27	0.92
ADC 1	381.27	0.88	ADC 9	381.28	1.05
ADC 2	381.26	0.86	ADC 10	381.26	0.95
ADC 3	381.28	0.89	ADC 11	381.28	0.9
ADC 4	381.24	0.89	ADC 12	381.27	0.94
ADC 5	381.27	0.95	ADC 13	381.3	0.98
ADC 6	381.25	0.96	ADC 14	381.26	0.91
ADC 7	381.3	0.92	ADC 15	381.3	0.91

6. Conclusions and Future Work

This paper presents a platform for the development stage of antenna arrays and beamforming algorithms. It allows us to test any antenna array with up to 16 feeds and unlimited algorithms without extensive reconfiguration. Also, it can directly feed conventional single antenna receivers and RF test equipment. With this system, it is not necessary to develop new hardware between design iterations or even for different systems, as only the software layer needs to be modified. With this, we aim to expedite the development stage of arrays and beamforming algorithms before producing dedicated solutions for industrial applications.

Several tests were conducted to assess the functionality and performance of the implementation, as well as the potential impact of scaling the solution. It is worth noting that even when 20 beams are produced, the resource utilization in the ZCU216 remains far below the available capacity. In terms of software, execution speeds of 1.2 kHz were achieved for the four-beam multithread configuration, with room for more complex algorithms to be implemented. After calibration and when directing the main lobe towards $\theta = 45^\circ$ and $\phi = 90^\circ$, the highest phase difference between the computed and measured phase was 0.36°. The average group delay measurement yielded 384.6 ns.

In the future, system optimization will be performed. Live tests using the receiver feedback and an antenna array should also be performed. Stationary and moving tests should be carried out to assess the impact the system has on the receiver's performance.

Author Contributions: Conceptualization, D.G., D.B., H.D., P.M.M. and S.L.; methodology, D.G., D.B., H.D., P.M.M. and S.L.; software, D.G., D.B. and S.L.; validation, H.D. and P.M.M.; formal analysis, D.G., D.B. and H.D.; investigation, D.B. and H.D.; resources, D.B., P.M.M. and S.L.; data curation, D.G., D.B. and H.D.; writing—original draft preparation, D.G., D.B. and H.D.; writing—review and editing, D.B., H.D., P.M.M. and S.L.; supervision, P.M.M. and S.L.; project administration, D.B., H.D. and P.M.M.; funding acquisition, P.M.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by national funds, through the Operational Competitiveness and Internationalization Programme (COMPETE 2020) [Project nº 179491; Funding Reference: SIFN-01-9999-FN-179491].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: Authors Diogo Gomes and Diogo Baptista were employed by the company Bosch Car Multimedia Portugal, S.A. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Chen, L.; Zheng, F.; Gong, X.; Jiang, X. GNSS High-Precision Augmentation for Autonomous Vehicles: Requirements, Solution, and Technical Challenges. *Remote Sens.* **2023**, *15*, 1623. [CrossRef]
2. Swaminathan, H.B.; Sommer, A.; Becker, A.; Atzmueller, M. Performance Evaluation of GNSS Position Augmentation Methods for Autonomous Vehicles in Urban Environments. *Sensors* **2022**, *22*, 8419. [CrossRef] [PubMed]
3. Hussain, A.; Akhtar, F.; Khand, Z.H.; Rajput, A.; Shaukat, Z. Complexity and Limitations of GNSS Signal Reception in Highly Obstructed Enviroments. *Eng. Technol. Appl. Sci. Res.* **2021**, *11*, 6864–6868. [CrossRef]
4. Abosekeen, A.; Iqbal, U.; Noureldin, A.; Korenberg, M.J. A Novel Multi-Level Integrated Navigation System for Challenging GNSS Environments. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 4838–4852. [CrossRef]
5. Silva Lorraine, K.J.; Ramarakula, M. A Comprehensive Survey on GNSS Interferences and the Application of Neural Networks for Anti-Jamming. *IETE J. Res.* **2023**, *69*, 4286–4305. [CrossRef]
6. Reid, T.G.R.; Houts, S.E.; Cammarata, R.; Mills, G.; Agarwal, S.; Vora, A.; Pandey, G. Localization Requirements for Autonomous Vehicles. *SAE Int. J. Connect. Autom. Veh.* **2019**, *2*. [CrossRef]

7. Dötterböck, D.; Pany, T.; Lesjak, R.; Prechtl, T.; Tabatabaei, A. PRN Sequence Estimation with a Self-Calibrating 40-Element Antenna Array. *Navig. J. Inst. Navig.* **2023**, *70*, navi.600. [CrossRef]
8. BniLam, N.; Principe, F.; Crosta, P. Large Array Antenna Aperture for GNSS Applications. *IEEE Trans. Aerosp. Electron. Syst.* **2024**, *60*, 675–684. [CrossRef]
9. Hehenberger, S.P.; Elmarissi, W.; Caizzone, S. Design and Installed Performance Analysis of a Miniaturized All-GNSS Bands Antenna Array for Robust Navigation on UAV Platforms. *Sensors* **2022**, *22*, 9645. [CrossRef]
10. Razgūnas, M.; Rudys, S.; Aleksiejūnas, R. GNSS 2×2 Antenna Array with Beamforming for Multipath Detection. *Adv. Space Res.* **2023**, *71*, 4142–4154. [CrossRef]
11. Kunzler, J.W.; Warnick, K.F.; Bartschi, J.M.; Ammermon, S.M.; Burnett, M.C. Bench Testing a System on Chip Adaptive Beamformer For GPS. In Proceedings of the 2022 United States National Committee of URSI National Radio Science Meeting (USNC-URSI NRSM), Boulder, CO, USA, 4–8 January 2022; pp. 300–301.
12. Schirmer, C.; Rugamer, A.; Kotterman, W.A.T.; Landmann, M.H.; Del Galdo, G. Evaluation of Array Antenna Systems for GNSS Applications Using Wave-Field Synthesis in an OTA Laboratory. In Proceedings of the 2017 11th European Conference on Antennas and Propagation (EUCAP), Paris, France, 19–24 March 2017; pp. 3370–3374.
13. Wang, H.; Yao, Z.; Fan, Z.; Yang, J.; Liu, G. A Robust STAP Beamforming Algorithm for GNSS Receivers in High Dynamic Environment. *Signal Process.* **2020**, *172*, 107532. [CrossRef]
14. Arribas, J.; Fernandez-Prades, C.; Gomez Lopez, M.A.; Ruiz, T.R. Receiver-Independent GNSS Smart Antenna for Interference Mitigation. In Proceedings of the 2022 10th Workshop on Satellite Navigation Technology (NAVITEC), Noordwijk, The Netherlands, 5–7 April 2022; pp. 1–8.
15. Pérez-Marcos, E.; Cuntz, M.; Konovaltsev, A.; Kurz, L.; Caizzone, S.; Meurer, M. CRPA and Array Receivers for Civil GNSS Applications. In Proceedings of the 2023 IEEE/ION Position, Location and Navigation Symposium (PLANS), Monterey, CA, USA, 24–27 April 2023; pp. 318–328.
16. Egea-Roca, D.; Arizabaleta-Diez, M.; Pany, T.; Antreich, F.; Lopez-Salcedo, J.A.; Paonni, M.; Seco-Granados, G. GNSS User Technology: State-of-the-Art and Future Trends. *IEEE Access* **2022**, *10*, 39939–39968. [CrossRef]
17. AMD Xilinx Zynq UltraScale+ RFSoC RF Data Converter v2.6 Gen 1/2/3/DFE LogiCORE IP Product Guide (PG269). Available online: https://docs.amd.com/r/en-US/pg269-rf-data-converter (accessed on 10 June 2023).
18. Ullah, K.; Venkatakrishnan, S.B.; Volakis, J.L. Millimeter-Wave Digital Beamforming Receiver Using RFSoC FPGA for MIMO Communications. In Proceedings of the 2022 IEEE 22nd Annual Wireless and Microwave Technology Conference (WAMICON), Clearwater, FL, USA, 27–28 April 2022; pp. 1–4.
19. Dusari, N.R.; Rawat, M. Multi Tile Synchronization and Calibration of Xilinx RF SoC ZCU216 for Digital Beamforming. In Proceedings of the 2022 IEEE Wireless Antenna and Microwave Symposium (WAMS), Rourkela, India, 5–8 June 2022; pp. 1–5.
20. Irfan, R.; Rasheed, H.; Toor, W.A. FPGA-Based Low Latency Inverse QRD Architecture for Adaptive Beamforming in Phased Array Radars. *Radioengineering* **2017**, *26*, 851–859. [CrossRef]
21. Lopes, F.F.; Silva, S.N.; Fernandes, M.A.C. FPGA Implementation of the Adaptive Digital Beamforming for Massive Array. In Proceedings of the 2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring), Helsinki, Finland, 25–28 April 2021; pp. 1–5.
22. Kunzler, J.W.; Ammermon, S.M.; Warnick, K.F. Progress Toward Airborne GPS Spatial Filtering Powered by Recent Advances in FPGA Technology. In Proceedings of the 2020 Intermountain Engineering, Technology and Computing (IETC), Orem, UT, USA, 2–3 October 2020; pp. 1–6.
23. Kunzler, J.W. Managing Radio Frequency Interference in Vehicular Multi-Antenna Transceivers. Ph.D. Thesis, Brigham Young University, Provo, UT, USA, 2022.
24. Huang, L.; Lu, Z.; Xiao, Z.; Ren, C.; Song, J.; Li, B. Suppression of Jammer Multipath in GNSS Antenna Array Receiver. *Remote Sens.* **2022**, *14*, 350. [CrossRef]
25. Madni, A.; Khan, W.T. Design of a Compact 4-Element GNSS Antenna Array with High Isolation Using a Defected Ground Structure (DGS) and a Microwave Absorber. *IEEE Open J. Antennas Propag.* **2023**, *4*, 779–791. [CrossRef]
26. Marranghelli, E.A.; La Valle, R.L.; Roncagliolo, P.A. Simple and Effective GNSS Spatial Processing Using a Low-Cost Compact Antenna Array. *IEEE Trans. Aerosp. Electron. Syst.* **2021**, *57*, 3479–3491. [CrossRef]
27. Hu, Y.; Bian, S.; Li, B.; Zhou, L. A Novel Array-Based Spoofing and Jamming Suppression Method for GNSS Receiver. *IEEE Sens. J.* **2018**, *18*, 2952–2958. [CrossRef]
28. Gong, Y.; Wang, L.; Yao, R.; Zhang, Z. A Robust Method to Suppress Jamming for GNSS Array Antenna Based on Reconstruction of Sample Covariance Matrix. *Int. J. Antennas Propag.* **2017**, *2017*, 9764283. [CrossRef]
29. Xu, Z.; Dong, Q.; Li, S.; Yue, F.; Wang, M.; Xia, Z.; Chen, X.; Zhang, S.; Zou, G.; Wang, H. Robust Wideband Interference Suppression Method for GNSS Array Antenna Receiver via Hybrid Beamforming Technique. *Remote Sens.* **2024**, *16*, 1913. [CrossRef]
30. Majoral, M.; Arribas, J.; Fernández-Prades, C. Implementation of a High-Sensitivity Global Navigation Satellite System Receiver on a System-on-Chip Field-Programmable Gate Array Platform. *Sensors* **2024**, *24*, 1416. [CrossRef]
31. Majoral, M.; Fernández-Prades, C.; Arribas, J. A Flexible System-on-Chip Field-Programmable Gate Array Architecture for Prototyping Experimental Global Navigation Satellite System Receivers. *Sensors* **2023**, *23*, 9483. [CrossRef] [PubMed]

32. Qiu, T.; Wang, X.; Du, Q.; Sun, Y. An Innovative FPGA-based Low-complexity and Multi-constellations Compatible GNSS Acquisition Scheme. *IET Radar Sonar Navig.* **2021**, *15*, 370–381. [CrossRef]
33. Thorn, E.; Kimmel, S.C.; Chaka, M. *A Framework for Automated Driving System Testable Cases and Scenarios*; Department of Transportation, National Highway Traffic Safety Administration: Washington, DC, USA, 2018.
34. National Marine Electronics Association. *NMEA 0183 Standard*; National Marine Electronics Association: Severna Park, MD, USA, 2023.
35. Advanced Micro Devices AMD Vitis HLS. Available online: https://www.amd.com/en/products/software/adaptive-socs-and-fpgas/vitis/vitis-hls.html (accessed on 10 September 2024).
36. Balanis, C.A. *Antenna Theory: Analysis and Design*, 3rd ed.; John Wiley and Sons, Inc.: Hoboken, NJ, USA, 2005; ISBN 047166782X.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.