IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁑ Peer-reviewed & Refereed journal ⁑ Vol. 13, Issue 7, July 2025

DOI: 10.17148/IJIREEICE.2025.13712

Hardware Accelerators for FFT Optimization: A Review of Genetic Algorithm-Based Approaches

Maenas Kirubahar E[1], Kishan S[2], Aaditya Pandey[3], Mukund Woodi[4]

Department of Electronics and Communication, Dayananda Sagar Academy of Technology and Management[1-4]

Abstract: Fast Fourier Transform (FFT) algorithms are fundamental in a wide range of digital signal processing applications, including audio analysis, image processing, and telecommunications. However, optimizing FFT implementations for speed, efficiency, and hardware utilization remains a complex task due to the numerous tunable architectural parameters such as radix type, stage configuration, word length, and pipeline depth. This paper presents the design and implementation of a hardware accelerator optimized through a Genetic Algorithm (GA), which intelligently explores the design space to identify parameter sets that yield optimal performance. The GA operates by encoding FFT configuration parameters into chromosomes and evolving them based on a fitness function that considers throughput, latency, power efficiency, and resource usage. A hardware model of the FFT accelerator is developed and evaluated through simulations to measure performance improvements against traditional fixed-parameter designs. Results demonstrate that GA-optimized FFT configurations lead to notable gains in processing speed and computational efficiency, validating the effectiveness of evolutionary algorithms in hardware design optimization. This work showcases the synergy between artificial intelligence and hardware engineering for advanced digital signal processing systems.

Keywords: Genetic Algorithm (GA), Hardware Acceleration, Fast Fourier Transform (FFT), Parameter Optimization, Digital Signal Processing (DSP), High-Level Synthesis, Hardware Design Automation, Signal Processing Architecture

I. INTRODUCTION

The Fast Fourier Transform (FFT) is a computationally efficient algorithm for computing the Discrete Fourier Transform (DFT), widely used in digital signal processing (DSP) applications such as image analysis, audio compression, wireless communication, and biomedical signal processing. Due to its critical role in real-time systems, optimizing FFT performance for speed, accuracy, and hardware efficiency is essential, especially in resource-constrained or performance-sensitive environments. Traditionally, FFT implementations rely on fixed architectural parameters such as radix type (radix-2, radix-4), stage count, butterfly structure, and word length. While these fixed configurations offer simplicity, they often result in suboptimal trade-offs between area, power consumption, latency, and throughput. Manual tuning of these parameters is not only time-consuming but also impractical when dealing with a large design space.

To address this challenge, this paper proposes a novel approach that leverages a Genetic Algorithm (GA) for the intelligent exploration and optimization of FFT design parameters. Genetic Algorithms, inspired by the principles of natural evolution, are highly effective in solving complex optimization problems involving multiple conflicting objectives. By encoding FFT architectural choices as genetic chromosomes, the GA iteratively searches for the most optimal configuration based on a fitness function that considers execution time, resource usage, and computational efficiency.

A hardware model of the FFT accelerator is developed, and simulations are conducted to evaluate the effectiveness of GA-optimized configurations against traditional designs. The results show significant improvements in performance and efficiency, highlighting the potential of integrating artificial intelligence techniques into hardware design workflows. This research aims to contribute to the advancement of adaptive and intelligent hardware design methodologies for high- performance DSP systems.

II. LITERATURE SURVEY

The evolution of digital signal processing (DSP) and artificial intelligence (AI) applications has created an increasing demand for customizable and efficient hardware accelerators, particularly for computationally intensive algorithms

DOI: 10.17148/IJIREEICE.2025.13712

such as the Fast Fourier Transform (FFT). As modern applications transition to real- time and embedded environments, it becomes crucial to not only accelerate these operations through specialized hardware but also to intelligently optimize architectural parameters like radix type, pipeline depth, and bit width. Recent research has explored the synergy between evolutionary computing—especially genetic algorithms (GAs)—and VLSI hardware design, offering promising approaches to dynamic and high- performance accelerator architectures.

Sathia et al. [1] explore GPU-accelerated genetic programming as a method to evolve hardware configurations in parallel, significantly reducing the time required to identify optimal solutions. Their implementation capitalizes on GPU parallelism to evaluate large populations in fewer generations, a concept that resonates closely with the optimization loop proposed in this paper, where evaluation cycles must remain efficient even with high hardware complexity.

The theoretical foundation of genetic algorithms is thoroughly laid out by Alhijawi and Awajan [2], who dissect GA operators such as crossover, mutation, and selection while demonstrating their adaptability across a wide range of engineering problems. Their study affirms that GAs are highly effective for exploring large and non-linear solution spaces, making them ideal for design space exploration (DSE) in hardware accelerators where traditional exhaustive methods are computationally prohibitive.

Huang et al. [3] present a hybrid GA model executed in parallel on GPUs, applied to generalized assignment problems. While not specific to FFT, their approach showcases how GAs can be adapted and parallelized for high-speed optimization, which directly supports the core methodology of this paper in terms of runtime performance and scalability of the GA.

Kao et al. [4] introduce DiGamma, a domain-aware GA framework tailored for co-optimizing deep neural network accelerators. The paper discusses how GAs can be made hardware-aware by incorporating domain- specific constraints during evolution, making the generated configurations not just functionally correct, but also resource-efficient and performance-balanced.

The application of such awareness in the context of FFT enables the GA to prioritize configurations that, for example, reduce DSP usage or maximize pipeline efficiency.

Fan et al. [5] contribute to the field of algorithm- hardware co-design by proposing a reconfigurable accelerator for CNNs. Their approach emphasizes the importance of balancing algorithmic requirements and hardware capabilities, using HLS (high-level synthesis) to bridge the gap. This co-design philosophy parallels the architecture in this paper, where FFT functionality is shaped dynamically based on parameters output from the GA.

In [6] the design and implementation of a pipelined FFT architecture, a core algorithm in digital signal processing (DSP), enhanced with an efficient clock gating technique. Clock gating is employed to minimize unnecessary switching activity in functional blocks during their idle periods, thereby reducing dynamic power consumption and improving overall energy efficiency of the system.

In-memory computing emerges as a frontier solution in the work of Leitersdorf et al. [7], who propose **Fourier PIM**, an in-memory FFT architecture that drastically reduces the cost of data movement—a major bottleneck in conventional systems. Their innovation in data locality and throughput aligns with this paper's goal of achieving high-speed FFT computation under hardware constraints.

Lu et al. [8] present **GFFT**, a task graph-based optimization framework for FFT workloads. Their work targets concurrency and task scheduling, offering a layered abstraction of FFT operations. Their results show that decomposing FFT into task graphs helps identify hardware-acceleratable portions, which can be exploited further using GA for fine-tuned hardware generation.

A more compiler-centric approach is offered by Woodruff et al. [9], who develop a system that translates real software programs into hardware FFT accelerators. Their approach bridges high- level language constructs to synthesizable hardware blocks, which complements the objective of this research to generate hardware from optimized algorithmic parameters. Their work also validates the feasibility of compiler-assisted hardware synthesis in dynamic reconfiguration environments.

The integration of AI into hardware accelerators has advanced the optimization of complex signal processing tasks like FFT. Genetic algorithms, in particular, enable intelligent tuning of FFT parameters for improved performance and efficiency. Embedded systems, such as those using ARM-based controllers, highlight the potential of real-time processing with minimal manual intervention. These approaches support the design of adaptable, low-power hardware accelerators for high-speed applications.[10]

Hendrickx et al. [11] explore neural architecture search (NAS) using genetic optimization to tune design parameters for efficient inference. Their use of multi-objective GAs and design-space constraints shows the power of evolutionary methods in solving real-time hardware synthesis challenges. Their framework supports intelligent DSE across multiple layers of abstraction, which mirrors the requirements of FFT optimization at both algorithmic and architectural levels.
Ozcan and Savaş [12] focus on efficient GPU implementations of the Number Theoretic Transform (NTT), a variant of FFT used in cryptographic algorithms. Their work highlights the use of parallel computing resources and optimal memory access patterns, both of which are relevant in guiding GA-based hardware generation to achieve optimal dataflow.

The paper [13] presents a study on various parameters influencing video processing performance, including block size, motion estimation schemes, motion types (slow, medium, and fast), compression levels (high and medium), and the impact of scaling or not scaling DCT coefficients. These factors are analyzed to understand their effects on processing efficiency and output quality

Sharma et al. [14] address GA-based embedded processor design space exploration. Their approach evaluates performance, area, and energy metrics for multiple configurations in an automated loop— highly comparable to the feedback-driven architecture in this paper, where hardware performance is looped back into the GA as fitness scores.
Li et al. [15] specifically apply GAs for FFT kernel tuning in AI accelerators. Their methodology integrates profiling data into the optimization loop, allowing the GA to adapt configurations to meet both speed and power goals. This direct application of GAs to FFT performance optimization supports the core premise of this research and validates the approach in industrial contexts.

Chen et al. [16] automate FFT core generation using AI-assisted parameter tuning and synthesis. Their study utilizes a modular FFT framework where parameters like stage depth, butterfly structure, and radix type are dynamically configured. Their results show notable improvements in area and performance, establishing a close match to the architecture proposed in this paper.

Rao and Srinivasan [17] develop a multi-objective GA optimization for low-power FFT cores targeted at wireless sensor networks. Their optimization considers not only performance but also energy constraints critical for battery-operated systems. Their evolutionary model is valuable for future enhancements of the current system, especially for IoT and mobile DSP applications.

Finally, Gupta and Verma [18] propose a resource-aware FFT accelerator that leverages machine learning-guided parameter exploration. Their method integrates hardware constraints like LUT and DSP availability into the optimization loop. This advanced form of intelligent exploration complements genetic algorithms and suggests hybrid models for future work.

Together, these studies form a comprehensive landscape of how genetic algorithms and intelligent design techniques are revolutionizing the way hardware accelerators are conceived, configured, and implemented. The use of AI-driven optimization not only improves performance and efficiency but also enables automation of hardware generation processes.
The current paper builds upon these advancements by presenting a closed-loop, GA-driven framework that generates an FFT accelerator optimized across multiple hardware dimensions, thereby offering a scalable, adaptable, and high-performance solution for modern DSP systems.

III. GAP ANALYSIS

Fast Fourier Transform (FFT) is an essential component in many digital signal processing (DSP) systems, and its performance is critical in applications such as wireless communication, radar, image processing, and biomedical signal analysis. Over the years, extensive research has been conducted in designing efficient FFT architectures and optimizing their performance using various hardware techniques. However, several significant gaps remain in the current state of research and implementation practices, particularly around automated, intelligent hardware optimization. This section presents a detailed gap analysis based on a review of related work and prevailing challenges in the domain.

Lack of Dynamic and Automated Parameter Optimization

Most existing FFT implementations use **fixed architectural parameters**, such as radix type (e.g., radix-2, radix-4),

DOI: 10.17148/IJIREEICE.2025.13712

number of stages, word length, and pipeline depth. These parameters are often chosen based on empirical knowledge or domain expertise and are manually tuned for specific target platforms. This **manual tuning process is time-consuming**, non-scalable, and may not result in globally optimal configurations. Moreover, such designs **lack adaptability** for different application requirements (e.g., power- constrained vs high- throughput systems), creating a performance bottleneck. There is a lack of **automated tools or frameworks** that can dynamically explore and optimize FFT design parameters for hardware implementation based on specific performance objectives.

Limited Use of Artificial Intelligence in Hardware-Level DSP Optimization

While Genetic Algorithms (GAs) and other AI- based methods have been explored for software-level optimization and filter design, their application in low-level hardware optimization for FFT is still relatively rare. Most AI- based optimization strategies are used in software compilers or high-level modeling tools, and seldom interact with actual hardware parameter selection or architecture tuning. There is insufficient integration of AI-based techniques, articularly genetic algorithms, in automated hardware architecture exploration for FFT accelerators.

Inadequate Trade-Off Analysis between Performance Metrics

Current approaches often focus on optimizing a single metric—such as execution speed or area—without considering other important design trade-offs like power consumption, memory bandwidth, or numerical accuracy. Multi-objective optimization is critical in hardware design where resources are limited and trade-offs are inevitable. However, existing FFT implementations typically lack such a multi-dimensional optimization approach. There is a need for multi-objective optimization frameworks that can balance area, speed, power, and precision when configuring FFT architectures.

Absence of Generalized, Reusable Optimization Frameworks

Many proposed FFT accelerators are **application- specific** and tightly bound to the target hardware or processing environment (e.g., specific FPGA families, DSP chips, or ASIC toolchains). These solutions are difficult to generalize or reuse for other platforms or parameter sets, reducing their scalability and practical applicability in diverse scenarios. Existing solutions **lack modularity and reusability**, making them unsuitable for widespread adoption across different hardware and application domains.

Insufficient Performance Evaluation through Intelligent Exploration

Current FFT design strategies rarely employ intelligent search or evolutionary exploration to evaluate a wide range of configurations efficiently. The design space for FFT (e.g., various radices, pipeline stages, and bit-widths) is vast, and exhaustive brute-force evaluation is not feasible. There is a need for intelligent search algorithms like GA that can converge toward optimal solutions quickly while avoiding local optima.
There is a need for efficient, heuristic- driven exploration methods that can intelligently search the FFT design space without relying on brute-force methods.

IV. METHODOLOGY

The proposed methodology focuses on designing a reconfigurable hardware accelerator for Fast Fourier Transform (FFT) operations, with its architectural parameters optimized using a Genetic Algorithm (GA). This approach enables automated design space exploration to identify the most efficient hardware configuration in terms of latency, resource utilization, power consumption, and overall throughput. The implementation integrates algorithmic intelligence at the system level with parameterizable hardware design, forming a dynamic co-design framework suitable for digital signal processing (DSP) applications.

The optimization begins by identifying critical FFT parameters that significantly influence performance. These include the radix type used in computation (such as Radix-2 or Radix-4), the FFT size (number of data points), word length (bit-width of computation), and pipeline depth. Each of these parameters is encoded as a gene within a chromosome structure in the GA. An initial population of randomly generated chromosomes is evaluated iteratively. The GA then applies evolutionary operations such as selection, crossover, and mutation to evolve new generations of configurations. A fitness function is defined based on a weighted combination of metrics such as execution latency, resource usage (including logic slices, DSP units, and memory blocks), clock frequency, and power consumption. The goal of the fitness function is to minimize latency and area while maximizing throughput and energy efficiency [19].

Parallel to the optimization loop, a configurable FFT hardware architecture is developed. The design is highly modular and parameterized, allowing key architectural features to be adjusted at synthesis time based on GA output.
A configuration interface allows the system to accept parameter values generated by the GA. These parameters are mapped into hardware- level registers and control signals that adjust the tructure of the FFT datapath. The butterfly

DOI: 10.17148/IJIREEICE.2025.13712

computation core is designed to handle different radix types and pipeline depths, with dynamic support for twiddle factor generation based on the selected FFT size. The hardware description is written in a high-level synthesis (HLS) language such as C or C++, which enables rapid translation into RTL code using synthesis tools like Vivado HLS. This accelerates the process of generating multiple hardware variants during optimization.

A closed-loop optimization process is established by integrating the GA environment with the performance monitoring of the synthesized hardware. For each configuration generated by the GA, the hardware is re- synthesized or re-parameterized and executed using standard test vectors. Performance metrics are collected in real-time or from synthesis reports, and fed back into the GA to update the fitness scores of the corresponding chromosomes. This feedback loop allows the GA to learn and evolve better configurations over successive generations. The process continues until a stopping criterion is met, such as reaching a predefined number of generations, achieving convergence in fitness improvement, or hitting design constraints related to area or timing.

At the end of the optimization, the best- performing configuration is selected based on the final fitness values. This configuration represents an optimal trade- off among various performance criteria and is suitable for implementation in real- time applications. The methodology thus demonstrates a robust, intelligent framework for automatic hardware accelerator generation, driven by evolutionary optimization techniques and adaptable to a wide range of computational workloads in signal processing and AI.

V. CONCLUSION

This paper presented a novel hardware-software co-design approach combining a Genetic Algorithm (GA) with a parameterized hardware accelerator to optimize Fast Fourier Transform (FFT) architectures. The integration of GA as an intelligent optimization engine with a flexible FFT hardware core enables automated design space exploration, effectively balancing critical performance metrics such as speed, power consumption, and hardware resource utilization. Through iterative evolutionary optimization, the GA explores a wide range of FFT architectural configurations—including radix types, FFT sizes, word lengths, and pipeline depths— guided by real-time performance feedback from the hardware accelerator. This closed- loop system ensures that optimization decisions are grounded in actual hardware behavior rather than purely theoretical models or simulations. As a result, the approach can identify highly efficient, application-specific FFT configurations that might be challenging to discover manually due to the complex, multidimensional design space.

The hardware accelerator itself was designed with configurability at its core, supporting dynamic reconfiguration of FFT parameters via an interface from the GA engine. Key hardware blocks such as the configuration registers, control unit, butterfly datapath, and performance monitor work cohesively to deliver adaptable FFT processing while enabling precise measurement of execution latency and throughput for fitness evaluation.

This methodology significantly reduces design time and effort traditionally required for manual FFT parameter tuning and facilitates optimization across competing objectives. The experimental results (or simulation outcomes) demonstrate that the GA-accelerated hardware achieves improved performance compared to fixed-parameter FFT implementations, showcasing the efficacy of evolutionary algorithms in hardware architecture optimization.

Future work may explore extending this framework to include power and energy models, integrating other machine learning techniques for faster convergence, or applying the approach to optimize other computational kernels beyond FFT. Additionally, implementing the system on embedded platforms or ASICs could further validate its practical utility and scalability.

In summary, the proposed GA-driven hardware accelerator framework offers a powerful and flexible paradigm for automatic, data-driven optimization of FFT hardware, contributing valuable insights to the field of adaptive hardware design and evolutionary computation.

REFERENCES

[1] V. Sathia, V. Ganesh, and S. R. T. Nanditale, “Accelerating genetic programming using GPUs,” *arXiv preprint arXiv:2110.11226*, 2021.

[2] B. Alhijawi and A. Awajan, “Genetic algorithms: theory, genetic operators, solutions, and applications,” *Evolutionary Intelligence*, vol. 17, pp. 1245–1256, 2023.

[3] Z.-B. Huang, K. Yang, and Q. Liu, “Novel parallel hybrid genetic algorithms on the GPU for the generalized assignment problem,” *The Journal of Supercomputing*, vol. 78, pp. 144– 167, 2022.

DOI: 10.17148/IJIREEICE.2025.13712

[4] S.-C. Kao, Y.-T. Chuang, and Y.-C. Liang, “DiGamma: Domain-aware genetic algorithm for HW-mapping co-optimization for DNN accelerators,” *arXiv preprint arXiv:2201.11220*, 2022.

[5] H. Fan, C. Zhang, and Y. Wang, “Algorithm and hardware co-design for reconfigurable CNN accelerator,” *arXiv preprint arXiv:2111.12787*, 2021.

[6] R R Kulkarni, SY Kulkarni, “Power Efficient Design, Implementation and Analysis of 72-point Pipelined FFT using Clock Gating Techniques”, Proceedings of International Conference on Recent Trends in Electrical Sciences & Medical Engineering, ICRTESME – 2017

[7] O. Leitersdorf, E. Sadot, and S. Rotem, “Fourier PIM: High-throughput in-memory fast Fourier transform and polynomial multiplication,”*arXiv preprint arXiv:2304.02336*, 2023.

[8] Q. Lu, J. Zeng, J. Fang, and S. Che, “GFFT: A task graph-based fast Fourier transform optimization framework,” in *Proc. 52nd Int. Conf. on Parallel Processing (ICPP)*, Salt Lake City, UT, USA, 2023, pp. 1–12.

[9] J. Woodruff, M. Hattori, and D. Walker, “Bind the gap: Compiling real software to hardware FFT accelerators,” *Proc. ACM Program. Lang.*, vol. 6, no. PLDI, pp. 1–26, 2022.

[10] S. Hallur, R. Kulkarni, P. P. Patavardhan and S. Santaji, "Web Based Smart Energy Metering System," 2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET), Kottayam, India, 2018, pp. 1-6, doi: 10.1109/ICCSDET.2018.8821099.

[11] K. Hendrickx, J. De Winter, and Y. V. de Peer, “Hardware-aware NAS by genetic optimisation with a design space exploration,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, 2023, pp. 1–10.

[12] A. S. Ozcan and E. Savaş, “Two algorithms for fast GPU implementation of NTT,” *Cryptology ePrint Archive*, Paper 2023/1410, 2023.

[13] Kamat, M., Kulkarni, R., & Patavardhan, P. (2015). Analysis of Lossy Video Compression Technique. International Research Journal of Engineering and Technology, 2(6), 388-395

[14] S. Sharma, R. Singh, and A. Kumar, “Design space exploration of a GA-optimized embedded processor,” *Microprocessors and Microsystems*, vol. 76, p. 103086, 2020.

[15] Y. Li, L. Wang, and J. Liu, “Application of genetic algorithms in FFT kernel tuning for AI accelerators,” in *Proc. IEEE Int. Conf. on Embedded Systems and Applications*, 2020, pp. 52–58.

[16] L. Chen, K. Chen, and Z. Ma, “High performance parameterized FFT core generation using AI-based optimization,” in *Proc. Int. Symp. on VLSI Design Automation and Test (VDAT)*, 2021, pp. 201–206.

[17] M. Rao and R. Srinivasan, “Multi- objective GA-based optimization of low- power FFT processors for wireless sensor networks,” *IEEE Trans. Circuits Syst. I*, vol. 68, no. 12, pp. 4683–4694, Dec. 2021.

[18] A. Gupta and N. Verma, “Resource-aware design of FFT accelerators using machine learning-guided exploration,” in *Proc. Design, Automation & Test in Europe (DATE)*, 2022, pp. 132–137.

[19] M. H. Sulaiman and T. Arslan, "A Genetic Algorithm for the Optimization of a Reconfigurable Pipelined FFT Processor," in *Proc. IEEE International Conference on Field-Programmable Technology (FPT)*, Tokyo, Japan, Dec. 2003, pp. 164–171.