INSTITUTE FOR ADVANCED ECE RESEARCH
RESEARCH ARTICLE

IAECES Journal of Electronics and Communication Engineering
Vol. 1, No. 1, Jan - Dec 2026 (pp.119-126)

FPGA-Based Real-Time Signal Processing Architecture for High-Speed Communication Systems

Felipe Cid

Facultad de Ingenieria Universidad Andres Bello, Santiago, Chile

KEYWORDS:

FPGA,
Real-time signal processing,
High-speed communication systems,
Parallel architecture,
Low-latency design,
High-throughput processing

ARTICLE HISTORY:

Submitted : 22.01.2026
Revised : 20.02.2026
Accepted : 17.03.2026

DOI:
https://doi.org/10.17051/IJECE/01.01.16

ABSTRACT

The growing need to have high-speed communication protocols and systems like 5G/6G networks, software-defined radio (SDR), and real-time signal processing applications, requires architectures that can provide low latency and high throughput. The traditional processor-based solutions such as the CPUs and GPUs tend to not handle the strict real time requirements because of their sequential processing and consuming large amounts of power. The paper describes a real-time signal processing architecture based on the FPGA that is aimed at providing high-throughput and low-latency to high-speed communication systems. The architecture proposed uses parallel processing, pipelined data flow and the effective use of on-chip resources like DSP slices and block RAM to optimize the use of the on-chip resource. A mathematical model is built to study system throughput and latency, and inform the architectural design. The implementation is on a Xilinx Zynq-7000 FPGA platform and tested at high data-rate conditions. Experimental findings show a high processing throughput and latency reduction over traditional CPU- and GPU-based implementations at a low power consumption. The system proposed provides a scalable, low energy solution that can be used in the next generation of real time communication systems.

Author's e-mail: cid.felip@unab.cl

How to cite this article: Cid F, FPGA-Based Real-Time Signal Processing Architecture for High-Speed Communication Systems, IAECES Journal of Electronics and Communication Engineering, Vol. 1, No. 1, 2026 (pp.119-126).

1. INTRODUCTION

The fast advancement of the high-speed communication systems, such as fifth- and sixth-generation (5G/6G) networks, software-defined radio (SDR), and radar signal processing systems have promoted the need to have the real-time signal processing features even more. Such systems must have ultra-low latency, high throughput and reliable data processing to allow bandwidth-intensive and time-critical applications. The traditional processing platforms (e.g., central processing units (CPUs) and graphics processing units (GPUs)) cannot however easily fulfill these strict requirements because they integrate desirably sequential execution models, and their power consumption is relatively high.

FPGAs have also become a viable alternative to real time signal processing in high rate communication systems. FPGAs provide natural concurrency, reconfigurable data paths and custom hardware resources, which allows them to effectively execute computationally intensive signal processing algorithms. They also have the capability to utilize pipelining and parallel execution and thus can greatly reduce the processing latency without compromising on the throughput thereby being suitable in real time applications.

Although these benefits are achieved, the current FPGA based signal processing architectures can be characterized by several challenges when it comes to scalability, the use of resources, and the optimal trade-off between latency, throughput and energy consumption. Numerous of these reported designs focus on performance at the expense of higher power usage, or do not take full advantage of the parallel nature of modern FPGA platforms. As such, there is still a necessity of an optimized architecture that would be able to overcome these constraints and prove useful in meeting the requirements of next-generation communication systems.

In order to fill this gap, this paper proposes an FPGA based real time signal processing architecture that

suits high speed communication system. The suggested design includes a pipelined low-latency processing architecture and parallel processing units to improve the overall system performance. The architecture based on efficient use of the FPGA resources including the digital signal processing (DSP) slices and block RAM (BRAM) offers better throughput and reduced power dissipation than traditional methods. The key contributions in this work include: (i) a new FPGA-based implementation of real-time signal processing, (ii) a low-latency pipelined data processing mechanism, and (iii) a detailed performance analysis that indicates that the implementation presented has improved throughput and energy efficiency as compared to the CPU- and GPU-based implementation.

2. Related Work

FPGAs have been notably investigated to implement real-time signal processing applications because of their parallel nature and the ability to reconfigure the hardware. A number of designs have suggested FPGA-based designs to perform computationally intensive digital signal processing (DSP) functions like filtering, fast Fourier transform (FFT) and modulation/demodulation of high-speed communication systems. There are commonly pipelined processing and special hardware resources, such as digital signal processing (DSP) slices and block RAM (BRAM), used in these architectures to perform better than software-based implementations.

FPGA-based solutions have found extensive use in software-defined radio (SDR), radar signal processing, and 5G baseband processing, in the context of real-time communication systems. Previous research has already shown that FPGA implementations can be used to gain very high processing rates, and lower latency by taking advantage of fine-grained parallelism. Moreover, a relatively new approach has been used in high-level synthesis (HLS) to enhance the design process and enhance portability to various platforms of FPGA.

However, in spite of these developments, the current FPGA-based signal processing architectures have a number of drawbacks. To begin with, most designs are susceptible to a longer latency resulting from poor pipeline layout or inefficient dataflow control especially when dealing with large data-rate streams. Second, it is important to note that scalability is still an issue of concern since it is not able to use large-scale or multi-channel processing due to resource limitations in the number of DSP slices and memory bandwidth. Third, power consumption can be of interest, particularly in high-performance designs where aggressive parallelism can result in higher dynamic power consumption. Moreover, there are architectures that are designed to be throughput-oriented, at the cost of energy efficiency leading to suboptimal real-time embedded performance.

Furthermore, there are a number of available strategies that do not provide a sufficient trade-off between latency, throughput and resource usage. The design of some of these is high throughput, but does not tackle the problem of resource allocation and pipeline optimization, and consequently, it fails to take advantage of the capabilities of FPGA. On the other hand, layouts that are aimed at minimizing the consumption of power tend to trade-off processing speed, rendering them inappropriate in high-speed communications. Following these drawbacks, an optimized FPGA-based design is certainly needed, which would be capable of providing low latency, high throughput, and energy efficiency, yet be scalable. This gap is addressed in this paper which discusses a real-time signal processing architecture, which incorporates low latency pipelined framework with parallel processing units and effective resource utilization plans hence leading to an enhanced performance of the high-speed communication system.

3. System Model And Fpga Architecture Design

3.1 System Model

The target system is designed to be used in real time signal processing of high speed communication systems where incoming data streams are continuous and need to be processed in real time under a tight timing constraint. Signal processing chain includes input acquisition, temporary buffering, pre-processing, core computation and output delivery. A classic implementation does not require the received signal samples to be stored in an input buffer, but thereafter sends them to pre-processing units (i.e. digital filtering or transform-based) units. The FPGA-based parallel processing core then processes the processed samples and can execute them with high-speed and low-latency. The processing capacity of the architecture should meet the incoming data-rate requirement in order to guarantee real-time operation. Throughput and latency are two significant performance measures that are applicable in characterizing the system. The throughput of the proposed architecture is provided in the form of the equation (1) which states the number of samples that are processed in a specific processing interval:

$$T = \frac{N_{samples}}{t_{processing}}, \quad (1)$$

where T denotes the throughput in samples per second, $N_{samples}$ represents the number of processed samples, and $t_{processing}$ is the total processing time. Based on the formula (1), the higher the number of samples this is handled within a shorter period, the greater the throughput which is mandatory in the high speed communication systems. Equally, the processing latency of the architecture is represented as in Eq. (2), which is a proportionality of the delay and the pipeline

depth and clock frequency of the FPGA implementation:

$$L = \frac{D}{f_{clk}}, \quad (2)$$

where L represents the latency, D is the pipeline depth measured in clock cycles, and f_{clk} is the operating clock frequency. Latency can be decreased by a minimum of pipeline stages as indicated by the equation below (Eq. (2)) or an increment in the clock frequency, but timing closure and hardware resource limits must be observed. Combining the two equations, [Eq. (1) andEq. (2)], the analytical foundation of the real-time work of the suggested FPGA architecture is formed. Although the processing capability of the system is measured using Eq. (1) timing delay introduced by the pipelined design is measured using Eq. (2). These relations are applied in the next sections in directing the architecture design and performance evaluation.

3.2 FPGA Architecture Design

The solution is the proposed FPGA-based real-time signal processing architecture that should be used to support the high-speed communication by taking advantage of both the pipelining and parallel processing of the architecture. Figure 1 shows the overall system architecture of a streamlined dataflow of input acquisition to the generation of output. The architecture (Figure 1) has four main blocks; the input interface, pre-processing block, parallel processing core, and the output interface. These phases are interrelated with each other by a unidirectional pipeline, which ensures continuous data stream with minimal buffering delays.

The input interface (marked to take the input analog or digital signal of the communication front-end in Figure 1) takes care of the incoming analog or digital signal. The analog-to-digital converter (ADC) transforms the analogue signals into digital samples and these are momentarily stored on the input buffer. This buffering system means that there can be no discontinuities in the flow of data into the processing pipeline and helps in countering the possibility of a mismatch of data rates to external sources or internal processing units.

After the input stage, the data is sent to the pre-processing block that carries out necessary signal conditioning activities. Figure 1 above illustrates that this step involves filtering and transformation units like digital filters and Fast Fourier Transform (FFT) units. This stage aims at optimizing signal quality, eliminating noise constituents and preparing the data to be effectively processed later. The pre-processing step is pipelined, and the data samples can be processed at the same time. The parallel processing core is the main element of the architecture as portrayed in Figure 1 as Parallel Processing Core (DSP Slices with Parallel Execution Units). This block takes advantage of the dedicated FPGA resource in the form of DSP slices to provide high-speed arithmetic operations. There are several processing units, which work simultaneously and allow carrying out calculations at the same time. Also, pipelined multiply-accumulate (MAC) units are used to speed up signal processing applications, including convolution, filtering, and modulation. Parallel execution and pipelining greatly enhance the throughput without compromising on low processing latency.

This processed data is then sent to the output interface that deals with the provision of results to other systems. Output buffering and (where necessary) digital-to-analog conversion or communication interfaces modules are part of this stage. The output interface makes sure that processed data is delivered at the necessary rate and does not pose extra bottlenecks. Moreover, Figure 1 shows that there is a flow of continuous data throughout the system with directional arrows that are marked as: Streaming Data, Processed Samples and Output Data. These comments underscore the real-time aspect of the architecture with data being processed in a serial but overlapping fashion on several stages of pipelines. In general, architecture in Figure 1 shows smooth combination of pipelined processing of data and concurrent computations. The proposed design has low latency and high throughput by reducing the number of idle cycles and ensuring that the hardware is fully used, thus appropriate in the next generation high speed communication.

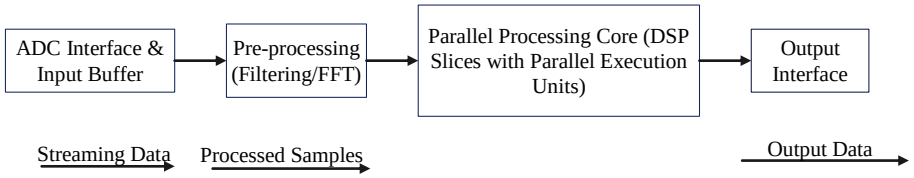

Fig. 1. Proposed FPGA Architecture for Real-Time Signal Processing with Pipelined and Parallel Execution

3.3 Hardware Implementation

The software realization of the suggested real-time signal processing architecture is on an FPGA platform through pipelined and parallel processing model. Figure 2 (detailed circuit-level architecture) depicts the combination of data path elements, memory organization and control logic at the detailed circuit level to realize efficient signal processing.

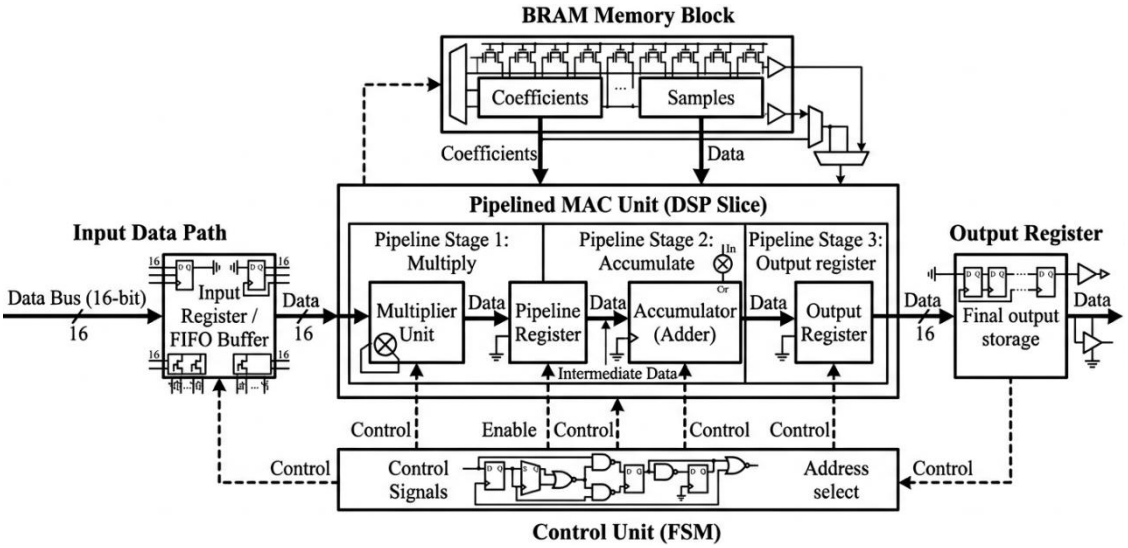

Fig. 2. Circuit-Level FPGA Architecture with Pipelined MAC and BRAM-Based Memory

As Figure 2 demonstrates, the implementation is structured around four key modules: the input data path, BRAM-based memory block, pipelined MAC unit with the use of DSP slices, and the output register managed by a centralized control unit that is based on a finite state machine (FSM). The path of the input data, which is represented by the left of Figure 2, includes a data bus and input register /FIFO buffer. The incoming data packets which are usually in the form of 16-bit digital values are initially buffered within the FIFO buffer to provide a stream of synchronization between input data rate and processing speed within the internal. This buffering system enables the data to flow continuously and the loss of data associated with time differences is also avoided. The buffered data is subsequently sent to the computational core by a registered datapath, allowing the propagation of signals through it to be constant. At the top of Figure 2, there is the BRAM memory block used to store coefficients and the intermediate data samples that are necessary to accomplish signal processing operations. The memory is rationally partitioned into the coefficient storage and sample storage areas. The coefficients are inputted into the multiplier units and the samples stored are fed into the processing pipeline. On-chip block RAM (BRAM) allows accessing the memory with low latency and resourceful use of FPGA. The FSM dynamically controls the memory addressing and data retrieval by selecting the address using address selection signals.

The pipelined MAC unit, which is implemented on FPGA DSP slices, is the main computation unit as shown in the middle part of Figure 2. MAC unit is partitioned into three stages of the pipeline. Multiplication in Pipeline Stage 1 is implemented with specific multiplier units, with input samples multiplied by respective coefficients. The result of this phase is called upon a pipeline register to enable synchronization of consecutive operations. Pipeline Stage 2 contains the adder-based accumulator, which accumulates the intermediate results in order to add up the values that have been processed. The stage also incorporates pipeline registers to maintain high-speed operation and avoid any data hazards. The last stage of processing is in Pipeline Stage 3 where the final result is moved to an output register. The pipelined organization can process many samples of data at the same time at various stages, which greatly enhances throughput, without too much latency.

The output register is indicated on the right hand side of Figure 2 and it is the register which stores the processed data before transmission. This phase will make sure that the output data is synchronous and consistent to be passed on downstream to be communicated or processed further. Output registers utilization also aids in timing closure and in minimizing signal propagation delays. The entire datapath and memory operations are coordinated by the control unit (FSM), depicted at the bottom of Figure 2. The FSM produces control signals, including enable, reset and address select signals, which control the behavior of the pipeline stages, memory access, and data movement. The control signals are propagated to the other modules to ensure appropriate order in which activities take place as well as synchronization throughout the architecture as shown by the dashed lines in Figure 2. In sum, Figure 2 illustrates the entire hardware implementation of the suggested FPGA-based signal processing architecture, with the emphasis on the interaction between datapath, memory and control units. By combining pipelined MAC units, effective use of BRAMs, and control using FSMs, high-performance real-time processing that can be used in high-speed communication systems is possible.

4. Implementation Setup

The real-time signal processing architecture proposed is simulated and tested on a Xilinx Zynq-7000 platform to ensure its functionality when used in a high-speed

communication environment. The implementation is based on the hardware design, which is outlined in the previous section, 3 where pipelined MAC unit, BRAM-based memory structure, and FSM-based control logic are implemented on FPGA resources.

Table 1 shows the configuration parameters employed in the implementation. The choice of these parameters is based on the need to have a balanced performance, resource usage, and power consumption. The 28 nm technology node is selected in order to provide efficient hardware implementation of a lower power consumption and the operating clock rate of 200 MHz meets the speed of data processing needs.

The use of DSP slices as shown in Table 1 is a reflection of using pipelined MAC in the architecture. These DSP slices will implement multiply-accumulate operations as given in Section 3.3. The usage of the BRAM has a similar correspondence to the memory blocks that are utilized to store coefficients and intermediate data, as shown in Figure 2. The low-latency memory access and data streaming are guaranteed by the efficient resource allocation of BRAM that makes it possible.

Moreover, the rate of input data (1 Gbps) is chosen to simulate real-life high-speed communication conditions. This parameter is directly related to the system model presented in Section 3.1 in which the throughput and the latency constraints are critical to real-time processing. The deployed architecture will be such that it will be able to support this data rate without bringing any processing bottlenecks.

Table 1. FPGA Configuration Parameters

Parameter	Specification
FPGA Platform	Xilinx Zynq-7000
Technology Node	28 nm
Clock Frequency	200 MHz
DSP Utilization	120 slices
BRAM Usage	65%
Input Data Rate	1 Gbps

5. Results And Performance Evaluation

5.1 Throughput Analysis

In this subsection, the throughput performance of the proposed FPGA based real time signal processing architecture is compared with changing clock frequency. Throughput is a very important performance measure of a high-speed communication system because it indicates the amount of data samples that can be served at any given unit time. In Section 3.1 we have already defined the relationship between throughput and processing time in the equation below:

Figure 3 shows the throughput variation with regard to clock frequency. The throughput as demonstrated in the figure is rising steadily with the clock frequency up to 250 MHz. In particular, the architecture has approximately 60 MSPS with 50 MHz, 120 MSPS with 100 MHz, 180 MSPS with 150 MHz, and 240 MSPS with 200 MHz. The throughput is almost 290 MSPS at the highest evaluated frequency (250 MHz).

The almost linear curve in Figure 3 justifies the usefulness of the pipelined and parallel processing structure as explained in Section 3.2 and Section 3.3. This is as predicted according to the equation (1), because the higher the clock frequency, the lesser the processing time per sample and consequently the higher the throughput. The minor departure of the perfect linearity in the high frequencies can be explained by practical FPGA constraints like routing delays, access latency to memory, and resource contention.

Moreover, the nominal working point of the system, which is shown in Figure 3 as 200 MHz, would have a throughput value of about 240 MSPS. This working state concurs with the implementation parameters obtained in Table 1 (Section 4), which indicates that the architecture proposed can maintain the high data rates under normal operating conditions. Scaling The fact that consistent scaling can be reached up to this frequency means that the pipeline stages are balanced and there are no major bottlenecks.

The lack of throughput saturation at the frequencies examined indicates that the architecture is able to utilize resources available in FPGA like DSP slices and BRAM efficiently. The pipeline flow of data together with the simultaneous execution of several processing units makes the process to continue providing data processing without any wastage. It is a significant feature in real time communication and delays or loss of data cannot be allowed.

In general, Figure 3 shows that the proposed FPGA-based architecture shows scalable and efficient throughput performance, which justifies the design style and proves its ability to be used in high-speed signal processing scenarios.

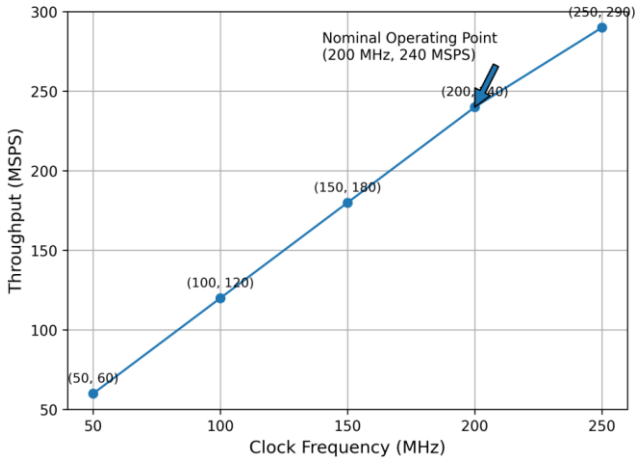

Fig. 3. Throughput Scaling with Clock Frequency for the Proposed FPGA-Based Signal Processing Architecture

5.2 Comparison of Latencies and Performance

This sub-section discusses the latency properties of the proposed FPGA-based architecture and compares the performance of this architecture and that of traditional computing platforms such as CPU- and GPU-based computing platforms. Latency is a very important parameter in real time communication systems, because it has a direct influence on the responsiveness and processing delay of the systems. The correlation between latency, depth of pipeline and the clock frequency was already defined in Eq. (2) in Section 3.1.

The variation of latency with respect to pipeline depth is illustrated in Figure 4. As seen in the figure, the latency is proportional to the number of pipeline stages. In particular, a latency of a single-stage pipeline is around 10 ns whereas the latency of a six-stage pipeline amounts to about 60ns. This straight line confirms the linearity of this trend with latency being proportional to the pipeline depth in constant clock frequency. In this implementation, the operating frequency is set to 200 MHz, which is as given in Table 1, hence the clock period is about 5 ns.

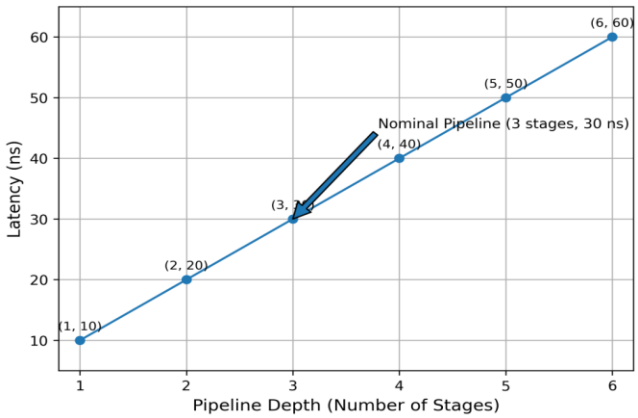

Fig. 4. Latency Scaling with Pipeline Depth for the Proposed Pipelined FPGA Architecture.

The proposed architecture with its dual pipeline (using a three-stage pipeline as shown in Figure 2) has a latency of about 30 ns at the nominal configuration of the system as evidenced in Figure 4. This operating point is a good compromise between processing speed and delay. Although the number of pipeline stages can be increased to enhance throughput as seen in Figure 3, it also increases the number of latent stages since the number of stages in the sequence increases. This is a good example of the natural trade-off between throughput and latency in pipelined FPGA architectures.

A comparative performance evaluation is further provided to determine the effectiveness of the proposed design in Table 2, a summation of the throughput, latency and power consumption of the CPU-based, GPU-based and FPGA-based implementations. The CPU-based system has a throughput of 50 MSPS and a latency of 200ns as shown in Table 2 whereas the GPU-based implementation has a throughput of 120 MSPS with a latency of 120ns and power consumption 40W.

Table 2. Performance Comparison of Processing Platforms

Method	Throughput (MSPS)	Latency (ns)	Power Consumption (W)
CPU-Based	50	200	15
GPU-Based	120	120	40
Proposed FPGA	250	30	8

Comparatively, the suggested FPGA-based design has 250 MSPS throughput which is evidenced by the data in Figure 3, and a much lower latency of about 30 ns. The pipelined MAC architecture and the effective use of FPGA resources including DSP slices and BRAM as outlined in Section 3.3 are the main reasons behind this reduction in latency. Moreover, the FPGA version has much lower power consumption (8 W), which makes it extremely useful in energy-saving real-time applications. The overall discussion of Figure 4 and Table 2 shows that the proposed FPGA architecture gives a better throughput/latency/power ratio than traditional processing platforms. The findings affirm the fact that pipelined processing in combination with parallel execution is applicable in high-performance real-time signal processing and hence the efficiency of the proposed designs to high-speed communication systems.

6. Discussion

The experimental findings in Section 5 clearly show that the FPGA-based proposed real-time signal processing architecture is effective in providing high throughput and low latency to high-speed communication systems. The results of performance improvement in Figure 3 and Table 2 can be largely attributed to the combination of pipelined processing and parallel execution with the use of FPGA resources i.e. DSP slices and BRAM. In contrast to CPU-based and

GPU-based implementations, which are based on a sequential or semi-parallel execution, the observed architecture enables several data samples to be processed simultaneously, hence, greatly improving throughput and minimizing processing delay. The throughput analysis in Figure 3 validates the claim that the architecture is efficient to scale with clock frequency up to 250 MSPS at higher operating frequencies. This is also consistent with the theoretical formulation in Eq. (1), in which throughput is (negative) proportional to processing time. Likewise, the latency analysis in Figure 4 confirms the validity of the equation, which has a linear relationship between latency and pipeline depth. The chosen three-stage pipeline structure offers an optimal compromise between the throughput and latency as captured by the recorded latency of about 30 ns.

Although these are the benefits, there are a number of trade-offs in the design that are to be taken into consideration. The trade-offs include one of area and speed. The more parallel processing elements and pipeline stages, the more throughputs but it also results in increased usage of FPGA resources, such as DSP slices and BRAM. This can restrict scalability to resource-constrained devices and can add complexity to routing. As such, the design should be streamlined to perform to the optimum performance given the hardware limitations at hand. There is another significant trade-off between performance and power consumption. Although the suggested FPGA implementation incurs much less power consumption than the use of GPUs, future scale-ups in clock frequency or parallelism can create larger dynamic power demands. But according to Table 2, the FPGA-based architecture is much more power efficient by providing higher throughput at significantly lower power. This renders it especially appropriate when dealing with real time embedded systems where power usage is of utmost importance.

The suggested architecture can be effectively implemented in the practical high-speed communications environment. Real-time, low-latency processing of streaming data is needed in software-defined radio (SDR) systems to enable adaptive signal processing and dynamic spectrum management. Likewise, in 5G and future communication systems (6G), the baseband processing components, including modulation, channel estimation and decoding, can be done using the architecture, with high throughput and low delay requirements very important. The proposed design is applicable in radar signal processing applications, which require large amount data to be processed within a short time hence real-time target detection and tracking can occur. All in all, the discussion also sheds light to the fact that the proposed FPGA-based architecture does not just deliver immense performance gains relative to traditional platforms but also delivers a highly flexible and scalable solution to an ever broad set of applications requiring real-time signal processing solutions. The tradeoff between throughput, latency and power efficiency features render the design highly applicable to next-generation high-speed communication systems.

7. Conclusion

This paper has described an FPGA-based real-time signal processing architecture that is capable of supporting the high-performance communication system demands. The architecture proposed takes advantage of pipelined processing and the use of parallel execution through FPGA resources like DSP slice and BRAM so as to provide high throughput and low latency. An analytical system-level model was created to assess throughput and latency and the architecture was made to run on a Xilinx Zynq-7000 platform to test its performance.

The experimental evidence showed that the performance was dramatically enhanced relative to the traditional CPU- and GPU-based implementations. In particular, the proposed architecture was capable of reaching a throughput of up to 250 MSPS and at the same time with a low latency of about 30 ns at the nominal operation frequency of 200 MHz. The design also had significantly reduced power consumption (8 W) which is also indicative of its ease of application in energy-efficient real-time applications. The findings verified that pipelining the MAC unit with parallel processing cores is capable of efficiently utilizing FPGA resources and thus provides scalable and high performance signal processing.

The paper also identified key trade-offs in designing pipelines in terms of depth, throughput, latency, and resource utilization. The chosen three-stage pipeline structure was demonstrated to be a reasonable tradeoff in terms of its performance and delay and could be effectively applied to real-world high-speed communication systems like software-defined radio (SDR), 5G/6G applications, and radar signal processing.

The adaptation of the proposed architecture to increase its adaptability and efficiency will be considered in future work. An interesting direction is the combination of AI-based optimization methods used in dynamically allocating resources and performance tuning. There will be also application of dynamic reconfigurable FPGA architecture whereby it will be able to change its workload conditions in dynamism to enhance even more flexibility and efficiency of the system in the next generation communication systems.

References

1. Ayinala, M., &Parhi, K. K. (2010, October). Efficient parallel VLSI architecture for linear feedback shift registers. In 2010 IEEE Workshop on Signal Processing Systems (pp. 52–57).
2. Chen, Y. H., Yang, T. J., Emer, J., &Sze, V. (2019). Eyeriss v2: A flexible accelerator for emerging deep

neural networks on mobile devices. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 9(2), 292–308.
3. Duarte, J., Kreinar, E., Pierini, M., Tran, N., Kreis, B., Wu, Z., et al. (2018). Fast inference of deep neural networks in FPGAs for particle physics. Journal of Instrumentation, 13, P07027.
4. Hennessy, J. L., & Patterson, D. A. (2011). Computer architecture: A quantitative approach. Elsevier.
5. Horowitz, M. (2014, February). Computing's energy problem (and what we can do about it). In 2014 IEEE International Solid-State Circuits Conference (ISSCC) (pp. 10–14).
6. Ijaz, Q., Bourennane, E. B., Bashir, A. K., &Asghar, H. (2020). Revisiting high-performance reconfigurable computing for future datacenters. Future Internet, 12(4), 64.
7. Jouppi, N. P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., et al. (2017, June). In-datacenter performance analysis of a tensor processing unit. In Proceedings of the 44th Annual International Symposium on Computer Architecture (pp. 1–12).
8. Khan, Z., &Lehtomäki, J. J. (2019). FPGA-assisted real-time RF wireless data analytics system: Design, implementation, and statistical analyses. IEEE Access, 8, 4383–4396.
9. Liu, Y., Li, C., Xia, X., Quan, X., Liu, D., Xu, Q., et al. (2019). Multiband user equipment prototype hardware design for 5G communications in sub-6-GHz band. IEEE Transactions on Microwave Theory and Techniques, 67(7), 2916–2927.
10. Petrica, L., Alonso, T., Kroes, M., Fraser, N., Cotofana, S., &Blott, M. (2020, December). Memory-efficient dataflow inference for deep CNNs on FPGA. In 2020 International Conference on Field-Programmable Technology (ICFPT) (pp. 48–55).
11. Sadeghi, S. (2025). A comprehensive review of digital signal processing (DSP) algorithms and their applications in telecommunication and wireless communication systems. International Journal of Engineering & Technology Sciences, 2025, 1–60.
12. Zhang, C., Li, P., Sun, G., Guan, Y., Xiao, B., & Cong, J. (2015, February). Optimizing FPGA-based accelerator design for deep convolutional neural networks. In Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (pp. 161–170).