Development of FPGA-Based Radar Back-End

Santiago Elian Mallerman, Juan Pablo Pascual, and Jorge Cogo

***Abstract*—This work presents the development of the digital back-end of a pulsed radar based on Field Programmable Gate Array (FPGA) technology. The development includes the stages dedicated to the generation of diverse baseband signals, including square pulses, chirps, and Barker codes, and the processing of the received signal through a series of correlators with adaptable sampling rates for pulse compression and oversampling scenarios. The system is physically implemented taking advantage of the facilities offered by a FlexRIO® development platform, as well as the associated development environment. The system parameters, like pulse duration, guard interval, and pulse repetition period, as well as parameters specific to each modulation scheme, are widely adjustable to meet user requirements such as maximum range, Doppler resolution and sensitivity. The stages of signal generation are verified digitally through experimental tests using RF measurement equipment, demonstrating a high consistency between them and what is theoretically predicted. The correlation stage is validated using a synthetic target generated by a digital delay and including a noisy channel, showing a strong correlation between the obtained output and the theoretical expectation. These results constitute an important advance in the development of a low-power, reconfigurable monostatic pulsed radar platform that will serve as a versatile testbed for algorithmic experimentation.**

Link to graphical and video abstracts, and to code:
https://latamt.ieeer9.org/index.php/transactions/article/view/10102

***Index Terms*—correlator bank, FPGA development, radar signal generation, reconfigurable radar**

I. Introduction

RADAR systems play an essential role in various applications, including surveillance, navigation, and remote sensing, due to their ability to detect and track targets in complex environments. Traditional radar systems are often constrained by fixed configurations, limiting their flexibility to different operational scenarios, hindering the exploration of alternative estimation algorithms and delaying progress in radar technology innovation.

Numerous studies on digital back-ends for radar systems have emerged to achieve greater flexibility. The studies have demonstrated the potential of digital technologies addressing a wide range of aspects, including signal processing, waveform generation, dynamic reconfiguration, among others. These works explore the use of Software Defined Radio (SDR) platforms, Digital Signal Processors (DSPs), Field-Programmable Gate Arrays (FPGAs) and Radio Frequency System-on-Chips (RFSoCs).

In [1], the authors analyze a generic architecture for radar signal processing stages on commercial SDR boards; while in [2], the authors address the development of a high-resolution radar prototype based on a commercial FPGA board. The prototype uses separate antennas for transmission and reception, allowing both functions to be performed simultaneously and enabling configurable range and velocity resolution, making it suitable for use in primary radar and synthetic aperture radar applications.

In [3] the authors summarize the steps involved in the development of a pulsed radar controller and signal processor implemented into a commercial FPGA-based transceiver and present application examples in the W, Ka, and 449 MHz bands. In [4] the authors describe the implementation of a configurable FPGA-based radar waveform generator.

In [5], the authors perform a real-time implementation of waveform generation and adaptive filtering using digital techniques on an FPGA. They test different time-bandwidth products, including phase modulation, linear frequency modulation, and nonlinear frequency modulation. In [6], a hardware integrated platform for radar signal processing is designed, where frequency modulated waveforms are processed using a dynamic reconfiguration method on an FPGA. In [7], the authors synthesize the combination of linear frequency modulation and phase shifts, which results in a lower peak-to-side lobe ratio and a narrower main lobe width.

In [8] the authors present a hardware platform that utilizes three PFGAs and one DSP to implement digital signal processing of a pulse-Doppler radar, and briefly describe the implementation of algorithms to digital down conversion, digital beam forming, pulse compression and moving target detection, including strategies of constant false alarm rate (CFAR). In [9] the authors present the design of an integrated wideband and narrowband radar signal direct sampling and digital processing system based on FPGA. In [10] the authors point out the development trends of FPGA-based radar signal processing.

In [11] the authors propose efficient and scalable parallelization methods of the pulse-Doppler radar signal processing chain using an eight cores DSP. In [12] the authors present a prototype of Ku-band radar based on a FPGA RFSoC device, for the use onboard a small Unmanned Aerial Vehicle (UAV) for detecting aerial targets. In [13] the author presents the design of a high-speed radar signal processor based on FPGA

The associate editor coordinating the review of this manuscript and approving it for publication was Roberto S. Murphy (*Corresponding author: Santiago Mallerman*).

Santiago Mallerman is with Instituto Balseiro, Universidad Nacional de Cuyo and Comisión Nacional de Energía Atómica, S. C. Bariloche, Argentina (e-mail: santiago.mallerman@ib.edu.ar).

J. P. Pascual is with Consejo Nacional de Investigaciones Científicas y Técnicas, Instituto Balseiro, Universidad Nacional de Cuyo and Comisión Nacional de Energía Atómica, S. C. Bariloche, Argentina (e-mail: juanpablo.pascual@ib.edu.ar).

J. Cogo is with Centro Interdisciplinario de Telecomunicaciones, Electrónica, Computación y Ciencia Aplicada, Universidad Nacional de Río Negro, S. C. Bariloche, Argentina (e-mail: jcogo@unrn.edu.ar).

This work was supported by UNCuyo 80020240400157UN 2373/2025-C.S. and by UNRN PI 40-B-1159.

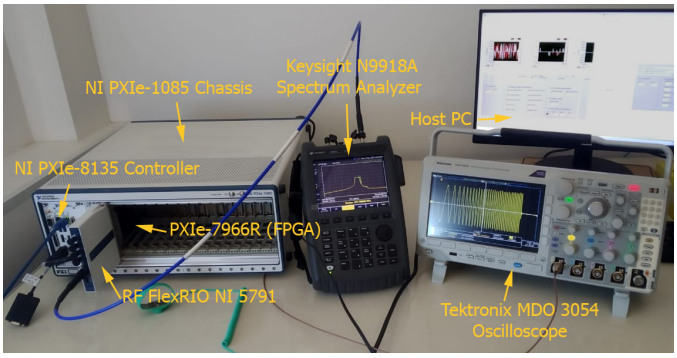
architecture, that incorporates a fusion processing algorithm to integrate different radar signal bands.

Commercial examples of compact radars featuring flexible digital back-ends designed for diverse user needs include those developed by the Japanese company FURUNO [14]. These radars use solid-state transmitters, operate in the X-band, and offer high-resolution and short-range capabilities. The digital processing stage is built on the FlexRIO platform from National Instruments.

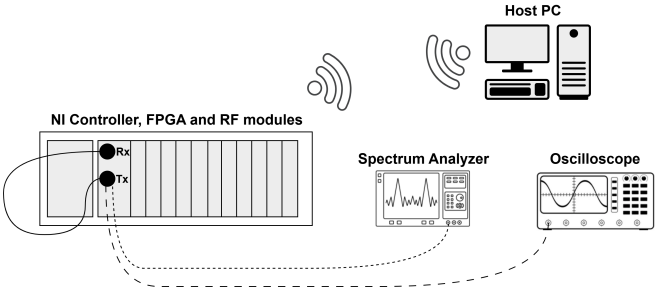
Inscribed within the state-of-the-art of these existing works, the primary motivation behind this work is to address the limitation of fixed configurations in radar systems by developing the digital back-end for an experimental, low-power, reconfigurable monostatic pulsed radar platform. This platform can accommodate various configurations and serves as a versatile testbed for algorithmic experimentation, enhancing radar performance and capabilities. This flexibility and adaptability can also be extended in the development of a radar with an antenna array featuring electronically steerable beams, controlled through the phase or relative frequency between its radiating elements.

The proposed radar platform uses FPGA technology and a commercial radio-frequency (RF) front-end. In addition, to enabling algorithmic experimentation, the reconfigurable radar platform serves as a valuable tool for education and training in radar engineering and signal processing. Its intuitive interface and real-time feedback mechanisms provide students and researchers with hands-on experience in radar system design, implementation, and optimization.

Section II presents an overview of hardware capabilities, development environment and modules features. Section III describes the control logic and essential parameters for radar operation. Section IV covers the generation and transmission stages of the radar system, while Section V focuses on the reception stage and subsequent digital processing. Section VI analyzes the test and measurements performed to validate the implemented system. Finally, Section VII discusses the conclusions.



(a)



(b)

Fig. 1. Experimental setup of the radar prototype: (a) photograph of the RF measurement instrumentation, (b) schematic of the test and measurement configuration, with each RF connection shown using a different line style.

TABLE I
RF TRANSCEIVER ADAPTER SPECIFICATIONS

Parameter	Value
RF Frequency Range	200 MHz – 4.4 GHz
Instantaneous Bandwidth	100 MHz
ADC Sample Rate	130 Msps
DAC Sample Rate	130 Msps
ADC Resolution	14-bit
DAC Resolution	16-bit
Front panel connectors	1 RX, 1 TX (Full-Duplex)
I/Q Data Streams	Independent I & Q

II. GENERAL SETUP OF THE DEVELOPED SYSTEM

The radar system development platform is built on National Instruments' FlexRIO® technology, incorporating the NI 5791 module for RF signal processing and the PXIe-7966 module for FPGA operations. Both modules are mounted on a chassis PXIe-1085. The chassis with both modules is shown on the left side of Fig. 1a. It also includes the PXIe-8135 module that consists of an Intel Core i7 embedded controller for PXI Express systems, used for communication with the computer and for controlling the different modules during the development stage.

The RF module serves as the RF front-end solution, providing comprehensive signal handling for both the transmission and the reception stages. During transmission, it handles the conversion of channel signals using the digital-to-analog converter (DAC), applies low-pass filtering to each channel, performs quadrature mixing for frequency upconversion, and subsequent amplification. During reception, the module is responsible for filtering, low-noise amplification, frequency downconversion, and digitization of signals using the analog-to-digital converter (ADC).

Key features and capabilities include a clock rate of 130 MHz, 14-bit data acquisition capabilities for in-phase (I) and quadrature (Q) channels, upconversion and downconversion of RF signals ranging from 200 MHz to 4.4 GHz, instantaneous bandwidth up to 100 MHz, flexible configuration and seamless integration with the FPGA module [15]. These features are summarized in Table I.

The development of the digital back-end within the FPGA module [16] was driven by LabVIEW® due to its native support. The implemented system consists of two clock domains, as depicted in the functional diagram of Fig. 2. One clock is derived internally from the master clock using FPGA clock management resources. This clock operates at 32.5 MHz within the generation and transmission stages, where the

Signal Generation block, responsible for pulse generation, is located. The generated signal is stored in a FIFO to be used as a reference signal in the reception stage. Subsequently, the signal is conditioned to fix potential phase differences, carrier frequency offsets or gain imbalances between the I and Q channels in the transmission port. This signal is then stored in a structure, denoted as Tx FIFO, which facilitates clock domain crossing without data loss.

The other clock domain operates at 130 MHz and serves as the master clock, managing the digital signal feeding the DAC of the transmission stage, the signal from the ADC of the reception stage, and supporting the Control Logic block in parallel. The signal from the ADC enters the Rx conditioner block, where it is filtered and compensated to correct imbalances in the quadrature demodulator. Next, the resulting signal is correlated with the reference signal obtained from the reference FIFO using a series of correlators. The number of correlators selected depends on the pulse shape and whether oversampling in range is required. These two parameters define the sampling frequency of the data in range dimension.

The configurable control logic, sends trigger signals throughout the entire system, defining reception, transmission and pulse generation windows, along with guard times to preserve the reception circuitry.

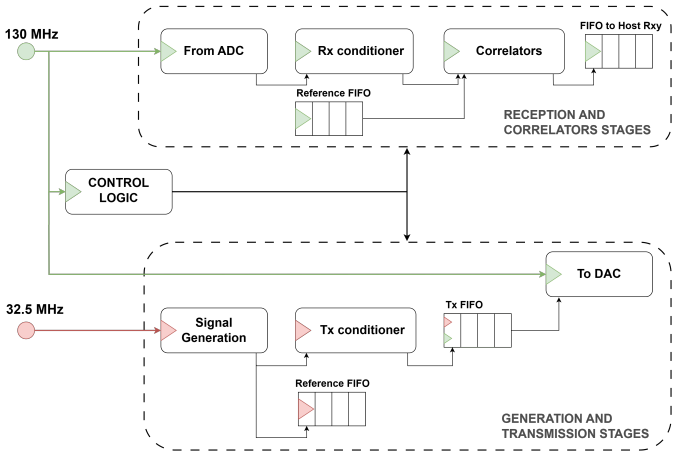


Fig. 2. Functional diagram of the digital circuit implemented in the FPGA.

Fig. 3 presents the user-friendly interface, which consists of three main panels displayed in the following order: Radar Parameters, Tx Signal Conditioner & Stats, and Rx Signal Conditioner & Stats. As its name suggests, the Radar Parameters panel division is designed for easy adjustment of the radar parameters within the Control Logic framework. It also includes a separate division for signal generation settings and sampling frequency, which are discussed in subsequent sections. The Tx and Rx Signal Conditioner & Stats panels display gain, IQ imbalances, frequency and phase compensation inputs in the transmission and reception chains, respectively. Finally, transmission and reception interesting data stats and LED indicators are shown. A similar user-friendly interface is implemented to enable the configuration of the RF module parameters, such as the frequencies of the local oscillators (LOs) used in the upconversion and downconversion stages, the gains of the amplifiers, and the bandwidths of the different filters.

III. Control Logic

The Control Logic framework forms the backbone of the radar system, enabling real-time, flexible adjustments to essential parameters in response to operational changes, such as pulse duration (τ), guard period (τ_g) and the Pulse Repetition Period (PRP), which provide the radar system with the capability to adapt to diverse target environments and operational scenarios.

Fig. 4 shows the implementation of this block, which takes three input parameters in terms of 130 MHz clock ticks: pulse duration, guard period, and the PRP, corresponding to the image fields `TPW`, `TGW` and `PRPW`, respectively. As outputs, the Control Logic provides the trigger signals that synchronize the entire system operation in both clock domains. To achieve this, ascending and descending counters are employed. These pulses extend beyond the theoretical minimum of 130/32.5 = 4 ticks to 5 ticks of the 130 MHz clock domain. This additional margin is incorporated to ensure robust clock domain crossing by providing adequate setup and hold time margins for the synchronization flip-flops, accounting for potential clock skew and jitter in the clock distribution network, thereby preventing metastability.

IV. Generation and Transmission Stages

The Signal Generation block produces the baseband pulses. The number of samples to be generated is determined by the pulse duration, which is typically a small fraction of the PRP. Therefore, pulse generation can be implemented in the slower clock domain, relaxing timing constraints and simplifying the pulse-synthesis implementation, and reserving the high-speed domain for the ADC/DAC interfaces. This module enables users to choose from various waveform types, including simple square pulses or compressed pulses as chirps or Barker sequences of two different lengths. The pulse compression technique is used to increase the range resolution without degrading the signal-to-noise ratio (SNR), or vice versa [17].

The signal parameters are set using the Radar Parameters panel displayed in Fig. 3. The simple pulse is an unmodulated pulse, whose complex envelope is simply a constant-amplitude signal and it only requires setting the signed 16-bit constant amplitudes for the I and Q components. For chirp waveform generation, the current sample number, start frequency, and end frequency are processed using the `sign` and `beta` controllers within the code. These controllers govern the algebraic and trigonometric operations performed according to

$$s(t) = \cos\left(\pi\beta t^2/\tau\right) + j\sin\left(\pi\beta t^2/\tau\right), \quad 0 \le t \le \tau, \quad (1)$$

where β is the value of the `beta` controller with the sign defined by means of the `sign` controller [17]. This process utilizes the High Throughput function palette, which is optimized for fixed-point operations within single-cycle timed loops, enabling one operation per clock cycle and cycle-accurate control when timing constraints are met [18]. The first

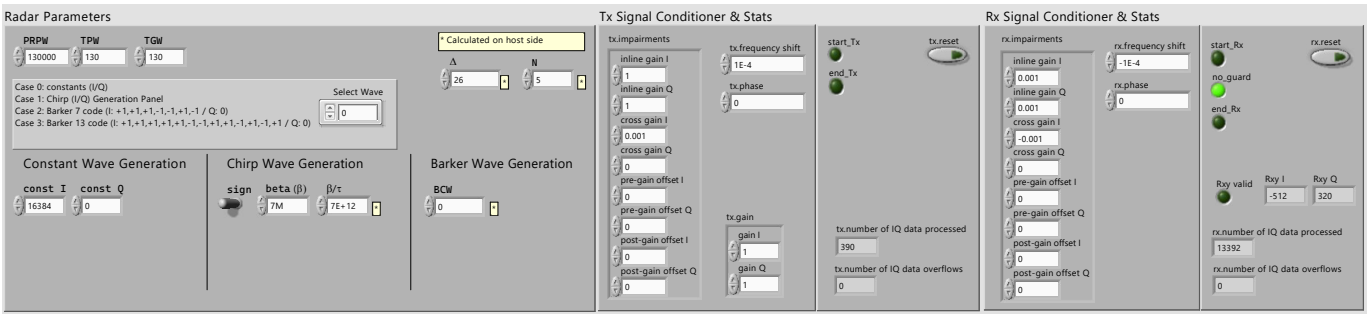


Fig. 3. User interface panels design for easy adjustments of radar parameters.

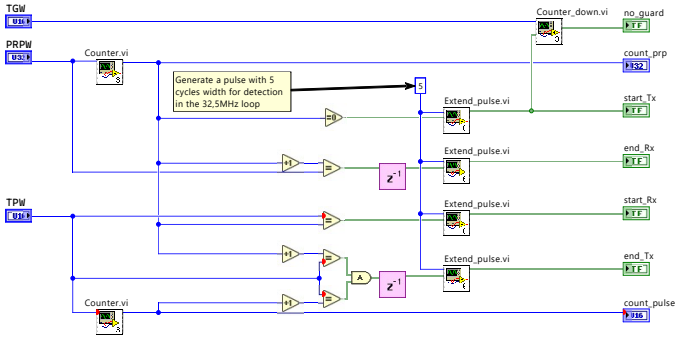


Fig. 4. Control Logic block with input parameters and synchronization signals.

16 samples of the chirp generated waveform are invalid due to the pipeline latency of the High Throughput Sine & Cosine function used. Therefore, these initial samples are discarded, and the signal generation must be extended by this number of samples to compensate.

The provided Barker codes are: the sequence of length 7, $[+1,+1,+1,-1,-1,+1,-1]$, and the sequence of length 13, $[+1,+1,+1,+1,+1,-1,-1,+1,+1,-1,+1,-1,+1]$. Depending on the chosen Barker sequence, a shift register of length $L = 7$ or $L = 13$ is used to cyclically obtain the Barker sequence values. The sequence of the resulting signal is given repeating each value of the Barker sequence a number of samples given by the chip width, determined by $\texttt{TPW}/L$ in clock ticks, that is displayed in the field `BCW`.

In order to validate developed system operation, an experimental test was performed. The signals were acquired using the equipment and the measurement setup shown in Fig. 1. The image of the Fig. 1a corresponds to a photograph of the experimental setup, which includes the specification of the measuring instruments. In the Fig. 1b a schematic diagram of the measurement configuration is provided. The RF transmission output is wired to three points, one at a time: the RF reception input, the spectrum analyzer and the oscilloscope. Each connection is illustrated in the figure using a distinct line style. The host PC communicates with the controller through a local area network. The parameters were chosen so that the signals could be acquired without distortion by the measuring instruments and could be displayed on their screens. Especially, even when the carrier frequency can be set in the range of 200 MHz to 4.4 GHz, a value of 260 MHz was chosen to remain below the oscilloscope bandwidth of 500 MHz.

In Fig. 5 the results of the signal generation are presented. Figs. 5a and 5d show the waveform and the spectrum, respectively, of the simple pulse generated with a pulse duration of 1 μs, and a voltage ranging approximately between $\pm$ 0.15 V, which is consistent with the complex envelope of the pulse generated with the constant I set to $2^{14} = 16384$. The spectrum displays a sinc-like shape centered at the carrier frequency, f_c, of 260 MHz, with a bandwidth of 1 MHz. Figs. 5b and 5e correspond to the waveform and the spectrum, respectively, of a chirp generated with $f_c = 260$ MHz, a PRP of 1 ms, $\beta = 15$ MHz and $\tau = 0.9\,\mu$s, i.e. $\beta\tau = 13.5$. Additionally, Figs. 5c and 5f display the waveform and the spectrum, respectively, of the Barker code of length 7, generated with the same carrier frequency and PRP, and $\tau \simeq 1.08\,\mu s$, which corresponds to $\texttt{TPW} = 140$ samples in the 130 MHz clock domain. These results match the theoretical predictions.

It is important to note that in all cases, the signal is bandpass, and the waveforms shown in Figs. 5a–c correspond to baseband due to the bandpass sampling performed by the oscilloscope itself. Furthermore, in Figs. 5e–f, a sharp spike is observed at 260 MHz, which is consistent with the frequency configured for LO during upconversion. The relatively high amplitude of this tone is attributed to LO leakage into the RF output of the mixer, which is further amplified by the integration time set on the spectrum analyzer during the experiment.

V. Reception and Correlators Stages

The first digital process after acquiring the IQ output from the ADC is the conditioning of the received signal, which is described in Fig. 6. This process uses standard libraries in the LabVIEW® FPGA module. These libraries perform the desired tasks to be executed, taking into account the pipeline delay of each module and the handshaking signals, as these operations take more than one clock cycle to complete, specifically using the four-wire handshaking protocol [18].

The first `subVI`, i.e. modular LabVIEW® component, used is responsible for correcting any imbalances between the IQ branches and it introduces a delay of 4 clock cycles. It then returns data in each cycle according to

$$\begin{bmatrix} I_{out} \\ Q_{out} \end{bmatrix} = \begin{bmatrix} G_I^{inline} & G_Q^{cross} \\ G_I^{cross} & G_Q^{inline} \end{bmatrix} \cdot \begin{bmatrix} I_{offset}^{pre} + I_{in} \\ Q_{offset}^{pre} + Q_{in} \end{bmatrix} + \begin{bmatrix} I_{offset}^{post} \\ Q_{offset}^{post} \end{bmatrix},$$

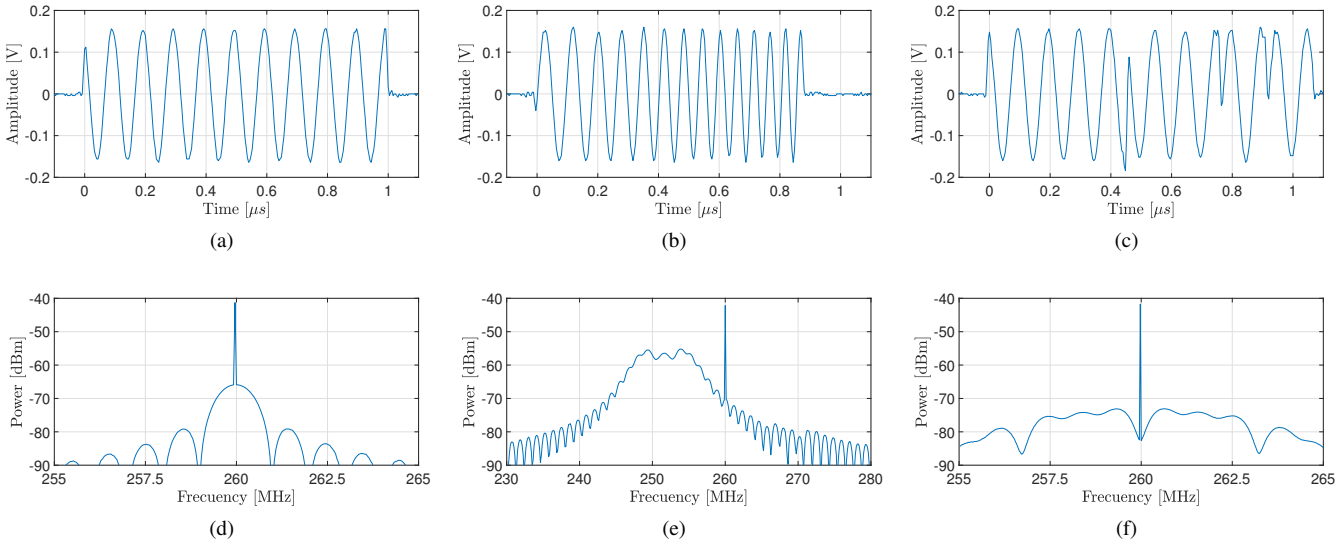


Fig. 5. Measurements of the generated signals. (a) Simple pulse waveform, (b) Chirp waveform, (c) Barker code waveform, (d) Simple pulse spectrum, (e) Chirp spectrum, (f) Barker code spectrum.

where $I_{\text{in}}, Q_{\text{in}}$ are the input in-phase and quadrature components, $I_{\text{out}}, Q_{\text{out}}$ are the corresponding outputs, G^{inline} and G^{cross} represent direct path and leakage gains, and $I^{\text{pre}}_{\text{offset}}, Q^{\text{pre}}_{\text{offset}}$ and $I^{\text{post}}_{\text{offset}}, Q^{\text{post}}_{\text{offset}}$ are the DC offsets applied before and after the gain matrix, respectively.

The second `subVI` applies a digital frequency shift using complex modulation with the sine/cosine harmonic pair generated by a numerically controlled oscillator (NCO). The `rx.frequency shift` control corresponds to the normalized frequency, in the range $[-0.5, +0.5)$. To denormalize the frequency in Hz, this value is multiplied by the sampling rate, 130 MHz. Additionally, the allowed values for the `rx.phase` control are in the range $[-0.5, +0.5)$, corresponding to phase shifts θ between $-180°$ and $+180°$.

This process introduces a delay of 12 clock cycles and then returns data in each cycle according to

$$\begin{bmatrix} I_{\text{out}} \\ Q_{\text{out}} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} I_{\text{in}} \\ Q_{\text{in}} \end{bmatrix}.$$

The reception signal conditioning panel in Fig. 3 should be referenced to associate the controllers with the defined functionality. Subsequently, the resulting signal feeds the correlation instance to obtain the statistic for the detection stage.

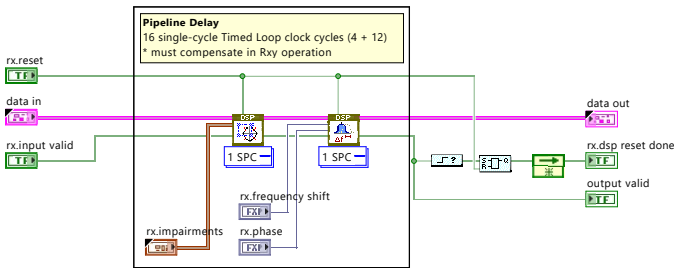


Fig. 6. Implementation of the received signal conditioning.

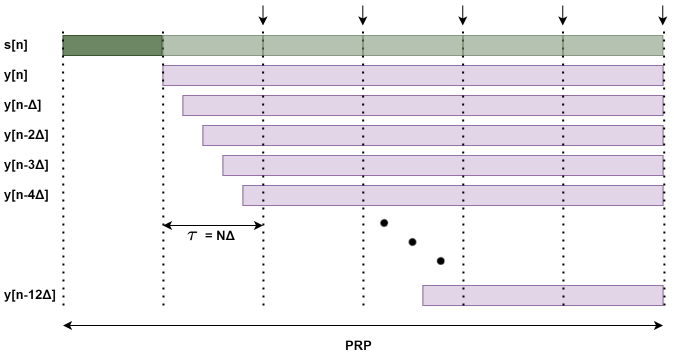


Fig. 7. Correlation instance. Received signal is progressively delayed and fed into each subsequent correlator. Output is returned based on the signal valid of each correlator and the mask.

The oversampling technique, along with a whitening transformation, allows for an increase in the number of independent samples without increasing the bandwidth, β, as pulse compression does [19].

The correlation of the received signal, $y[n]$, with the reference signal, $s[n]$, for every time index $0 \leq k < N_{PRP}$, is defined as

$$r[k] = \sum_{n=0}^{N_{PRP}-1} y^*[n-k]s[n], \tag{2}$$

where N_{PRP} is the number of samples in the PRP. However, depending on the transmitted pulse time–bandwidth product [17], $\beta\tau$, and the oversampling factor used, it is only necessary to compute the correlation results at instants spaced every Δ samples. For simplicity, we constrain $\Delta = \texttt{TPW}/\texttt{N}$, where `N` is an integer. To meet the FPGA timing constraints, we implemented a bank of `N` correlators that simultaneously compute correlations every Δ samples. The number of correlators is limited to $\texttt{N} \leq 13$ by the available FPGA resources, while still providing flexibility for adaptation to different

sampling rates and oversampling scenarios. In the real-time implementation, one input of all correlators in the bank is fed with a periodic extension of $s[n]$, while the other input of each correlator is fed with a version of the received signal $y[n]$ progressively delayed by Δ samples, as schematically illustrated in Fig. 7. This architecture is equivalent to (2). For the sampling instants associated with the index values $i = 1, \ldots, \texttt{PRPW}/\texttt{TPW} - 1$ (indicated with arrows at the top of Fig. 7), each correlator implements the expression

$$r_i[m] = \sum_{n=(i-1)\texttt{TPW}}^{i\texttt{TPW}-1} y^*[n-m]s[n], \tag{3}$$

where the lag m takes the values of the successive feed delays, i.e. $0, \Delta, \ldots, (\texttt{N}-1)\Delta$. It is important to note that only the first N correlators are needed to obtain the result. Accordingly, depending on N, a mask is defined to select the valid correlators. To compensate for the effect of delays in the input signal, the correlation results are delayed in the opposite order to which the samples of $y[n]$ were delayed, resulting in a sample every Δ ticks. Fig. 8 shows the correlator implementation, in which successive complex multiplications between the two signals are accumulated as dictated by (3).

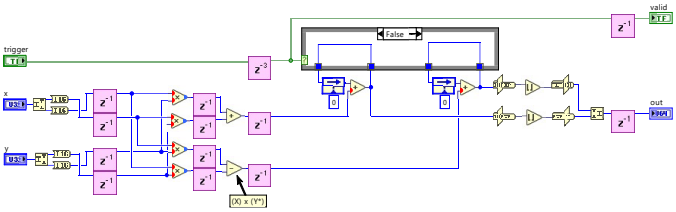


Fig. 8. Correlator implementation.

Finally, the data from the indicated correlator is selected through the multiplexer output based on the signal valid of each correlator and the mask. A pipeline strategy is implemented in order to break down the longest path and ensure that the timing requirements imposed by the 130 MHz clock are met.

Fig. 3 displays the configuration for a 1 μs simple pulse, i.e. with $\beta\tau = 1$, corresponding to 130 samples in the 130 MHz clock domain, 1 μs guard period, and an oversampling factor of 5 over the range resolution, which results in a Δ parameter value of 130/5 = 26, that corresponds to $\texttt{N} = 5$ valid correlators results with lags at sample number 0, 26, 52, 78 and 104.

VI. Results and Discussion

The experimental test configuration in Fig. 1b shows the RF transmission output looped back directly to the RF reception input. A synthetic stationary target at approximately 1.5 km away from the radar is generated entirely within the FlexRIO platform by digitally delaying the transmitted pulse by 1300 samples. Additionally, an additive white Gaussian noise (AWGN) source is included to achieve an SNR of 12 dB. Artificial IQ impairments and a 13 kHz oscillator frequency error are introduced at the transmission stage and compensated for at the reception stage through the implemented conditioning modules. A comparative analysis is performed between the normalized IQ components and normalized energy at the output of the ideal matched filter for the reference pulse generated at 130 MHz against the normalized IQ components at the output of the correlator bank for the received pulse. The ideal matched filter uses the generated reference pulse sampled at 130 MHz and does not include any conditioning stage.

The first test corresponds to a simple pulse duration $\tau = 1\,\mu$s, which corresponds to 130 samples, and a PRP of 1 ms. Figs. 9a and 9d present the IQ components and normalized signal modulus at the output of the ideal matched filter, respectively. Note that the resolution is consistent with that of a simple pulse of width τ, or, equivalently, a range resolution of 0.15 km, computed as $\Delta R = c\tau/2$, where c is the speed of light. Figure 9g shows the output of the correlation stage, where 9 samples are observed over the support 2τ, consistent with the chosen oversampling factor of 5. The output of the correlation stage shows that the target is correctly detected at a distance of 1.5 km from the radar.

Next, Figs. 9b and 9e show the IQ components and the normalized signal modulus at the output of the ideal matched filter for a generated chirp pulse duration of $\tau = 0.9\,\mu$s and $\beta = 5$ MHz. Using the Rayleigh resolution definition, i.e. the peak-to-first null distance of the matched filter waveform output [17], the resolution normalized to the corresponding value for the simple pulse results 0.3. Thus, the improvement in range resolution compared to that expected for the simple pulse is significant. However, it is slightly different from the theoretical approximation, $\Delta R \simeq c/2\beta = 30$ m, or normalized to the corresponding value for the simple pulse, $\Delta R_n \simeq (c/2\beta)\,/\,(c\tau/2) = 1/\beta\tau = 0.22$, which is defined as the width between the peak and the first null. Since $\beta\tau = 4.5$ does not satisfy the $\beta\tau \gg 1$, condition required to obtain the approximation for ΔR, the first null's exact location must be computed from the original expression [17]

$$\Delta R_n = \frac{1}{2}\left(1 \pm \sqrt{1 - \frac{4}{\beta\tau}}\right). \tag{4}$$

Then, from (4) the theoretical normalized resolution results 0.33, which shows a better agreement with the measured resolution than the obtained using the approximation. Finally, as in the case of the simple pulse, in Fig. 9h the output of the correlation stage shows that the target is correctly detected.

Similar comparisons are done for Barker sequences. Figures 9c and 9i correspond to IQ components at the output of the ideal matched filter and the output of the correlation stage, respectively, for a sequence of length $L = 7$ and $\tau = 1.08\,\mu$s, while Fig. 9f displays the normalized signal modulus matched filter output. In this case, the resolution cell width is expressed [17] as $\Delta R_n \simeq 1/L \simeq 0.14$, where the range resolution is given by $c\tau/2L \simeq 23$ m. Consistency between theory and measurements is observed, confirming that the matched filter response reflects the expected resolution improvement provided by the Barker sequence as well as accurate target location.

To quantify the sidelobe level of the waveform outputs of the matched filters, we use the sidelobe level relative to the peak [17], whose values calculated for the experimental results for the chirp pulse and for the Barker sequence are

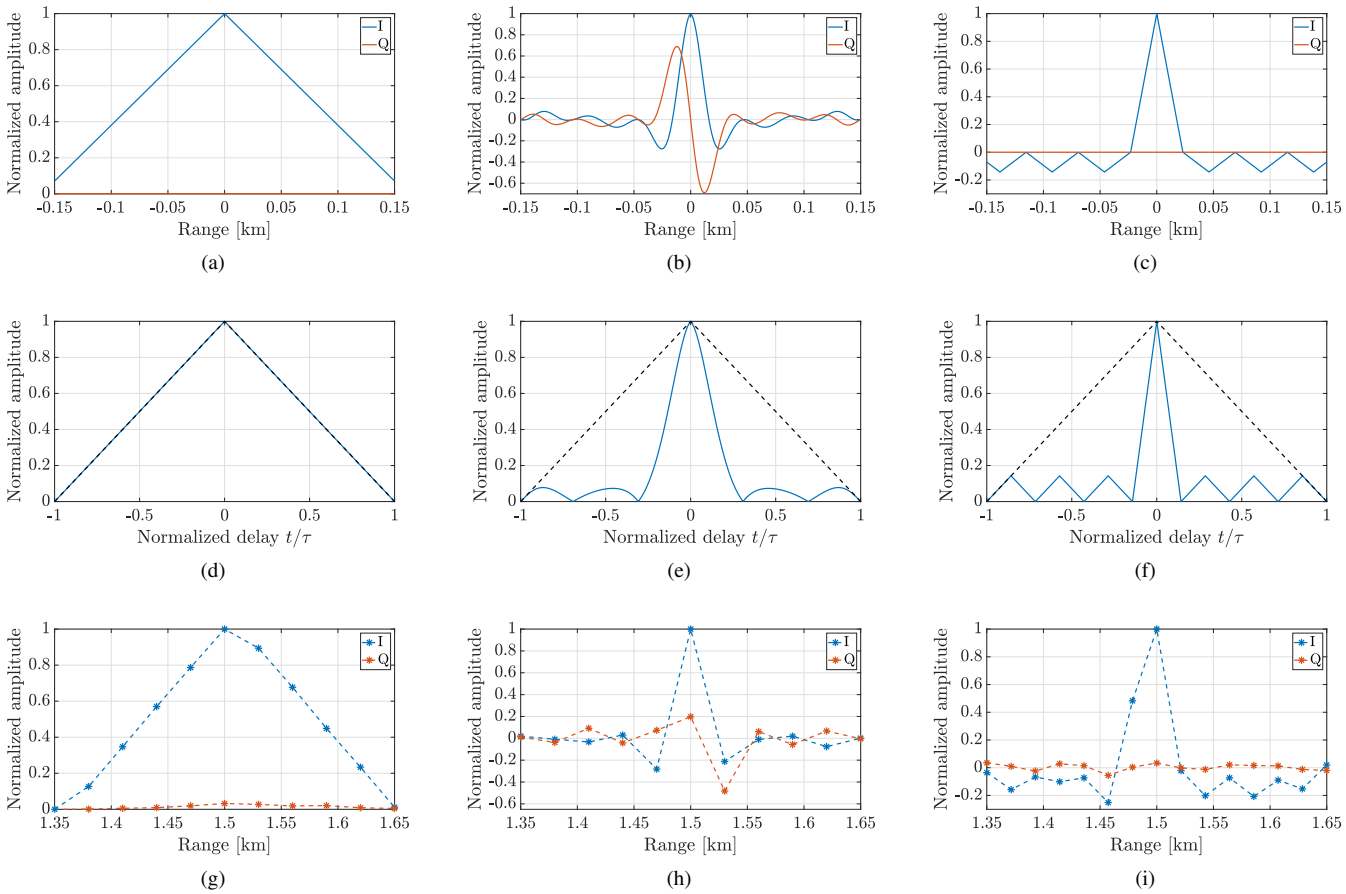


Fig. 9. Normalized IQ components at the output of the ideal matched filter for the reference pulse sampled at 130 MHz. (a) Simple pulse, (b) Chirp β = 5 MHz, (c) Barker code L=7. Normalized signal modulus at the output of the ideal matched filter for the reference pulse sampled at 130 MHz. (d) Simple pulse, (e) Chirp β = 5 MHz, (f) Barker code L=7. Normalized IQ components at the output of the correlator bank for the received pulse. (g) Simple pulse, (h) Chirp β = 5 MHz, (i) Barker code L=7.

-22 dB and -16.9 dB, respectively. These values are in close agreement with the theoretical ones, which are -21.8 dB for the chirp pulse, obtained numerically from its sinc-like waveform and -16.9 dB for the Barker sequence, obtained as $-20\log_{10}(L)$ [17].

Table II describes the resources used in the PXI FPGA Module for FlexRIO® for the design compiled with Xilinx ISE 14.7. The FPGA model is a Virtex-5 XC5VSX95T with 512 MB of onboard DRAM. By examining the design process and comparing different implementations, it is evident that the correlator bank consumes most of the resources. Also, note that there is room to implement additional functionalities.

TABLE II
FPGA RESOURCES USAGE

Device Utilization	Used	Total	Percent
Total Slices	11846	14720	80.5
Slice Register	20889	58880	35.5
Slice LUTs	35802	58880	60.8
Block RAMs	13	244	5.3
DSP48s	100	640	15.6

VII. CONCLUSION

The article covers the development of a digital back-end based on FPGA technology for a flexible radar platform and the validation of the entire system by means of experimental tests. This system successfully implements capabilities for generating and processing diverse baseband signals, establishing it as a versatile tool for studying, developing, and testing various estimation methods under practical conditions.

A comprehensive control logic framework enables dynamic adjustments of critical parameters, allowing the radar system to adapt to a wide range of operational requirements. Additionally, the system's ability to make these adjustments in real time enables it to respond swiftly to changes in the operational environment, enhancing its versatility and functionality.

Advanced DSP modules were integrated to compensate for potential phase differences, carrier frequency shifts, or gain imbalances between the I and Q channels at the receiving port. This approach resulted in a rapid and reliable implementation that improves data quality and system efficiency. Various waveforms were generated to improve range resolution and signal-to-noise ratio, both essential for target detection and enhancing Doppler estimation accuracy. A bank of correlators, adaptable to different sampling rates, was developed, enabling

the system to handle compressed pulses and operate effectively in oversampled scenarios.

The generated signals were verified digitally and through experimental tests using RF measurement equipment, demonstrating close agreement with the theoretical predictions. FPGA resource usage suggests that, as a future step, additional pulse shapes could be synthesized, and correlators may be optimized to further enhance module performance. In conclusion, this research serves as an initial step toward the implementation and practical application of phased array radar technology, with potential for future real-world applications and enhancements.

REFERENCES

[1] T. Brimeyer, C. L. Martin, E. Loew, G. Farquharson, S. Khatri, and S. Paul, "A generic radar processor design using software defined radio," in *33rd Conf. Radar Meteorology Amer. Meteor. Soc.: Cairns, AU.*, vol. 29, jan 2007, pp. 796–806.

[2] H. Nicolaisen, T. Holmboe, K. V. Hoel, and S. Kristoffersen, "High resolution range-Doppler radar demonstrator based on a commercially available FPGA card," in *Int. Conf. Radar*, 2008, pp. 676–681. doi: 10.1109/RADAR.2008.4654006 .

[3] C. Martin, E. Loew, and C. Burghart, "A general purpose FPGA based radar controller and signal processor," in *92nd American Meteorological Society Annual Meeting*, jan 2012.

[4] S. Yoganand, S. S. Babu, and C. Madhu, "FPGA based RADAR signal emulator for signal processing test applications," *Int. Journal of Engineering Research and Applications*, vol. 4, pp. 362–367, feb 2014.

[5] A. Orduyilmaz, G. Kara, M. Serin, A. Yildirim, A. C. Gürbüz, and M. Efe, "Real-time Pulse Compression Radar Waveform Generation and Digital Matched Filtering," in *IEEE Radar Conf.*, 2015, pp. 426–431. doi: 10.1109/RADAR.2015.7131037 .

[6] Y. Zhang, Z. Wang, and J. Wang, "Integrated radar signal processing using FPGA dynamic reconfiguration," in *CIE Int. Conf. Radar*, 2016, pp. 1–4. doi: 10.1109/RADAR.2016.8059575 .

[7] F. Yinjuan, L. Yong, H. Qiongdan, and Z. Kunhui, "Design and analysis of LFM/Barker RF stealth signal waveform," in *IEEE 11th Conf. Ind. Electron. Appl.*, 2016, pp. 591–595. doi: 10.1109/ICIEA.2016.7603653 .

[8] M. Yang, J. Yang, Y. Hou, and C. Jin, "Implementation architecture of signal processing in pulse doppler radar system based on FPGA," *The Journal of Engineering*, no. 21, pp. 7335–7338, 2019.

[9] Y. Zhao, H. You, and Y. Zhang, "An FPGA-based direct sampling and digital processing system for wideband and narrowband radar signal," *J. Phys.: Conf. Ser.*, no. 1624, 2020.

[10] H. Wang, T. Shan, X. Qiao, and H. Cheng, "Research on the design of FPGA-based radar signal processing system," in *2021 7th International Conference on Condition Monitoring of Machinery in Non-Stationary Operations (CMMNO)*, 2021, pp. 36–39.

[11] A. Klilou, A. Arsalane, M. Najoui, and S. Belkouch, "An efficient and scalable parallel mapping of pulse-doppler radar signal processing chain on a multi-core DSP," *Microprocessors and Microsystems*, vol. 85, 2021.

[12] V. Wasik, L. Casadebaig, B. Gabard, B. Gigleux, C. Loic, Y. Paichard, and H. Jeuland, "Prototyping a radar sensor based on a xilinx RFSoC device for detect-and-avoid onboard small UAV," in *EuMW : EuMC +EuMIC + EURAD 2023*, sep 2023.

[13] J. Xu, "Design of high-speed radar signal processor based on FPGA," *Journal of Combinatorial Mathematics and Combinatorial Computing*, vol. 125, pp. 93–107, mar 2025.

[14] T. Kashiwa, Y. Asada, and T. Kobayashi, "Weather radar: Design-to-deployment using the NI platform," FURUNO, Tech. Rep., 2014.

[15] N. Instruments, *User Manual and Specifications NI 5791R RF Transceiver Adapter Module*, may 2013.

[16] ——, *NI FlexRIO FPGA Module Installation Guide and Specifications*, oct 2011.

[17] M. A. Richards, *Fundamentals of Radar Signal Processing*. New York: McGraw-Hill, 2005.

[18] N. Instruments, "The NI LabVIEW high-performance FPGA developer's guide," National Instruments, Tech. Rep., feb 2014.

[19] I. R. Ivić, D. S. Zrnić, and S. M. Torres, "Whitening in range to improve weather radar spectral moment estimates. part I: Formulation and simulation," *J. Atmos. Ocean. Technol.*, vol. 20, pp. 1422–1448, nov 2003. doi: 10.1175/1520-0426(2003)020¡1433:WIRTIW¿2.0.CO;2 .

Santiago Elian Mallerman received his Telecommunications Engineering degree (2022) from the National University of Río Negro (UNRN) and a Master's degree in Engineering (2024) from Instituto Balseiro (IB), Argentina. He is currently a Teaching Assistant at both institutions. His research interests encompass weather radar systems, digital signal processing, and digital electronics, with a particular emphasis on applying signal processing techniques to communication systems.

Juan Pablo Pascual received his Eng. (2006) and Ph.D. (2014) degrees at the Universidad Nacional de La Plata (UNLP), Argentina. He worked in the design and FPGA implementation of control strategies for Electric Arc Furnaces in Tenaris (2008-2010). His postdoctoral experience include a research visit to the Instituto Balseiro in Bariloche (2015). He is currently a Professor at the Instituto Balseiro, a Researcher at the CONICET and a Researcher at the CNEA, Argentina. His research interests focus on statistical signal processing in radar, digital communication and agriculture technology.

Jorge Cogo received his Engineering degree (2008) and Master's degree (2016) from the Universidad Nacional de La Plata (UNLP), Argentina. He worked on the design and implementation of GNSS receivers for space applications, mainly as a software engineer (2008-2017). He is currently a professor and researcher at the Universidad Nacional de Río Negro. His research interests focus on statistical signal processing in radar, digital communications, GNSS and agriculture technology.