An FPGA-SoC-Based Arbitrary Waveform Generator with Runtime Configuration

Guillaume Hansen, Tianyang Fang, Mikhail Gromov, and Jafar Saniie
Embedded Computing and Signal Processing (ECASP) Research Laboratory (http://ecasp.ece.iit.edu)
Department of Electrical and Computer Engineering
Illinois Institute of Technology, Chicago, IL, U.S.A.

Abstract*— Programmable signal generators are essential in laboratories and educational environments, yet conventional designs are often constrained by rigid waveform synthesis and fixed hardware structures. Modern FPGA-based architectures overcome these limitations by enabling real-time configurability, delivering a flexible and adaptable platform that significantly enhances traditional signal generation capabilities. This paper presents a runtime-configurable signal generator implemented on a System-on-Chip (SoC) platform. The design employs a hybrid hardware–software architecture that integrates the processing system with programmable logic, supporting both predefined and arbitrary waveform generation, including frequency sweeps and dynamic parameter reconfiguration. A custom signal conditioning stage provides programmable amplitude and offset control while maintaining compatibility with standard 50-ohm laboratory loads. User interaction is facilitated through an Ethernet-based graphical interface, enabling remote configuration and real-time definition of custom waveform shapes. Experimental results confirm that the proposed system offers efficient, flexible, and runtime-configurable signal generation, making it well-suited for research, testing, and educational applications.

Keywords*—Arbitrary waveform generator, FPGA-SoC, Red Pitaya, waveform synthesis, digital signal generation, embedded instrumentation.

I. Introduction

The generation of precise and configurable electronic signals is fundamental to experimental research, device testing, and engineering education. Signal generators enable controlled excitation of electronic systems and are widely used in laboratories for the evaluation and characterization of circuits, sensors, and communication systems [1]. Modern applications increasingly require flexible waveform generation, frequency sweeping capabilities, and remote control interfaces to support automated testing and data acquisition workflows. As a result, programmable and arbitrary waveform generators have become essential tools in many research environments [2]–[5], [7].

Traditional signal generation solutions typically rely on dedicated laboratory instruments or microcontroller-based implementations. While commercial instruments offer high precision and advanced functionality, they are often limited in their ability to be customized for variaty experimental needs. Conversely, microcontroller-based signal generators offer flexibility but are typically constrained in waveform complexity, generation bandwidth, and runtime configurability [5] and [6]. Recent research has explored the use of field-programmable gate arrays (FPGAs) and digital direct synthesis (DDS) techniques to address these limitations, enabling higher-frequency operation and improved signal fidelity [4], [8], [9], [10], [11].

Despite these advances, many FPGA-based signal generation systems remain relatively rigid, with waveform generation pipelines that are difficult to reconfigure dynamically during operation. Furthermore, integrating user-friendly control interfaces and remote configuration capabilities remains a challenge in many existing designs. The increasing availability of System-on-Chip (SoC) platforms that combine programmable logic with embedded processors provides new opportunities to develop flexible signal generation systems that support both high-performance hardware processing and software-driven configurability.

In this work, we present a runtime-configurable signal generation architecture implemented on a System-on-Chip platform. The system leverages the RedPitaya STEMLab board, which integrates a Zynq-based processing system (PS) with programmable logic (PL), enabling a hybrid hardware–software architecture for waveform synthesis and signal generation. The processing system is responsible for waveform parameter computation and user interaction, while the programmable logic performs real-time waveform streaming to the digital-to-analog converter (DAC). This architecture enables efficient generation of both predefined and arbitrary waveforms with minimal latency.

To facilitate flexible user interaction, the system includes a graphical user interface (GUI) that allows users to configure waveform parameters, including frequency, amplitude, offset, and waveform type. The system also supports arbitrary waveform generation through direct user input, allowing custom signal shapes to be defined graphically and transmitted to the hardware in real time. Communication between the user interface and the signal generator is implemented through an Ethernet-based control interface, enabling remote configuration and integration with automated experimental setups.

The primary contributions of this work include:

1. A hybrid PS–PL signal generation architecture implemented on an FPGA-based SoC platform.
2. A runtime-configurable waveform generation mechanism supporting both standard and arbitrary waveforms.
3. A dual-port memory architecture that enables efficient streaming of waveform samples to the DAC.

979-8-3315-6026-3/26/$31.00 ©2026 IEEE

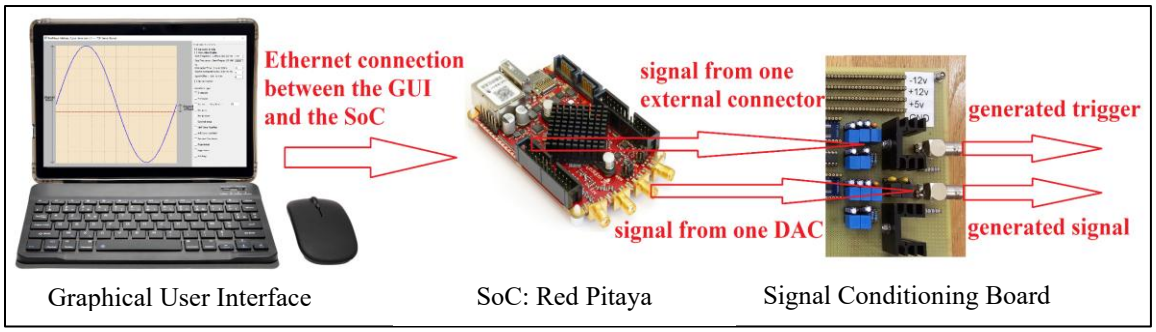


Fig.1. Key Components of the System Architecture

4. An Ethernet-based remote control interface enabling flexible integration into automated testing environments.
5. Experimental validation of the system demonstrates waveform generation up to 100 kHz with configurable amplitude, offset, and frequency sweep capabilities.

The overall system architecture that enables these capabilities is illustrated in Fig. 1.

II. System Design

A. Overall Architecture

The proposed signal generator is implemented on a System-on-Chip (SoC) platform that integrates a processing system (PS) and programmable logic (PL) on a single device. The implementation uses the RedPitaya STEMLab 125-14 board, which incorporates a Xilinx Zynq SoC featuring an ARM-based processor coupled with FPGA fabric and high-speed digital-to-analog converters (DACs).

The architecture follows a hybrid hardware–software design in which waveform synthesis and control are handled by the processing system, while high-speed waveform streaming and timing-critical operations are implemented in programmable logic. This separation allows computationally flexible operations to be performed in software while maintaining deterministic timing and high throughput in hardware.

Unlike conventional FPGA-based signal generators that implement fixed waveform synthesis pipelines entirely in hardware, the proposed architecture separates waveform computation and waveform streaming between the processing system and programmable logic. This partitioning allows waveform parameters and signal shapes to be modified at runtime without re-synthesizing FPGA logic.

The system consists of four primary subsystems:

1. **Processing System (PS)** – responsible for waveform parameter computation, communication with the user interface, and system configuration.
2. **Programmable Logic (PL)** – implements the waveform generation pipeline and manages high-speed DAC data streaming.
3. **Waveform Memory Subsystem** – stores waveform samples that define one period of the generated signal in the Dual-Port RAM on the FPGA.
4. **Signal Conditioning Hardware** – adapts the DAC output signal to meet required voltage levels and output impedance.

Communication between the PS and PL is implemented through the AXI bus, enabling efficient transfer of waveform samples and configuration parameters. User interaction is performed through a graphical interface running on a host computer, which communicates with the embedded system via an Ethernet connection. A high-level representation of the system architecture is shown in Fig. 2.

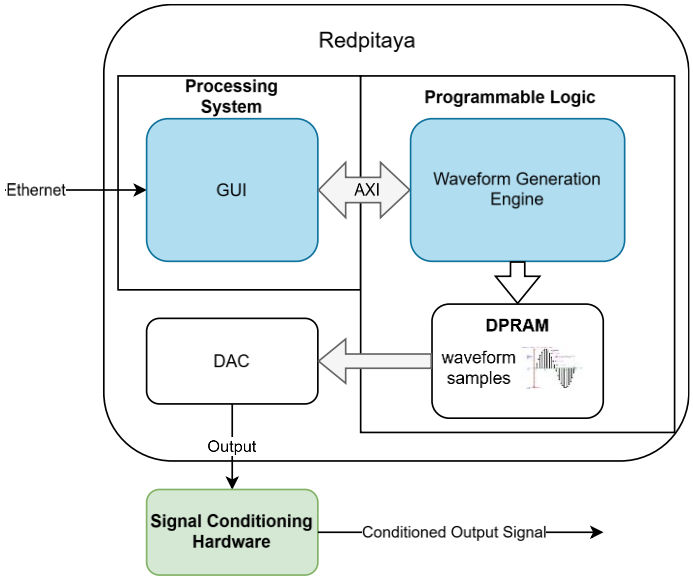


Fig. 2. High-level representation of the system architecture

B. Waveform Generation Engine

The waveform generation engine is implemented in the programmable logic of the SoC and is responsible for producing a continuous stream of digital samples that are sent to the DAC. To support flexible waveform generation, the system represents each signal period as a sequence of discrete samples stored in a waveform memory. The internal structure of the waveform generation engine is shown in Fig. 3.

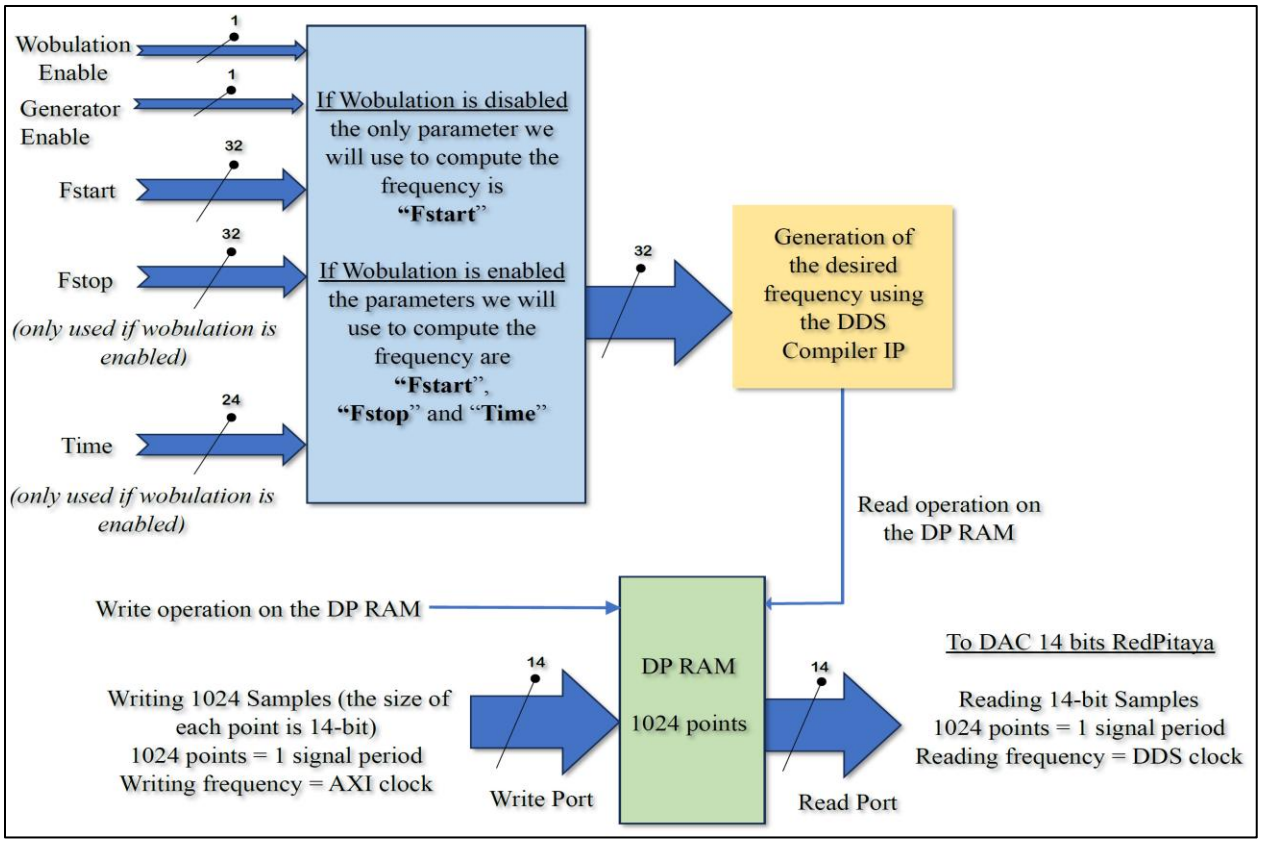


Fig. 3. Waveform generation engine PL design

In this implementation, each waveform period is defined by 1024 samples, which provides sufficient resolution for generating smooth waveforms while maintaining efficient memory usage. A dual-port RAM (DPRAM) is used to store the waveform samples. The write port of the memory is connected to the processing system through the AXI interface, allowing waveform data to be written from software. The read port continuously streams samples to the DAC through the waveform generation pipeline. The output signal frequency is controlled by adjusting the rate at which waveform samples are read from memory. This is accomplished using a configurable Direct Digital Synthesis (DDS) frequency generator implemented in the programmable logic. The DDS module generates the sampling clock that determines the read rate of waveform samples. Since one waveform period is represented by 1024 samples, the sample read frequency must be 1024 times higher than the desired output signal frequency. For example, generating a 100 kHz waveform requires a sample clock of 102.4 MHz. The RedPitaya's DAC supports sampling frequencies up to 125 MHz, which comfortably satisfies this requirement.

A custom register management module controls the waveform generation parameters, including frequency, waveform enable/disable state, and frequency sweep settings. These parameters are written by the processing system and stored in control registers within the programmable logic. This architecture enables continuous waveform streaming while allowing waveform parameters to be modified dynamically without interrupting signal generation.

C. *Runtime Configuration*

The runtime configuration of the signal generator is managed by the processing system running on the ARM processor of the SoC. The processing system is responsible for receiving user commands, computing waveform samples, and updating the configuration registers of the programmable logic.

A lightweight Linux operating system is deployed on the processing system using the Buildroot framework, which provides a minimal software environment tailored for embedded applications. The operating system hosts a control application responsible for communication with the external user interface.

User interaction is performed through a graphical user interface (GUI) running on a host computer. Shown in Fig. 4., the GUI allows users to configure waveform parameters such as frequency, amplitude, offset, waveform type, and frequency sweep settings. Communication between the host computer and the embedded system is implemented using a TCP/IP connection over Ethernet.

When a waveform configuration command is received, the processing system performs the following steps:

1. Compute the digital samples describing one period of the waveform.
2. Transfer the waveform samples to the DPRAM through the AXI interface.

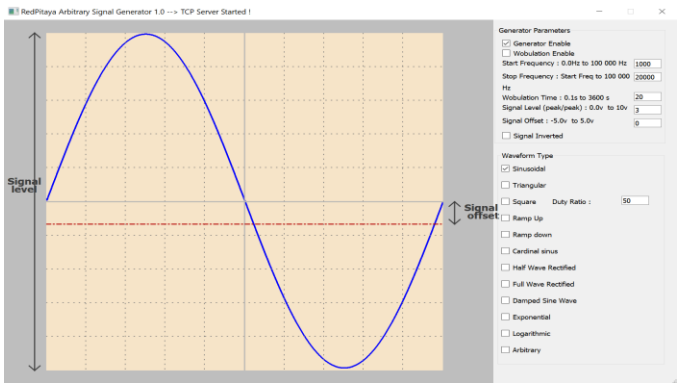


Fig. 4. The Graphical User Interface.

3. Update the control registers governing waveform generation parameters.
4. Trigger the waveform generator to begin streaming the updated signal.

For arbitrary waveform generation, users can define waveform shapes directly through the GUI interface. The resulting waveform samples are transmitted to the processing system and loaded into waveform memory in real time. This mechanism enables rapid waveform reconfiguration without requiring system reprogramming.

D. Signal Conditioning Hardware

The analog output produced by the DAC requires additional conditioning to provide the voltage range and output characteristics expected from a laboratory signal generator. To address this requirement, a custom signal conditioning board was developed. The signal conditioning hardware performs amplitude scaling, offset control, and output drive amplification to ensure compatibility with standard laboratory loads. A schematic of the signal conditioning board is shown in Fig. 5.

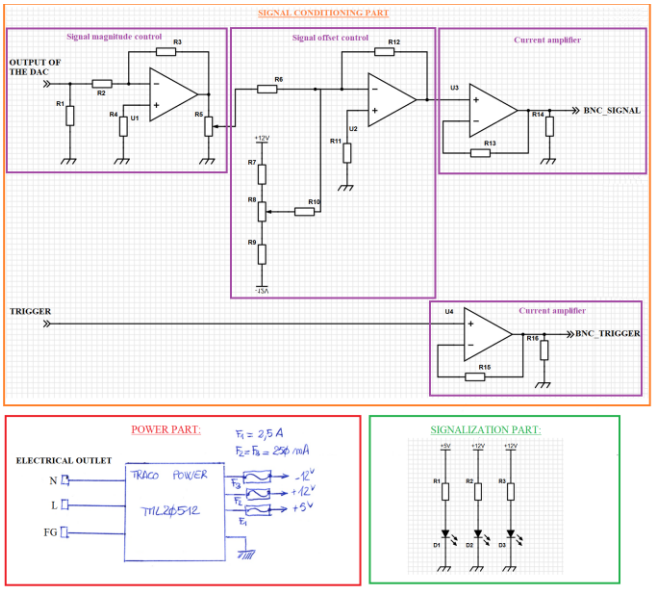


Fig. 5. Signal conditioning board schematic diagram.

Amplitude and offset adjustments are implemented using operational amplifier circuits combined with digitally controlled potentiometers. The potentiometers are controlled through an SPI interface connected to the programmable logic, allowing signal levels to be configured programmatically by the processing system.

The final stage of the signal conditioning circuit consists of a power operational amplifier that provides sufficient current drive capability for the output stage. This stage ensures compatibility with standard measurement equipment such as oscilloscopes and laboratory instruments. By separating waveform generation and analog signal conditioning into distinct subsystems, the architecture maintains flexibility in digital waveform synthesis while ensuring that the final analog output meets the electrical requirements of laboratory instrumentation.

III. Experimental Evaluation

A. Experimental Setup

The proposed signal generator was implemented and evaluated using the RedPitaya STEMLab 125-14 platform. The system consists of the Zynq-based SoC board, a custom signal conditioning module, and a host computer running the graphical user interface. Communication between the host computer and the embedded system is established through an Ethernet connection. The waveform generation engine implemented in the programmable logic streams digital samples to the on-board 14-bit DAC. The resulting analog signal is then processed by the signal conditioning board, which provides programmable amplitude scaling, offset adjustment, and output drive capability compatible with a standard 50-ohm load.

To evaluate the performance of the signal generator, the conditioned analog output was connected to a digital oscilloscope using high-impedance probes. The oscilloscope was used to observe waveform shapes, verify signal parameters, and confirm the correct operation of the waveform generation engine. The experimental setup used for the evaluation is shown in Fig. 6.

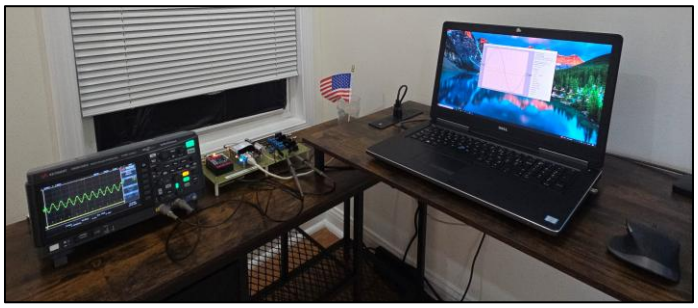


Fig. 6. Experimental setup used to evaluate the proposed waveform generator.

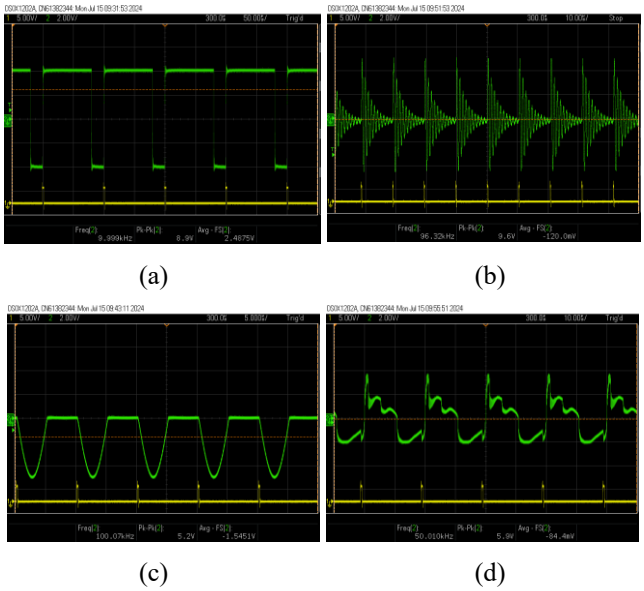
B. Generated Waveforms

The system was tested by generating a variety of waveform types supported by the signal generator (see Fig. 7). The waveform parameters were configured through the graphical user interface and transmitted to the embedded system through the Ethernet control interface.

These results demonstrate the capability of the waveform generation engine to synthesize both predefined and user-defined signals. In the arbitrary waveform mode, waveform samples defined by the user are transmitted to the processing system and written to waveform memory, allowing custom signal shapes to be generated in real time.

C. *Frequency and Waveform Operation*

The signal generator supports output frequencies ranging from near DC up to 100 kHz. This frequency range is determined by the waveform representation and DAC sampling rate used in the system.



(a) (b)

(c) (d)

Fig. 7. Examples of four generated signals using the system. (a): 10 kHz inverted square signal with a duty ratio of 80. (b): Damped sin Wave with a frequency sweep from 60 kHz to 100 kHz. (c): 100 kHz inverted Half Wave rectified signal with an amplitude of 10V peak to peak. (d): 50 kHz Arbitrary signal

In the proposed architecture, each waveform period is represented by 1024 samples stored in waveform memory. The output frequency is therefore determined by the rate at which samples are read from memory. To generate a waveform with frequency f_{out}, the waveform engine must stream samples at a rate of:

$$f_{sample} = 1024 \times f_{out} \quad (1)$$

For a maximum output frequency of 100 kHz, the required sampling rate is 102.4 MHz. The RedPitaya DAC supports sampling rates up to 125 MHz, which allows the system to operate within the supported frequency range while maintaining sufficient temporal resolution for waveform reconstruction. Experimental observations confirm that the generator produces stable signals across the supported frequency range and that waveform parameters configured through the graphical interface are correctly applied by the waveform generation engine.

D. *System Functionality*

In addition to waveform generation, several configurable signal features were verified experimentally. These include:

1. Waveform selection among multiple predefined signal types
2. Arbitrary waveform generation defined through the GUI
3. Adjustable signal amplitude and offset
4. Signal inversion capability
5. Frequency sweep functionality

The frequency sweep feature allows the output frequency to vary linearly between user-defined start and stop frequencies over a configurable time interval. This capability is useful for applications such as system identification and frequency response measurements.

The experimental results confirm that the hybrid PS–PL architecture enables reliable runtime configuration of waveform parameters while maintaining continuous waveform generation through the programmable logic.

IV. CONCLUSION

This paper presented a runtime-configurable signal generator implemented on an FPGA-based System-on-Chip platform. The proposed design leverages the hybrid architecture of the RedPitaya STEMLab board, combining the flexibility of an embedded processing system with the deterministic timing capabilities of programmable logic. In this architecture, waveform parameters and sample values are generated and configured by the processing system, while the programmable logic performs continuous waveform streaming to the on-board digital-to-analog converter.

The waveform generation engine utilizes a dual-port memory structure to store waveform samples representing one signal period. By controlling the sample streaming rate through a configurable DDS-based clock generation mechanism, the system can produce output signals across a wide frequency range while maintaining consistent temporal resolution. This architecture enables both predefined waveform generation and arbitrary waveform synthesis through software configuration.

Experimental evaluation demonstrated successful generation of multiple waveform types, including sinusoidal, square, rectified, and arbitrary signals defined through the graphical user interface. The system supports signal frequencies up to 100 kHz while providing configurable amplitude, offset, and frequency sweep functionality. The custom signal conditioning hardware ensures compatibility with standard laboratory measurement equipment by providing adjustable voltage levels and a 50-ohm output interface.

The results confirm that the proposed hybrid PS–PL architecture enables flexible runtime configuration while maintaining continuous waveform generation in programmable logic. The design demonstrates how modern FPGA-based SoC platforms can support software-defined instrumentation through the tight integration of embedded processing and hardware signal generation.

Future work may focus on extending the frequency range of the system, improving waveform resolution, and integrating additional modulation capabilities. The proposed architecture establishes a robust and extensible framework for

reconfigurable, customizable instrumentation platforms. Its modular design and scalability make it particularly well-suited for advanced research, rigorous testing, and pedagogical applications in engineering and scientific environments.

REFERENCES

[1] L. Limin and Suzhihua, "Function Signal Generator," 2013 Fourth International Conference on Digital Manufacturing & Automation, Shinan, China, 2013, pp. 623-625.

[2] Z. Zhengiun and W. Fuping, "The Specific Implementation of a Programmable Arbitrary Digital Signal Generator," 2020 Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC), Dalian, China, 2020, pp. 460-465.

[3] C. YunYun, "The Design of Virtual Function Signal Generator," 2009 International Conference on Energy and Environment Technology, Guilin, China, 2009, pp. 337-339.

[4] Z. Daodong, P. Yikai, and P. Hongping, "Design of DDS Signal Generator Based on FPGA," 2021 4th International Conference on Pattern Recognition and Artificial Intelligence (PRAI), Yibin, China, 2021, pp. 51-54.

[5] S. Vaněk, K. Kugler and P. Janů, "Signal Generator Controlled by Microcontroller," 2022 International Symposium ELMAR, Zadar, Croatia, 2022, pp. 31-36.

[6] M. Popa and A. S. Popa, "Programmable Signal Generator based on Advanced Tricore Microcontrollers," 2007 IEEE International Conference on Signal Processing and Communications, Dubai, United Arab Emirates, 2007, pp. 828-831.

[7] H. M. Gawas and M. J, "Design and development of an IoT based Signal Generator for Testing Equipments," 2024 Zooming Innovation in Consumer Technologies Conference (ZINC), Novi Sad, Serbia, 2024, pp. 151-156.

[8] H. Yang, S. -B. Ryu, H. -C. Lee, S. -G. Lee, S. -S. Yong and J. -H. Kim, "Implementation of DDS chirp signal generator on FPGA," 2014 International Conference on Information and Communication Technology Convergence (ICTC), Busan, Korea (South), 2014, pp. 956-959.

[9] Q. Zhang, P. Qu, S. Sun, E. Yu and W. Yan, "Design of Full Digital Signal Generator and FIR Digital Filter System Based on FPGA," 2022 41st Chinese Control Conference (CCC), Hefei, China, 2022, pp. 4501-4506.

[10] Z. Zhao, L. Wang, J. Chen, Z. Cai, Y. Lv and Y. Feng, "The design and implementation of signal generator based on DDS," 2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN), Guangzhou, China, 2017, pp. 920-923.

[11] G. Goavec-M'erou, J.-M Friedt, "Buildroot-based Red Pitraya (STEMLab) system development", 2024. [online]