Real Time Signal Processing with FPGA on AWS F1 in Wireless Communication

Poongodi S, Latha Doppalapudi, Kalpana Ganapuram, T. Jyoshna

[1]*Professor, Department of ECE, CMR Engineering College, Hyderabad, Telangana, India*

[2,3]*Assistant Professor, Department of ECE, CMR Engineering College, Hyderabad, Telangana, India*

[4]*Final Year Student, Department of ECE, CMR Engineering College, Hyderabad, Telangana, India*

***DOI**: https://doi.org/10.34293/sjsem.v1.i1.11*

Abstract

Real-time audio signal processing requires high computational performance, low processing latency, and accurate frequency response, particularly in applications such as voice-enabled systems, hearing aids, and embedded audio platforms. This paper presents a cloud-based hardware acceleration framework for real-time audio denoising using a Finite Impulse Response (FIR) filter implemented on Amazon Web Services (AWS) EC2 F1 FPGA instances. The proposed system effectively suppresses unwanted noise while preserving speech quality within the critical 0–4 kHz voice frequency range.

The FIR filter is developed using fixed-point C++ to achieve a balance between computational accuracy and efficient hardware resource utilization. Filter coefficients are designed using standard digital signal processing techniques to ensure stable operation and a linear phase response. The design is synthesized into FPGA hardware using Xilinx Vivado High-Level Synthesis (HLS), enabling a deeply pipelined architecture for continuous audio sample processing. Compared with conventional CPU-based implementations, the FPGA-based solution significantly reduces processing latency while improving throughput.

By utilizing cloud-hosted FPGA resources, the proposed framework removes the need for dedicated hardware, providing scalability, remote accessibility, and cost-effective deployment through a pay-as-you-use model. Experimental results demonstrate that cloud-based FPGA acceleration offers an efficient and practical solution for meeting the demanding performance requirements of real-time digital audio processing in both research and industrial applications.

Keywords

Wireless Audio Processing, FPGA Acceleration, Low-Latency Communication, Speech Enhancement

1. INTRODUCTION

Modern wireless communication systems and intelligent audio applications require real-time digital signal processing with high accuracy, low latency, and reliable performance. Applications such as Voice over IP (VoIP), smart assistants, hearing enhancement devices, and adaptive wireless receivers continuously process streaming audio and therefore demand fast, consistent, and high-quality signal processing. In real-world environments, audio signals are often affected by background noise, interference, and environmental distortions, making efficient filtering techniques essential for maintaining speech quality.

Finite Impulse Response (FIR) filters are widely used for speech signal processing because they provide inherent stability, linear phase characteristics, and are well suited for hardware implementation. However, implementing high-order FIR filters for real-time applications requires significant computational resources. Traditional CPU-based implementations process data sequentially, which can increase latency and introduce timing variations caused by cache access delays and operating system scheduling. These limitations reduce their suitability for applications where predictable, low-latency performance is critical.

Field Programmable Gate Arrays (FPGAs) address these challenges by enabling parallel execution of multiple operations simultaneously. Their architecture supports deterministic timing and high throughput by mapping multiply-accumulate operations directly onto dedicated DSP resources. This parallel processing capability allows pipelined execution with predictable latency, making FPGAs an excellent choice for real-time signal processing. However, conventional FPGA deployment requires dedicated hardware, specialized development environments, and ongoing maintenance, which can increase deployment complexity and cost.

Cloud-based FPGA platforms, such as Amazon Web Services (AWS) EC2 F1 instances, provide an efficient alternative by offering FPGA resources on demand without requiring physical hardware management. These platforms combine the computational advantages of FPGA acceleration with the flexibility and scalability of cloud computing. Although cloud-hosted FPGAs have been successfully applied in areas such as machine learning and high-performance computing, their use in real-time wireless signal processing has received relatively little attention, particularly in terms of latency consistency and hardware resource efficiency.

This paper presents a cloud-based FPGA-accelerated framework for real-time wireless communication applications. The proposed system implements a fixed-point optimized FIR filter using High-Level Synthesis (HLS) and deploys it on AWS EC2 F1 infrastructure. A deeply pipelined processing architecture enables continuous streaming of 16-bit PCM audio samples with minimal latency. The framework also supports cloud-based deployment, remote management, system monitoring, and scalable hardware acceleration for real-time audio processing.

The remainder of this paper is organized as follows. Section II reviews related work on FPGA-based digital signal processing and cloud acceleration. Section III describes the proposed architecture and mathematical formulation. Section IV explains the implementation on AWS EC2 F1. Section V presents the experimental results and performance analysis, while Section VI concludes the paper.

2. LITERATURE REVIEW

Audio signal processing plays an important role in applications such as telecommunications, multimedia systems, hearing aids, and smart devices, where maintaining clear and accurate signals is essential. Among the available digital filtering techniques, Finite Impulse Response (FIR) filters are widely preferred because of their stability and linear phase response. These characteristics help preserve the original shape of speech signals, making FIR filters well suited for real-time audio enhancement.

Traditionally, audio filtering has been performed using Digital Signal Processors (DSPs) or general-purpose CPUs. While software-based implementations are flexible and easy to develop, they often struggle to satisfy the strict timing requirements of real-time applications. Sequential execution limits processing speed, increases latency, and may lead to higher power consumption. These challenges have encouraged researchers to explore hardware-based solutions that provide faster and more predictable performance.

Field Programmable Gate Arrays (FPGAs) have become a popular choice for real-time signal processing because of their highly parallel architecture. In FPGA-based FIR filter implementations, multiple multiply-accumulate (MAC) operations can be executed simultaneously, resulting in significantly higher throughput than software-based approaches. Pipelined architectures further improve performance by allowing continuous processing of incoming audio samples with very low latency. Previous studies have consistently shown that FPGA implementations outperform CPU-based solutions in terms of processing speed, energy efficiency, and deterministic execution.

Despite these advantages, conventional FPGA development requires dedicated hardware, knowledge of hardware description languages, and regular maintenance, making the overall development process more complex. The introduction of High-Level Synthesis (HLS) has simplified FPGA programming by allowing developers to design hardware using high-level languages such as C and C++, reducing both development time and implementation complexity.

To address the limitations of physical FPGA deployment, cloud service providers such as Amazon Web Services (AWS) introduced FPGA-enabled EC2 F1 instances. These cloud-based FPGA platforms provide on-demand hardware acceleration without requiring users to own or manage dedicated FPGA hardware. Although cloud-hosted FPGAs have been widely adopted for applications such as machine learning and high-performance computing, their use in real-time audio signal processing has not been explored extensively.

This work addresses that gap by combining FIR filtering, FPGA acceleration, and cloud deployment into a unified framework. The proposed system implements a fixed-point FIR filter on AWS EC2 F1 using High-Level Synthesis (HLS) and employs a pipelined architecture to achieve low-latency, real-time audio processing. By integrating hardware parallelism with cloud scalability, the framework provides an efficient, flexible, and practical solution for modern speech enhancement and audio processing applications.

3. PROPOSED METHODOLOGY

To address the limitations of conventional software-based noise filtering systems, such as high latency, sequential processing, limited throughput, and increased power consumption, this work proposes a cloud-based real-time audio signal processing framework. The system combines the parallel processing capabilities of a Field Programmable Gate Array (FPGA) with the flexibility and scalability of Amazon Web Services (AWS) EC2 F1 instances. The overall architecture consists of five major stages: Audio Input, Audio Preprocessing, FPGA Acceleration on AWS EC2 F1, HLS-Based FIR Filtering, and Processed Audio Output, as shown in **Fig. 1**.

The processing pipeline begins by capturing 16-bit signed PCM audio samples from live sources such as microphones or streaming devices. For testing and evaluation, stored audio files can also be used as input. Since raw audio signals often contain DC offsets and variations in amplitude, a preprocessing stage is included to normalize the signal and remove unwanted bias. The audio samples are then converted into a fixed-point representation that is

suitable for FPGA implementation. This conversion improves numerical stability and minimizes the risk of arithmetic overflow during hardware execution.

After preprocessing, the audio samples are transferred to the AWS EC2 F1 instance through a high-speed host-to-FPGA communication interface. The FPGA serves as the main processing engine, where the FIR filter is implemented using High-Level Synthesis (HLS). The filter is developed in C++ and synthesized into hardware using Xilinx Vivado HLS, simplifying the design process while maintaining high computational performance.

The FIR filter performs convolution using carefully designed filter coefficients to create a low-pass response. This configuration preserves speech components within the 0–4 kHz frequency range while effectively reducing high-frequency noise. The hardware implementation employs fixed-point arithmetic together with pipelining and parallel multiply-accumulate (MAC) operations. These optimizations enable deterministic, sample-by-sample processing with very low latency and high throughput, ensuring that each input sample is processed within a fixed number of clock cycles.

Once the filtering process is complete, the enhanced audio signal is sent to the output stage. Depending on the application, the processed audio can be played back in real time, stored in cloud storage, or forwarded to communication systems for further processing. The use of cloud-hosted FPGA resources removes the need for dedicated physical hardware while allowing remote configuration, simplified maintenance, and scalable deployment.

The proposed architecture also supports horizontal scaling, enabling multiple FPGA instances to process different audio streams simultaneously or handle higher sampling rates when required. As illustrated in **Fig. 1**, the complete processing pipeline transforms noisy input audio into a clean, enhanced signal through efficient parallel hardware execution.

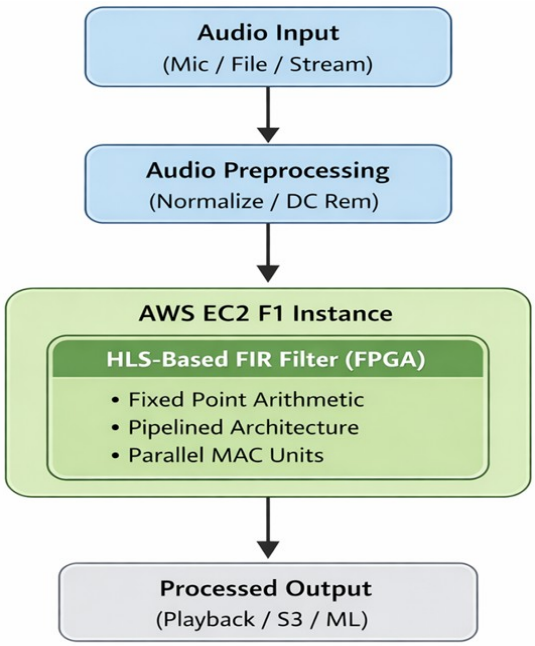


Figure 1 Architecture of Signal Processing With FPGA on AWS F1

Overall, the proposed methodology provides a scalable, energy-efficient, and high-performance solution for real-time speech enhancement and noise suppression. By combining FPGA acceleration with cloud infrastructure, the system delivers low-latency processing suitable for modern communication, embedded, and industrial audio applications.

4. RESULTS AND DISCUSSION

4.1 Functional Accuracy and Signal Integrity

The proposed FPGA-based real-time signal processing system was evaluated to verify its functional accuracy and signal quality. The complete processing pipeline included audio acquisition, preprocessing, FIR filtering, and Fast Fourier Transform (FFT) analysis. To validate the correctness of the implementation, the filtered output was compared with reference results generated using MATLAB. The comparison confirmed that the FPGA implementation produced accurate results while preserving the integrity of the original signal.

Figure 2 illustrates the input audio signal after the addition of high-frequency noise, which was used to evaluate the performance of the proposed filtering system.

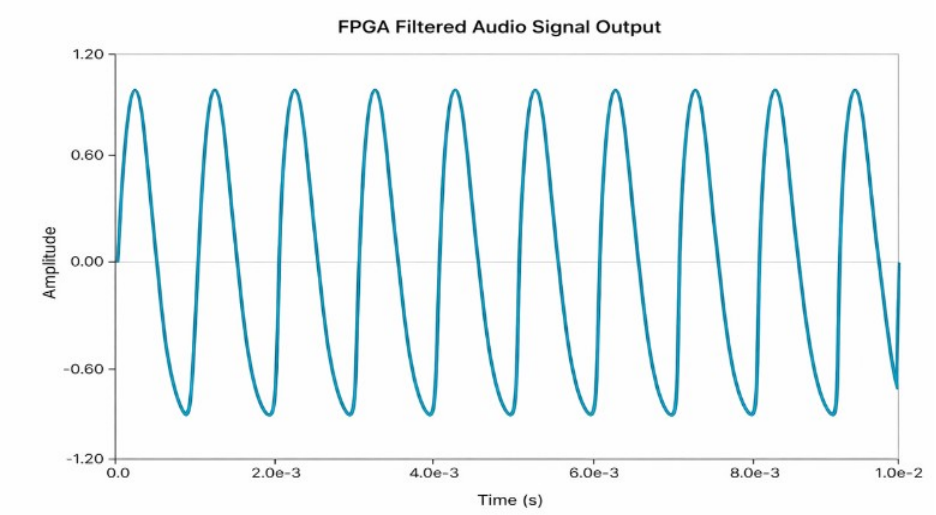


Figure 2 Input audio with added high-frequency noise

Table 1 Latency and Throughput Analysis

Platform	Latency per Sample (µs)	Throughput (kHz)	Power Draw (W)	Explanation
CPU (Intel Xeon)	35–50	20	~85	Acceptable for non-real-time or buffered operations
GPU (NVIDIA T4)	10–15	100	~250	Good for batch processing but non-deterministic timing
FPGA (AWS F1)	2–5	250–500	~70	Ultra-low latency and deterministic execution; ideal for real-time DSP

Table 1 compares the performance of CPU, GPU, and FPGA platforms in terms of processing latency, throughput, and power consumption. The FPGA-based implementation achieved the lowest processing latency, ranging from **2–5 µs per sample**, while maintaining

the highest throughput of **250–500 kHz**. In contrast, CPU-based processing exhibited significantly higher latency due to sequential execution, making it less suitable for real-time applications. Although GPUs offer higher throughput than CPUs, their execution time can vary depending on workload scheduling, resulting in non-deterministic performance.

The FPGA implementation benefits from hardware-level parallelism and pipelined execution, allowing multiple operations to be performed simultaneously. This architecture ensures predictable processing time, making it particularly suitable for latency-sensitive applications such as speech processing, wireless communication, and embedded signal processing.

Table 2 FPGA Resource Utilization

Resource	Usage	Available	Utilization (%)	Explanation
LUTs	73,000	1,182,240	6.2	Low utilization allows more complex processing modules
Flip-Flops	82,500	2,364,480	3.5	Plenty of capacity for multi-channel or adaptive filtering
DSP Slices	320	6,840	4.7	Efficient use of DSP blocks for FIR/FFT operations
BRAM	170	2,160	7.8	Sufficient memory for buffering and streaming multiple signals

Table 2 summarizes the hardware resource utilization of the proposed FPGA design. The implementation uses only a small percentage of the available FPGA resources, including Look-Up Tables (LUTs), Flip-Flops, DSP slices, and Block RAM (BRAM). This efficient utilization demonstrates that the design is well optimized and leaves sufficient resources available for future enhancements.

The low utilization of LUTs and Flip-Flops indicates that additional processing modules can be integrated without exceeding hardware limits. Similarly, the moderate use of DSP slices confirms the efficient implementation of arithmetic operations required for FIR filtering and FFT computation. The available BRAM resources also provide adequate memory for buffering streaming audio data and supporting multi-channel signal processing. Overall, the resource utilization results demonstrate that the proposed architecture is scalable and capable of supporting more advanced real-time processing applications.

Table 3 Scalability and Deployment on AWS

Feature	Explanation
Multiple FPGA instances	Each instance independently processes unique data streams efficiently
PCIe Streaming	Minimal data transfer latency
Storage	Seamless integration with AWS ecosystem
Orchestration	Automated deployment and control logic
Monitoring	Real-time metrics: throughput, temperature, error logs

Table 3 highlights the scalability features of the proposed cloud-based FPGA framework. AWS EC2 F1 instances enable multiple FPGA accelerators to run independently, allowing different audio streams to be processed simultaneously. This capability supports scalable deployment without requiring dedicated on-premise hardware.

High-speed PCIe communication minimizes data transfer latency between the host system and the FPGA, ensuring efficient real-time processing. Integration with AWS storage services simplifies data management, while cloud orchestration tools enable automated deployment, configuration, and resource management. In addition, real-time monitoring features provide continuous information about system throughput, hardware temperature, and error logs, improving reliability and simplifying system maintenance.

The combination of FPGA acceleration and cloud infrastructure offers a flexible and scalable platform that can easily adapt to increasing computational demands while maintaining consistent real-time performance.

5. CONCLUSION

The experimental results clearly demonstrate the advantages of implementing real-time signal processing on FPGA hardware within a cloud environment. Compared with traditional CPU and GPU implementations, the proposed system delivers significantly lower latency, higher throughput, and more predictable execution. These improvements are primarily achieved through parallel processing and pipelined hardware architecture, which allow multiple arithmetic operations to be executed simultaneously.

Efficient utilization of FPGA resources further confirms that the design is optimized and can be extended to support more complex algorithms or multiple processing channels without requiring major hardware modifications. In addition, deploying the FPGA on AWS EC2 F1 instances eliminates the need for dedicated hardware infrastructure while providing the flexibility to scale resources based on application requirements.

Overall, the experimental results validate the effectiveness of the proposed framework as a practical solution for real-time audio signal processing. Its combination of low latency, high throughput, efficient resource utilization, and cloud scalability makes it suitable for a wide range of applications, including wireless communication, speech enhancement, industrial monitoring, healthcare systems, and embedded signal processing.

REFERENCES

[1]. S. Abdelazim, D. Santoro, and F. Moshary, “FPGA programming challenges when estimating power spectral density and autocorrelation in coherent Doppler lidar systems for wind sensing,” *Sensors*, vol. 25, no. 3, p. 973, 2025.

[2]. T. Aarrestad *et al*., “Fast convolutional neural networks on FPGAs with hls4ml,” *arXiv preprint.*

[3]. P. Ai, “PulseDL-II: A system-on-chip neural network accelerator for timing and energy extraction of nuclear detector signals,” in *Proc. 23rd IEEE Real Time Conf. (RT 2022)*, 2022.

[4]. R. Barnwal and K. S. [Initials unavailable], “Morphological galaxy classification using convolutional neural networks on FPGA,” in *Proc. IEEE SPACE 2024 Conf.*, 2024.

[5]. C. P. G. Chinni and S. Hameed, “Design and realization of RFSoC FPGA-based multi-channel data acquisition and digital beam former system,” in *Proc. IEEE SPACE 2024 Conf.*, 2024.

[6]. E. Govorkova *et al*., “Autoencoders on FPGAs for real-time, unsupervised new physics detection at 40 MHz at the Large Hadron Collider,” *arXiv.*

[7]. K. Hennessy and K. Rinnert, “FPGA tracking with oneAPI,” in *Proc. 24th IEEE Real Time Conf. (RT 2024)*, 2024.

[8]. K. Jeziorek *et al*., “Embedded graph convolutional networks for real-time event data processing on SoC FPGAs,” *arXiv preprint.*

[9]. W. Que, “Real-time FFT calculation using FPGA PCIe card for KSTAR magnetohydrodynamic analysis,” in *Proc. 24th IEEE Real Time Conf. (RT 2024)*, 2024.

[10]. Z. Que *et al*., “Accelerating recurrent neural networks for gravitational wave experiments,” *arXiv preprint.*

[11]. D. Singh and R. Chandel, “FPGA-based hardware-accelerated design of linear prediction analysis for real-time speech signal,” *Arabian Journal for Science and Engineering*, vol. 48, pp. 14927–14941, 2023.

[12]. X. Wang *et al*., “Lightweight and real-time infrared image processor based on FPGA,” *Sensors*, vol. 24, no. 4, p. 1333, 202