Real-Time FPGA-based Range Radar Demonstrator with a High-Definition Multimedia Interface Output

Nikola Ž. Petrović
Department of Electronics
School of Electrical Engineering, University of Belgrade
Belgrade, Serbia
p.z.nikola@etf.bg.ac.rs

Vladimir M. Milovanović
Department of Electrical Engineering & Computer Sciences
Faculty of Engineering, University of Kragujevac
Kragujevac, Serbia
vlada@kg.ac.rs

***Abstract*—This paper presents a real-time, FPGA-based range radar demonstrator with a High-Definition Multimedia Interface (HDMI) output. The primary aim of the proposed Range Radar demonstrator is to offer a real-time, cost-effective, and user-friendly visualizing solution for radar systems with FMCW-based outputs. The developed demonstrator comprises three main components: the signal pre-processing block, the digital signal processing block, and the HDMI-based transmitter. The functionality of the proposed design was demonstrated using TI's AWR2243 Single-Chip 76 to 81-GHz FMCW Transceiver front end. The digital back end implementation was carried out on the Digilent's Nexys Video FPGA board. Real-time plotting of raw ADC signals and FFT-processed signals was successfully achieved on a Full HD monitor with a refresh rate of 60 Hz.**

***Index Terms*—Chisel, Demonstrator, FPGA, FMCW, HDMI, Radar, Real-time**

I. Introduction

As the automotive industry advances toward developing semi- and fully autonomous vehicles, radar systems are emerging as essential technology for enabling autonomous and intelligent functions in modern vehicles, including advanced driver assistance systems (ADAS) [1], [2]. While cameras are intuitive for ADAS due to their similarity to human vision [3], they require substantial processing power. Furthermore, low light and poor weather conditions present additional challenges for camera-based systems. In contrast, radar sensors can operate effectively in various lighting conditions (day or night) and under severe weather (rain, sun, fog, blizzard). Radar signals do not require a direct line of sight to the target and can penetrate optically opaque materials such as foliage, buildings, soil, or even human tissue. For automakers, radar systems offer an aesthetic advantage over cameras, as they can be seamlessly concealed behind a vehicle's plastic bumper.

These benefits of radar sensors were quickly acknowledged, making automotive radars a crucial tool in reducing the frequency and severity of road accidents and a key sensor for collision avoidance as well as pedestrian and cyclist detection. Both short-range and long-range radar sensors continuously monitor a vehicle's surroundings in real-time, delivering essential data to ADAS systems to enhance driver safety and comfort. From short-range (SRR) to long-range (LRR) applications, automotive radars provide high versatility and precision, serving as fundamental components for assisted driving at Levels L0/L3 up to fully autonomous L4/L5 levels of driving.

Today, automotive radar is integrated into many high-end vehicles to support key safety and comfort features, including Adaptive Cruise Control (ACC), Autonomous Emergency Braking (AEB), Blind Spot Detection (BSD), Cascaded Imaging Radar (IMR), Front and Rear Cross-Traffic Alert (FCTA/RCTA), Lane Change Assistance (LCA), Parking Assist (PA), 360° Radar Perception, and Reverse Autonomous Emergency Braking (R-AEB) [1]–[5].

With the increasing applications of radar systems in the automotive industry, the automotive radar market is expanding, widening the gap between available radar systems and real-time radar demonstrators. This paper presents a real-time FPGA-based range radar demonstrator with HDMI output to help close that gap. The goal of this work was to develop a user-friendly, real-time system for processing and visualizing radar signals. The paper is structured as follows: Section II describes the Demonstrator Architecture, Section III covers the experimental setup, and Section IV concludes the paper.

II. System Architecture

Block diagram of the proposed demonstrator is shown in Fig. 1. By functionality, Demonstrator can be splitted in three main parts: Pre-Processing Block, Digital Signal Processing Block and HDMI Transmitter Block.

A. Pre-Processing Block

The Pre-processing block includes data deserializers, a clock domain crossing (CDC) component, and a Word Alignment Block. Clock domain crossing is achieved through an asynchronous AXI-Stream FIFO, while the deserializer modules are implemented using AMD SelectIO IPs, operating with a Double Data Rate (DDR) clock and a deserialization factor of 1:8.

Since the AWR2243 is capable of transmitting complex data with a width of 12, 14, or 16 bits for both the real and imaginary components via its serial interface, the 8-bit deserialized output from the Data Deserializer block needs to be

This work was financially supported by the Ministry of Science, Technological Development and Innovation of the Republic of Serbia under contract number: 451-03-65/2024-03/200103.

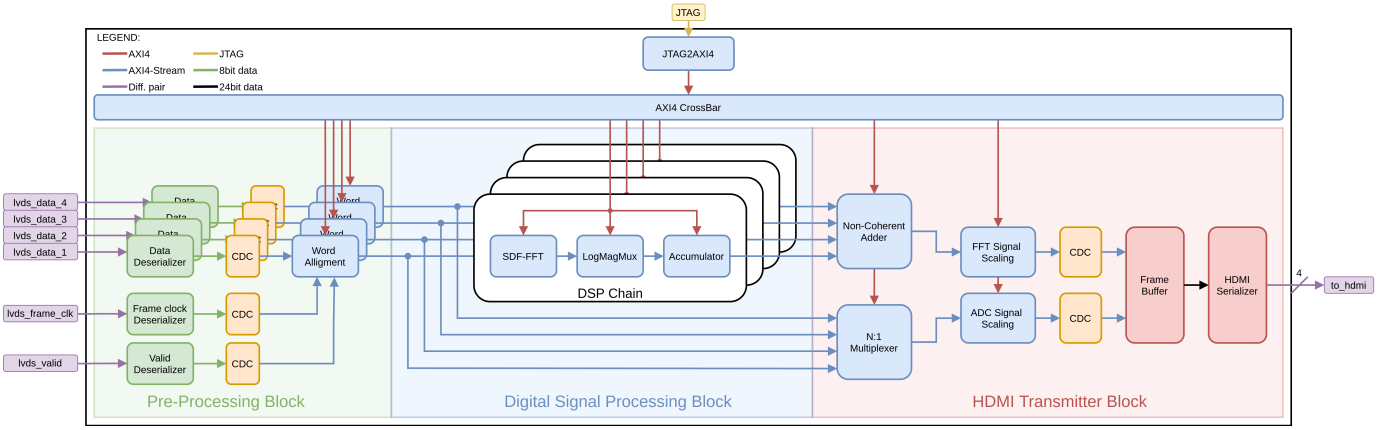

Fig. 1. Block diagram of the proposed Range Radar Demonstrator. Blocks of the same color belongs to the same clock domain. CDC (Clock Domain Crossing) blocks are realized as Asynchronous FIFOs. PLL generating different clocks was omitted from the block diagram.

correctly formatted. To accomplish this, the control registers of the Word Alignment Block must be configured with details on how the radar sensor's ADC transmits data. This information ensures the proper alignment of the deserialized 8-bit data into 32-bit complex data, with 16 bits allocated to the real and imaginary parts of the signal. For instance, if the ADC sends only real data values with the MSB first, an appropriate value must be set in the Word Alignment Block's control registers. The Word Alignment Block supports a wide range of Texas Instruments AWR2243 data configuration options. In addition to data alignment, it also performs bitslip detection and correction using the frame clock and data valid signals.

B. Digital Signal Processing Block

The Digital Signal Processing (DSP) chain includes a Single-path Delay-Feedback (SDF) FFT, a Log-Magnitude Multiplexer (LogMagMux), and an Accumulator module. There is a DSP chain for each radar receiver, with each chain's output connected to the input of a Non-Coherent Adder. The Non-Coherent Adder can be configured either to sum all inputs or to output a single selected input.

The SDF-FFT used [6] in proposed design offers extensive parameterization options. The complex FFT input and output are composed of fixed-point real and imaginary parts, each with configurable total width and fractional bit count (or binary point location). Further parameters include the SDF FFT scheme (radix-2, radix-4, or radix-2^2) and a choice between decimation-in-time (DIT) or decimation-in-frequency (DIF) for each scheme. The number of selected radix stages is also configurable, directly determining the maximum sample count (i.e., FFT size) that the FFT instance can process. Additionally, while the FFT size is set at compile-time, the instance can handle smaller FFT sizes through runtime configuration using control registers. Block diagram of the used SDF-FFT can be found in [6].

The block diagram for the Log-Magnitude Multiplexer (LogMagMux) module can be found in [7], and it shares the same parameterized interface as the SDF-FFT. The LogMagMux computes the squared magnitude of the input complex signal, as well as the magnitude and its log2 value. To calculate the magnitude, the LogMagMux can replace the complex square root operation with the JPL approximation method [8].

The interface of the Accumulator block has the same parametrized input and output data paths, with configurable total width and fractional bit count (or binary point location), similar to the previously mentioned blocks. A block diagram of the used Accumulator is available in [9]. Both the depth of the accumulator (which matches the size of the FFT) and the maximum number of accumulated FFT windows are parameterized. Additionally, once the maximum number of accumulated FFT windows is set at compile time, the Accumulator instance can accumulate any smaller number of FFT windows at runtime through appropriate control registers.

C. HDMI transmitter Block

Since the size of the FFT window can be configured at runtime and the number of Radar ADC samples can vary, it is necessary to resample both the FFT and ADC data before transferring them to the Frame Buffer in order to display the complete FFT window (or all samples of the Radar chirp) on the monitor. The resampling and scaling of the data are performed in the ADC and FFT Scaling Blocks, with their block diagrams illustrated in Fig. 2.

The Trigger block is utilized to capture the rising or falling edge of the ADC signal. The type of edge and the trigger level can be configured by writing the appropriate values to its control registers. On the other hand, the DataCounter block facilitates the transfer of FFT data to the Frame Buffer. Both the Trigger Block and the DataCounter block are continuously prepared to receive input.

The Scaler block is employed to scale the value of the input signal in order to fit data better on the Frame buffer plot. Input data can be either divided or multiplied by 2^N ($N \in [1, .., 15]$) by configuring the control register of the Scaler block.

The interpolation and decimation factors of the Interpolator and Decimator blocks can be configured at runtime through the

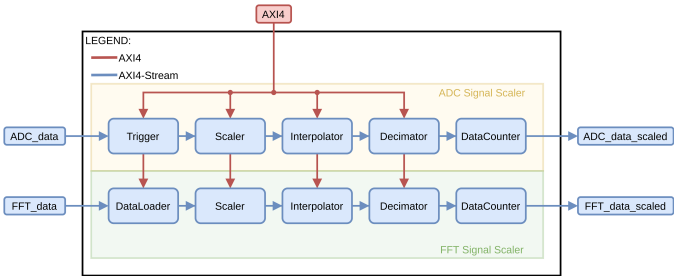

Fig. 2. Block diagram of the ADC and FFT scaling blocks.

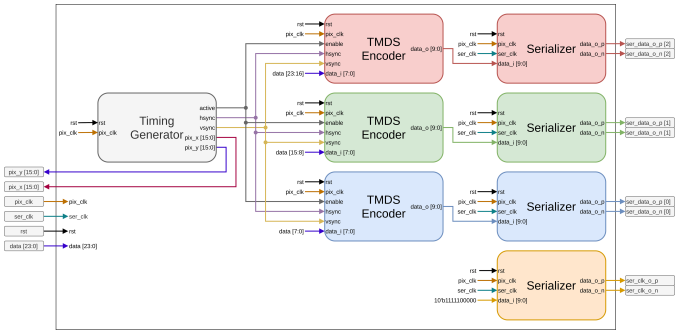

Fig. 3. Block diagram of the HDMI Serializer Block. Pair of TMDS Encoder and Serializer is used for each RGB color and additional Serializer is used to generate LVDS clock signal needed for HDMI.

appropriate control registers. Because both the Interpolation and Decimation blocks are dynamically adjustable, the Data-Counter block ensures that the number of data samples sent to the Frame Buffer matches the x-axis size of the Frame Buffer grid, while any excess samples are discarded. This approach guarantees that the Frame Buffer consistently receives the exact number of data samples required for accurate plotting.

The Frame Buffer is responsible for plotting both the complex time-domain signal from the radar ADC and the processed FFT signal by mapping them to the appropriate RGB values. It generates two distinct grids: one for the complex time-domain signals and another for the frequency domain signals. The Frame Buffer then sends the RGB data (8 bits for each color) to the HDMI Serializer block. The HDMI Serializer block, illustrated in Fig. 3, includes a Timing Generator, Transition-Minimized Differential Signaling (TMDS) Encoders (one for each RGB color), and Serializers.

TMDS is essential for the serial transmission of high-speed digital signals, as its encoding algorithm reduces electromagnetic emissions, ensures DC balance on the wires, and facilitates reliable clock recovery. The encoding process aims to minimize signal transitions to reduce interference between channels while maintaining enough transitions for effective clock recovery. By keeping the counts of ones and zeros on the line nearly equal, the DC balance aspect of the encoding enhances the noise margin.

The Timing Generator generates the control signals needed by the TMDS Encoder Block, including the vertical synchronization (vsync) and horizontal synchronization (hsync) signals. Furthermore, it also provides the Frame Buffer with the (x, y) pixel location information as a secondary function.

The HDMI Serializer utilizes four Serializer Blocks: three for each RGB color and one for the clock signal serialization. These Serializer Blocks were implemented as AMD IP OSERDESE2 using a Master-Slave DDR configuration. To achieve Full HD 60Hz video output, the frequency of the pix_clk was set to 148.5 MHz, while the frequency of the ser_clk for the OSERDESE2 was set to 742.5 MHz.

III. Experimental Results

The proposed Real-Time Range Radar Demonstrator was implemented on the Digilent Nexys Video FPGA board. For JTAG communication with the FPGA and to write to the control register of the demonstrator, a C232HM USB 2.0 Hi-Speed to MPSSE Cable was utilized. A Python script, based on the pyftdi library, was employed to interface with the JTAG via the C232HM USB cable. The Texas Instruments AWR2243 Single-Chip 76 to 81-GHz FMCW Transceiver served as the Radar Transceiver, and the FPGA's HDMI output was connected to a Dell Full HD Monitor. The experimental setup is illustrated in Fig. 4.

Three different targets were utilized in this experimental

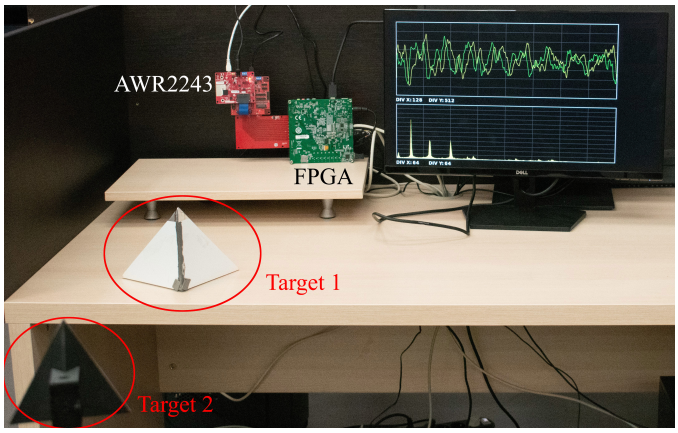

Fig. 4. Experimental setup for the proposed Real-Time Range Radar Demonstrator. Third radar target is farthest and is not shown in the image.

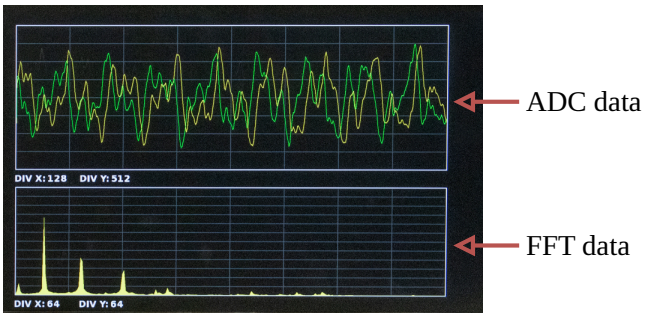

Fig. 5. The HDMI output of the proposed demonstrator displays complex ADC data in the upper section of the monitor, while FFT data is shown in the lower section.

setup. Two of these targets are visible in Fig. 4, while the third, which is the farthest away, is omitted from the image. Corner radar reflectors were used as the targets. The lower graph displayed on the monitor (Fig. 5) shows the FFT magnitude of these targets. The upper graph depicts the complex ADC data, with the green signal representing the real part and the yellow signal indicating the imaginary part of the data. The DIV X and DIV Y values correspond to the x and y divisions of the graph, which can be used to calculate the number of samples plotted (DIV X) or the amplitude of the signal (DIV Y). The DIV X value updates in response to changes in the interpolation and decimation factors, while the DIV Y value is determined by the scaling factor of the Scaler Block.

To ensure the FFT signal fits within the plot diagram, its value was divided by 2 by adjusting the relevant control registers in the Scaler Block, while the ADC data was divided by 8 in the same manner. From the Table I it can be seen that only the first 512 samples of the FFT would be plotted. Given that the FFT size is 1024, with an interpolation factor of 6 and a decimation factor of 2, the new number of data samples after the Decimator Block would effectively be 3072. The DataCounter truncates the first 1536 input samples (which corresponds to the x-axis size of the Frame Buffer graphs), meaning only the first half of the interpolated signal is sent to the Frame Buffer, resulting in the effective plotting of just the first 512 FFT samples. This is confirmed in Fig. 4, where the FFT DIV X is equal to 64 and there are a total of 8 divisions. The configurations of the demonstrator are summarized in Table I.

TABLE I
DEMONSTRATOR CONFIGURATION SUMMARY

FFT	
FFT Size	1024
DIF/DIT	DIF
Run-time configurable	true
Input data type	complex
LogMagMux	
Operation	magnitude
Accumulator	
Depth	1024
Numer of Frames	128
Non-Coherent Adder	
Operation	add all
N:1 Multiplexer	
Pass	channel 1
ADC Signal Scaling	
Trigger treshold	768
Triger edge type	rising
Scaler factor	div by 8
Interpolation factor	3
Decimation factor	2
FFT Signal Scaling	
Scaler factor	div by 2
Interpolation factor	6
Decimation factor	2

The Texas Instruments AWR2243 was configured to have 1024 ADC samples, providing a maximum range of approximately 20 meters in Complex2x mode of operation. Only one transmitter was activated, while all four receiver channels were enabled. A summary of the Texas Instruments AWR2243 configuration can be found in Table II.

TABLE II
TEXAS INSTRUMENTS AWR2243 CONFIGURATION

Parameters	
Tch_active [us]	61.2
Tch_idle [us]	7.5
Frequency Slope [MHz/us]	63
N_tx	1
N_rx	4
Fs [MHz]	18.5
V_res [m/s]	0.43
D_res [cm]	4.3
ADC start time [us]	5.34
TX start time [us]	0
Ramp end not sampled [us]	0.5
IFmax [MHz]	16.65
Total swept BW [GHz]	3.86
N_chirps	64
fb_min [kHz]	40
D_min [cm]	9.52
Active chirping time [ms]	4.4
V_min [m/s]	0.43
Azimuth_res[deg]	28.66
N_adc	1024
Radar_cube_raw_data [kB]	1024
Throughput [Gbits/s]	2.38
V_max [kmph]	49
D_max [m]	39

IV. CONCLUSION

The Real-Time Portable Radar Demonstrator with a Digital Signal Processing Chain and HDMI output, was developed and implemented on the Digilent Nexys Video FPGA board. The ADC signals from the TI AWR2243 were processed, and both the ADC and FFT signals were displayed on a 60Hz Full HD monitor. While other radar demonstrators exist [10]–[13], to our knowledge, this is the first Range Radar Demonstrator that is fully realized in hardware, with capabilities for both real-time digital signal processing and plotting.

REFERENCES

[1] M. Stolz, M. Li, Z. Feng, M. Kunert and W. Menzel, "High resolution automotive radar data clustering with novel cluster method," 2018 IEEE Radar Conference (RadarConf18), 2018, pp. 0164-0168, doi: 10.1109/RADAR.2018.8378550.

[2] I. Bilik et al., "Automotive multi-mode cascaded radar data processing embedded system," 2018 IEEE Radar Conference (RadarConf18), 2018, pp. 0372-0376, doi: 10.1109/RADAR.2018.8378587.

[3] P. Swami, A. Jain, P. Goswami, K. Chitnis, A. Dubey and P. Chaudhari, "High performance automotive radar signal processing on TI's TDA3X platform," 2017 IEEE Radar Conference (RadarConf), 2017, pp. 1317-1320, doi: 10.1109/RADAR.2017.7944409.

[4] F. Arzur, M. Le Roy, A. Pérennec, G. Tanné and N. Bordais, "Hybrid architecture of a compact, low-cost and gain compensated delay line switchable from 1 m to 250 m for automotive radar target simulator," 2017 European Radar Conference (EURAD), 2017, pp. 239-242, doi: 10.23919/EURAD.2017.8249191.

[5] F. Fembacher, F. B. Khalid, G. Balazs, D. T. Nugraha and A. Roger, "Real-Time Synthetic Aperture Radar for Automotive Embedded Systems," 2018 15th European Radar Conference (EuRAD), 2018, pp. 517-520, doi: 10.23919/EuRAD.2018.8546620.

[6] V. M. Milovanović and M. L. Petrović, “A single-path delay feedback (SDF) fast Fourier transform (FFT) generator,” www.github.com/milovanovic/sdf-fft, accessed: 2024/10/20

[7] M. L. Petrović and V. M. Milovanović, ”On Efficient Hardware Realizations of Fixed-Point Complex Number Magnitude and its Logarithms.” 2023 10th International Conference on Electrical, Electronic and Computing Engineering (IcETRAN). IEEE, 2023.

[8] B. K. “Levitt and G. A. Morris, An improved digital algorithm for fast amplitude approximations of quadrature pairs,” DSN Progress Report 42, 1977, p. 40.

[9] V. M. Milovanović and M. L. Petrović, “Digital Accumulator written in Chisel HDL,” https://github.com/milovanovic/accumulator, accessed: 2024/10/20

[10] B. Nuss, A. Diewald, J. Schoepfel, D. Martini, N. Pohl and T. Zwick, ”76GHz OFDM Radar Demonstrator with Real-Time Processing for Automotive Applications,” 2020 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM), 2020, pp. 1-4, doi: 10.1109/ICMIM48759.2020.9299057.

[11] S. Turso, A. Konforta, T. Bertuch and P. Knott, ”An S-band portable radar demonstrator,” 2017 European Radar Conference (EURAD), 2017, pp. 215-218, doi: 10.23919/EURAD.2017.8249185.

[12] K. Savci, A. Y. Erdogan and T. O. Gulum, ”Software defined L-band noise radar demonstrator,” 2016 17th International Radar Symposium (IRS), 2016, pp. 1-4, doi: 10.1109/IRS.2016.7497330.

[13] A. S. Paine, K. J. Homer, J. C. Medley and P. G. Richardson, ”Real-time STAP hardware demonstrator for airborne radar applications,” 2008 IEEE Radar Conference, 2008, pp. 1-5, doi: 10.1109/RADAR.2008.4720860.