



Review of DIF Fast Fourier Transform Filterbank for FPGA Application

Siddhant Vishwakarma¹, Dr Vijay Yadav²

¹M.Tech Scholar, Department of Electronics & Communication Engineering, LNCT, Bhopal, India

²Associate Professor, Department of Electronics & Communication Engineering, LNCT, Bhopal, India

Abstract— Fast Fourier Transform (FFT) filterbanks play a crucial role in modern signal processing applications, particularly in systems where real-time processing and resource efficiency are paramount. The Decimation-In-Frequency (DIF) FFT algorithm has emerged as an optimal choice for many such applications due to its computational efficiency and streamlined hardware implementation. This review explores the utilization of DIF FFT filterbanks in Field-Programmable Gate Arrays (FPGAs), highlighting their potential to provide high-speed, parallel processing capabilities while maintaining resource optimization. By examining the architecture, design methodologies, and implementation challenges, this review delves into the advancements in FFT filterbanks tailored for FPGA platforms. The discussion extends to a comparative analysis of different FPGA-based DIF FFT filterbank designs, their trade-offs in terms of power consumption, latency, and scalability, and their application across fields such as telecommunications, biomedical engineering, and audio signal processing.

Keywords— DIF-FFT, DSP, FPGA, Area, Power, Delay, Xilinx.

I. INTRODUCTION

The Fast Fourier Transform (FFT) is a cornerstone algorithm in digital signal processing (DSP), enabling efficient computation of the Discrete Fourier Transform (DFT). Among its various forms, the Decimation-In-Frequency (DIF) FFT algorithm has gained significant traction due to its computational efficiency and suitability for hardware implementation. In particular, the emergence of Field-Programmable Gate Arrays (FPGAs) as versatile hardware platforms has further propelled the development of FFT-based systems, given their inherent capabilities for parallel processing, reconfigurability, and low-latency performance.

DIF FFT filterbanks have become indispensable in applications requiring real-time signal analysis and processing, such as wireless communication systems, image processing, and spectral analysis in biomedical devices. Their modular structure and computational efficiency make them particularly attractive for FPGA implementation, where resource constraints and power consumption must be meticulously managed. However, designing and implementing DIF FFT filterbanks for FPGAs is not without challenges, as it demands careful consideration of factors such as resource utilization, precision, scalability, and performance optimization.

The importance of this review lies in its focus on examining the current state of DIF FFT filterbanks tailored for FPGA applications. While substantial research has been conducted in the field of FFT algorithms and their implementation, the specific intersection of DIF FFT filterbanks and FPGA-based systems necessitates a comprehensive analysis. The convergence of these technologies has enabled significant breakthroughs in performance, allowing engineers to tackle complex signal processing tasks that were previously infeasible.

This review begins with an overview of the fundamental principles of the DIF FFT algorithm and its role within FFT filterbanks. It then delves into the architectural considerations and design methodologies specific to FPGA implementation, highlighting techniques that enhance performance while minimizing resource consumption. Subsequently, it presents a comparative analysis of various FPGA-based DIF FFT filterbank implementations, exploring their strengths and limitations across different application domains.



The discussion extends to address the challenges associated with FPGA-based DIF FFT filterbanks, such as balancing computational accuracy with hardware constraints, managing power efficiency, and ensuring scalability for evolving application requirements. To provide a forward-looking perspective, the review also explores emerging trends in the field, including the adoption of advanced FPGA technologies, integration of machine learning techniques for optimization, and leveraging hardware-software co-design approaches to push the boundaries of performance.

By synthesizing insights from existing research, this review aims to serve as a valuable resource for researchers, engineers, and practitioners seeking to harness the potential of DIF FFT filterbanks in FPGA-based systems. It underscores the transformative impact of these technologies in enabling efficient and scalable signal processing solutions, while also identifying avenues for future research and development to address existing limitations and unlock new possibilities.

II. RELATED WORK

L. H. Arnaldi et al., [1] shows the problem of optimization of multirate filterbanks is addressed. The factors that define the efficiency of these multirate systems are investigated and the implementations of structures in stages are analyzed. The latter, together with the polyphase implementations of the filters, allows obtaining optimal filterbanks in the use of resources for the FPGAs.

U. Kumar Malviya et al., [2] present a region and speed upgraded FFT module for the DSP processor. Continuous calculation of signs should meet ongoing situations and profoundly expects to make calculation faster as could be expected, thus a fast FFT proposed. Two kinds of strategies can be utilized for FFT activity: first is Decimation in time (DIT) and second is Decimation in frequency (DIF). The functional speed of these calculations relies upon the multipliers utilized for calculation. Multipliers in the FFT cycle are the vital element for the speed execution and assuming multiplier changes as far as region and speed by and large FFT processor additionally execution gets an increment. This work plan FFT processor with Vedic multiplier and new semi-pipelined Fast Fourier transform (SPFFT) with changed

duplication plan gave a give region streamlined equipment engineering. In ordinary technique, radix-2 2 had been utilized exclusively for single-way defer criticism structures. Later with many sorts of examination works radix, 22 was reached out to multipath postpone replacement (MDC) designs.

D. Massicotte et al., [3] work proposes the execution of the butterfly handling components (BPE) where the idea of the radix-r butterfly calculation has been figured out as the mix of α radix-2 butterflies carried out in equal. An effective FFT execution is doable utilizing our proposed multiplexed and pipelined BPE. Contrasted with a cutting-edge reference in view of pipelined and equal construction FFTs, and FPGA based execution uncovers that the most extreme throughput is worked on by a variable of 1.3 for a 256-point FFT and arrive at a throughput of 2680 MSps on Virtex-7. The investigation reaches out to address key execution estimations measurements like throughput, dormancy and asset usage.

L. Ache et al., [4] In this work, a fixed-point pipelined fast Fourier transform (FFT) processor is planned with radix-2 k calculation and single-way postpone criticism (SDF) engineering. Plus, the processor embraces a word length improvement procedure to diminish rationale and memory asset use. Through this procedure, the word length expected for each butterfly activity stage can be straightforwardly determined and acquired by direct recipe estimation with next to no exploratory reproduction, giving the hypothetical premise to the word length setup of the fixed-point pipelined FFT processor. The plan and execution results show that the fixed-point FFT processors utilizing the proposed word length design streamlining technique enjoy huge benefits of lower rationale asset occupation while guaranteeing the handling accuracy.

Q. Yuan et al., [5] In this work, another hybrid radices reconfigurable FFT processor is proposed, which supports registering methods of radices 2/4/8 and their blend, fulfilling 2^n ($n=6,7 \dots, 15$) point FFT computation. This processor gives a sharing to various radices FFT, and proposes the radix-4/8 address struggle free rule from radix-2 FFT address struggle free rule. The information prefetching guarantees continuous activity stream and further develops the estimation



speed. This plan carries out utilitarian check on Xilinx XC7V2000T FPGA, and plays out the design under the TSMC 28nm interaction with a functioning frequency above 800MHz and an area of 1.15 mm². The computation speed moves toward the hypothetical worth of the given the quantity of butterfly units, and the estimation exactness arrives at 10⁻⁵.

V. Harish et al.,[6] In DSP processors or different applications which use duplicate aggregate units (Macintosh) and so forth, augmentation of huge numbers is the primary bottleneck. Duplicating two n-digit double numbers requires n(n - 1) adders and n² AND entryways, which consumes additional time, power and region for enormous n since the equipment scales as the square of n in this way, there is a need to plan a paired multiplier which consumes lesser region, power and deferral yet overall, there will be tradeoff between region, power and postponement. With the contracting of innovation, we can marginally think twice about region. This work proposes a productive strategy for marked double duplication utilizing Urdhva-Tiryagbhyam method, Karatsuba calculation and effective convey select adder. Urdhva-Tiryagbhyam procedure is referred to for its low postponement.

M. Nazmy et al.,[7] The requirement for high throughput low dormancy FFT execution in correspondence frameworks has prompted the development of various models with different equal data sources and results. This work presents an original conventional hybrid engineering for equal pipelined radix-2 k FFT utilizing FF (Feed Forward) design which is otherwise called MDC (Multi-way Postpone Commutator). The proposed engineering offers another FFT information buffering calculation for an equal pipelined ordinary request inputs FFT utilizing MDC. Our engineering presents an extraordinary decrease in idleness to 25% when contrasted with best in class equal pipelined FFT structures.

A. K. Y. Reddy et al.,[8] Most usually involved calculations in computerized signal handling are Fast Fourier Transform (FFT) and Opposite Fast Fourier Transform (IFFT). In FFT butterfly unit is the essential unit which contains parts like multipliers and adders. Executing these parts assumes an exceptionally extreme part in planning of butterfly structure. To expand the presentation of FFT further an inexact radix-8

Stall multiplier utilizing surmised full recoding adder and Kogge-Stone adder is proposed. By utilizing proposed multiplier execution of the FFT is expanded by 20%. When contrasted and traditional FFT utilizing ordinary radix-8 corner multiplier.

Y. Chandu et al.,[9] In the realm of correspondence, power utilization, speed and precision are the main angles for acquiring the zenith in associating individuals. In a hurry, the handling unit in getting are time, power and region consuming, streamlining is the key element that can turn the game around. Time space to frequency area discussion is the most noteworthy need activity in the less than desirable finish of the correspondence channel. The Fast Fourier Transform (FFT) and Reverse Fast Fourier Transform (IFFT) includes butterfly Radix approach for change, in this work we examine about contrasting Radix-2, Radix-4 and Radix-8 for FFT. A new calculation is executed by the reorientation of the calculation of Radix-8, which thus decreases the mind-boggling augmentation activity. Radix-8 FFT has given persuading result which is carried out, the inertness, region and power utilization is diminished altogether.

N. Le Ba et al.,[10] Radix-2 k defer criticism and radix-K postpone commutator are the most notable pipeline engineering for FFT plan. This work proposes a clever radix-2² various defer commutator engineering using the upsides of the radix-2² calculation, like straightforward butterflies and less memory necessity. Along these lines, it is more equipment productive while carrying out parallelism for higher throughput utilizing numerous defer commutators or feed-forward information ways. Here, we propose a better information booking calculation in light of memory to dispense with energy expected to move information along the defer lines. A 1024-point FFT processor with two equal information ways is executed in 65-nm CMOS process innovation.

F. Qureshi et al.,[11] This work presents examination of fast Fourier transform (FFT) calculations for pipelined designs that can be produced by utilizing paired tree portrayal. Those calculations have different fidget factors, but the butterfly tasks stay same. Fidget variable can be carried out by various

methods, which has different equipment cost. This examination depends on fidget factors equipment cost of each FFT calculation. In results, we have shown the compromise between the equipment parts of fidget factor for chosen FFT calculations.

A. K. Singh et al.,[12] FFT, IFFT calculation is fundamentally significant in Computerized Signal Handling (DSP). This work comprises of plan of Four-point Radix-2 construction that comprises of mind-boggling duplication and addition, effective execution of these units assumes extremely basic part in plan. Stall multiplier with Kogge Stone Adder (KS) is utilized to decrease postpone in plan when contrasted with plan with comparing Corner multiplier and Wave Convey Adder (RCA). These days for VLSI frameworks chip region is frequently compromised for the deferral. The Plan is executed on Xilinx ISE Plan suite 14.7 (Family Spartan 3E and Gadget XC3S1200E) stage.

III. CHALLENGES

The implementation of Decimation-In-Frequency (DIF) FFT filterbanks on FPGA platforms offers significant advantages but also presents several challenges that need to be addressed to achieve optimal performance. These challenges can be broadly categorized into hardware, algorithmic, and application-specific aspects.

1. Resource Utilization and Optimization

- **FPGA Resource Constraints:** FPGAs have limited resources, such as logic elements, DSP blocks, and on-chip memory. Efficient utilization of these resources is critical when implementing complex DIF FFT filterbank designs.
- **Scalability:** Scaling the design to handle larger input sizes or higher throughput often leads to increased resource consumption, making it challenging to balance performance and resource usage.
- **Area vs. Speed Trade-Offs:** Optimizing for high-speed performance often increases area usage, while resource-saving designs may compromise processing speed, creating a need to strike the right balance.

2. Latency and Real-Time Processing

- **Pipelining Challenges:** Achieving low-latency, real-time performance requires effective pipelining strategies. However, designing deep pipelines for FFT computations can lead to complex control mechanisms and increased power consumption.
- **Memory Access Delays:** FFT computations require frequent access to intermediate results stored in memory. Managing these accesses efficiently to avoid bottlenecks is a significant challenge.

3. Precision and Numerical Accuracy

- **Fixed-Point vs. Floating-Point Arithmetic:** FPGAs commonly use fixed-point arithmetic for efficiency, but this can lead to quantization errors and reduced accuracy, especially for high-precision applications.
- **Error Propagation:** In filterbanks with multiple stages, errors from earlier stages can propagate and accumulate, degrading the overall output quality.

4. Power Consumption

- **Dynamic Power Usage:** The high-speed switching of FPGA logic and memory elements can lead to increased dynamic power consumption, which is critical in applications requiring energy efficiency, such as portable or embedded systems.
- **Trade-Offs Between Power and Performance:** Balancing power efficiency with high-performance requirements is particularly challenging in FPGA implementations.

5. Design Complexity

- **Parallelism Management:** Leveraging the parallel processing capabilities of FPGAs requires careful partitioning of tasks and synchronization between parallel units, which increases design complexity.
- **Hardware Mapping:** Efficiently mapping the mathematical operations of the DIF FFT algorithm onto FPGA hardware components, such as



multipliers and adders, demands in-depth expertise in both algorithm design and FPGA architecture.

6. Scalability Across Applications

- **Application-Specific Constraints:** Different applications (e.g., telecommunications, biomedical devices, audio processing) have unique requirements, such as throughput, latency, and precision, necessitating tailored FFT filterbank designs.
- **Interfacing Challenges:** Integrating the FFT filterbank with other subsystems (e.g., ADCs, DACs, or network interfaces) can introduce additional complexities in terms of data synchronization and format compatibility.

7. Implementation of Advanced Architectures

- **Higher-Order FFT Stages:** Implementing higher-order FFT stages to support large-scale applications may require innovative architectural designs, which are challenging to realize within FPGA constraints.
- **Adaptive Designs:** Modern applications often demand adaptive FFT designs that can dynamically adjust to varying input sizes or system requirements, adding further complexity to the design process.

IV. CONCLUSION

The implementation of Decimation-In-Frequency (DIF) FFT filterbanks on FPGA platforms represents a critical advancement in high-performance signal processing, offering unparalleled speed, parallelism, and flexibility for a wide range of applications. Despite challenges such as resource constraints, latency, power consumption, and design complexity, ongoing innovations in FPGA architectures, algorithmic optimization, and design methodologies are driving significant improvements in efficiency and scalability. By addressing these challenges through adaptive designs, hardware-software co-design, and emerging techniques such as machine learning-based optimization, DIF FFT filterbanks are poised to play an even greater role in enabling real-time, resource-efficient solutions for applications in telecommunications, biomedical engineering, and beyond.

This review underscores the transformative potential of this technology while highlighting opportunities for future research and development.

REFERENCES

- [1] L. H. Arnaldi, "Multistage Multirate Filterbank for FPGA Resource Optimization," in *IEEE Embedded Systems Letters*, vol. 16, no. 3, pp. 259-262, Sept. 2024, doi: 10.1109/LES.2023.3337323.
- [2] U. Kumar Malviya, "Design and Verification of High-Speed Radix-2 Butterfly FFT Module for DSP Applications," 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), 2020, pp. 37-42, doi: 10.1109/ICOEI48184.2020.9143051.
- [3] D. Massicotte, M. A. Jaber, C. Neili and M. A. Ouameur, "FPGA Implementation for the Multiplexed and Pipelined Building Blocks of Higher Radix-2k FFT," 2020 IEEE 11th Latin American Symposium on Circuits & Systems (LASCAS), 2020, pp. 1-4, doi: 10.1109/LASCAS45839.2020.9069029.
- [4] A. Li, L. Pang, Y. Zhou, C. Yang, Y. Xie and H. Chen, "Word length Optimization Method for Radix-2k Fixed-Point Pipeline FFT Processors," 2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP), 2019, pp. 1-4, doi: 10.1109/ICSIDP47821.2019.9173398.
- [5] Q. Yuan, H. Zhang, Y. Song, C. Li, X. Liu and Z. Yan, "The Design and Implementation of High Speed Hybrid Radices Reconfigurable FFT Processor," 2019 IEEE 13th International Conference on ASIC (ASICON), 2019, pp. 1-4, doi: 10.1109/ASICON47005.2019.8983511.
- [6] V. Harish and K. S., "Comparative Performance Analysis of Karatsuba Vedic Multiplier with Butterfly Unit," 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), 2019, pp. 234-239, doi: 10.1109/ICECA.2019.8821955.
- [7] M. Nazmy, O. Nasr and H. Fahmy, "A Novel Generic Low Latency Hybrid Architecture for Parallel Pipelined Radix-2k Feed Forward FFT," 2019 IEEE International



International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online) Volume 13, Issue 11, November 2024)

- Symposium on Circuits and Systems (ISCAS), 2019, pp. 1-5, doi: 10.1109/ISCAS.2019.8702144.
- [8] A. K. Y. Reddy and S. P. Kumar, "Performance Analysis of 8-Point FFT using Approximate Radix-8 Booth Multiplier," 2018 3rd International Conference on Communication and Electronics Systems (ICCES), 2018, pp. 42-45, doi: 10.1109/CESYS.2018.8724107.
- [9] Y. Chandu, M. Maradi, A. Manjunath and P. Agarwal, "Optimized High Speed Radix-8 FFT Algorithm Implementation on FPGA," 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI), 2018, pp. 430-435, doi: 10.1109/ICOEI.2018.8553791.
- [10] N. Le Ba and T. T. Kim, "An Area Efficient 1024-Point Low Power Radix-22 FFT Processor With Feed-Forward Multiple Delay Commutators," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 65, no. 10, pp. 3291-3299, Oct. 2018, doi: 10.1109/TCSI.2018.2831007.
- [11] F. Qureshi and J. Takala, "Twiddle factor complexity analysis of Radix-2 FFT algorithms for pipelined architectures," 2017 51st Asilomar Conference on Signals, Systems, and Computers, 2017, pp. 1034-1037, doi: 10.1109/ACSSC.2017.8335506.
- [12] A. K. Singh and A. Nandi, "Design of four point Radix-2 FFT structure on Xilinx," 2017 International Conference on Intelligent Computing and Control (I2C2), 2017, pp. 1-4, doi: 10.1109/I2C2.2017.8321928.