A Review of the Design & Implementation of Efficient Architecture For DFT

Maithreyi[1], B. Vinay[2], C. Adarsha B V [3], D. Harsha B V [4], E. Mr. Ranjith H D [5]
[1,2,3,4] *Student, Dept. of ECE, Mangalore Institute of Technology & Engineering, Karnataka, India*
[5] *Senior Assistant Professor, Dept. of ECE, Mangalore Institute of Technology & Engineering, Karnataka, India*

Abstract: **The Discrete Fourier Transform (DFT) is a critical tool for digital signal processing, used to convert signals from the time domain to the frequency domain. This project aims to design an efficient DFT architecture using Verilog hardware description language and validate it using Cadence tools. Over the years, DFT implementation has evolved significantly. Initially, DFT was performed using general-purpose processors, but the demand for real-time processing has led to the exploration of specialized hardware solutions.**

The Fast Fourier Transform (FFT) has been a major advancement, reducing computational complexity from O(N^2) to O(N log N). Current trends are focused on making FFT algorithms faster in hardware, such as in Field-Programmable Gate Arrays (FPGAs) and Application-Specific Integrated Circuits (ASICs). This project seeks to capitalize on these advancements by implementing a 2-point radix-2 Decimation-In-Time FFT algorithm. It will utilize sophisticated techniques like carry-save adders and Booth multipliers in its system architecture for improved performance. The project's scope involves designing a 2-point Decimation-In-Time FFT architecture with a bit-width of [3:0] and considering only real values, excluding the imaginary part in the testbench.

Cadence tools will be employed to perform software simulation for validation, ensuring accuracy and efficiency. This work aims to provide an efficient hardware solution for real-time applications in the digital signal processing domain, pushing the boundaries of current FFT implementation. The simulation was performed using Cadence tools. The methodology encompasses the design of carry-save adders and booth multipliers, the design of the Multiply-Accumulate (MAC) unit, and the architecture design for DFT.

Index Terms: **Discrete Fourier Transform (DFT), Digital Signal Processing (DSP), Verilog HDL, Cadence tools, Fast Fourier Transform (FFT), Computational Complexity, Real-Time Processing, FPGA, ASIC,2-point Radix-2 DIT-FFT, Carry-Save Adders, Booth Multipliers, MAC Unit, Hardware-Efficient Architecture, Software Simulation, Real Values, Bit-Width, Signal Transformation, Specialized Hardware Solutions, Hardware Descriptive Language**

I. INTRODUCTION

The DIT-FFT algorithm is an efficient method for computing the Fast Fourier Transform. It works by breaking down a sequence of time-domain samples into smaller subsequences, splitting the data into even-indexed and odd-indexed elements, and applying the FFT algorithm to these smaller groups. The results are then combined to obtain the final frequency domain representation. This divide-and-conquer approach reduces computational complexity from O(N^2) to O(N log N), making it highly efficient for processing large sets of data in real-time applications. DIT-FFT is commonly used in digital signal processing for real-time signal analysis and filtering, as well as in telecommunication for efficient modulation and demodulation, particularly in OFDM and other systems.

II. LITERATURE SURVEY

[1] Keerthan (2018) proposed a new bit-slicing-based scheme that optimizes data flow and processing stages involved in the FFT algorithm to reduce its power consumption and computational complexity. This would thus significantly reduce hardware complexity while enhancing the overall efficiency of FFT computation. Experimental results showed tremendous enhancements in both power efficiency and computational performance over traditional FFT implementations, which should make the modified FFT one of the potential solutions for power-sensitive and high-performance signal processing applications.

[2] Meng (2023) presented a chipset-based architecture to efficiently process the spectrum using the FFT algorithm. In this design, higher flexibility in integration and scalability has been achieved by enhancing computational throughput and energy efficiency. The proposed chipset design will permit efficient resource management and will have adaptable usages in many spectrum processing requirements. Comparative analyses show better performance in terms of speed, power consumption, and area efficiency.

[3] Elango (2022) presented a new digital logic design, optimizing bit reversal and address generation for FFT processors to reduce latency and power consumption. Their approach, enthroned by new algorithms and circuit designs, could be very useful in high-speed and real-time signal processing applications. It demonstrates that, compared to conventional techniques, the proposed logic shows an enhancement in computational speed and a reduction in hardware complexity with reduced energy usage; hence, it has the potential to improve the performance of FFT processors in many digital signal processing applications.

[4] Priya (2021) investigated the improvement of FFT processor design with area efficiency using pipelined architectures. They exploited algorithms like Radix-2 and Radix-4, parallel processing, and efficient memory management to reduce hardware complexity and improve computational performance. Their experimental results show that their proposed pipelined FFT architecture greatly reduced chip area and power consumption and thus advanced the design of FFT processors.

[5] Garrido (2021 developed the most innovative FFT design using constant multipliers for enhancing the computational efficiency of the system. Here, general multiplier hardware is replaced by pre-computed constant values, hence reducing the complexity and power consumption. This approach simplifies It designs of the hardware of IT and accelerates FFT. Experimental results demonstrate significant performance and energy efficiency to meet requirements in embedded systems and real-time signal processing applications.

[6] Padma (2020) surveyed high-performance FFT architectures for digital signal processing applications, tracing their evolution from conventional designs to optimized implementations. The authors present outlines of the design considerations, optimization techniques, and performance metrics to achieve high-performance FFT processors that can help researchers and practitioners develop efficient and scalable FFT architectures.

[7] Elango (2022) In terms of minimizing the core area and energy consumption for effective VLSI design, the work of the author found relevance for wireless communication. Advanced optimization and associated resource-sharing techniques exploited in this project led to better computational efficiency at higher throughputs. In this way, the proposed FFT/IFFT core evidenced giant chip area and energy efficiency improvements, very useful in all MIMO-OFDM systems for modern design and to guarantee performance and resource use balancing.

[8] Kavitha (2019) presented a novel architecture in FFT processors that integrates an advanced version of the CORDIC algorithm with the Radix-2r FFT algorithm for flexibility and efficiency of the processor, abolishing conventional problems of accuracy and speed of convergence. It further allows the Radix-2r algorithm to efficiently process larger data sets, improving computational throughput while reducing latency. It confirms that the design methodology and implementation results represent considerable improvements in resource utilization and processing speed for FPGA-based FFT processors over existing ones.

[9] Daniel (2019) developed flexible FFT architectures for different applications. The focus of their work was on parameterizable designs that scale easily in size and accommodate diverse hardware constraints. These authors achieved higher throughput and reduced latency by using innovative memory management schemes and architectural optimizations. Their designs are faster and more area-efficient compared to traditional ones.

[10] Sarada (2018) proposed a new FFT processor with a low power consumption rate along with high throughput. The feedforward architecture, as adopted here, is more suitable for pipelining than the traditional

feedback-based structures. The processor enhances the power efficiency and processing speed of the system through the reduction in computational complexity and by adopting the low-power circuit design methodology. Results show that it outperforms previous designs in energy consumption and throughput, hence making it very suitable for modern low-power and high-performance signal processing applications.

[11] Chen (2018) described the VLSI design of fixed-point reconfigurable FFT processors targeting the cost-effective trade-off with precision. In the same line of research, the authors have proposed some new methodologies for the improvement of performance and flexibility, trying to meet the challenge of cost-effective solutions in these demanding applications. It maintains accuracy while providing in-depth information on new design paradigms of reconfigurable FFT processors.

[12] Qadeer (2023) proposes a VLSI-based, new Radix-2 DIT FFT algorithm that optimizes computational speed and hardware efficiency for FFT processing. The approach makes use of parallel processing and pipelining to increase throughput by reducing latency and demonstrates considerable improvement in performance over traditional FFT techniques.

III. METHODOLOGY

Implementation of DFT is structured as:

A. Design of Carry-Save Adder (CSA) designed:

A carry-save adder (CSA) speeds up multi-operand additions by saving carries instead of propagating them immediately. It computes partial sums and carry-outs independently, which is efficient for operations involving multiple additions, like in multiplication.

B. Design of Booth Multiplier:

Booth's multiplier efficiently multiplies binary numbers by encoding sequences of bits in the multiplier, reducing the number of partial products and handling both positive and negative numbers.

C. Design of MAC unit:

The MAC unit combination of Carry-Save Adder and Booth multiplier can perform complex operations by multiplying and accumulating efficiently.

D. Design and implementation of an efficient architecture for 2-DFT:

For a 2-point DFT, a MAC (Multiply-Accumulate) unit of Carry-Save Adder and Booth Multiplier is used to efficiently perform the computation. The Radix-2 DIT FFT method is applied to simplify the process into two stages: one for addition and one for subtraction of inputs.

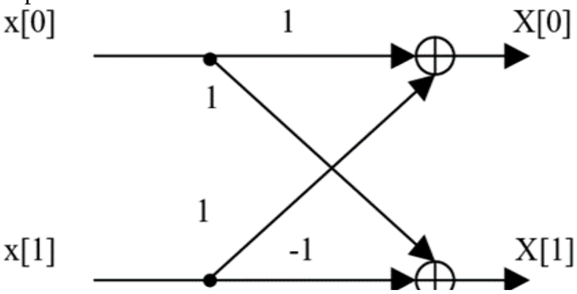

Fig .1 Signal flow graph of 2-point DIT-FFT.

The 2-point DIT-FFT algorithms break down larger DFT computations into smaller, more manageable computations. In this case, it illustrates how the inputs are combined through addition and subtraction to produce the DFT outputs efficiently.

IV. RESULT

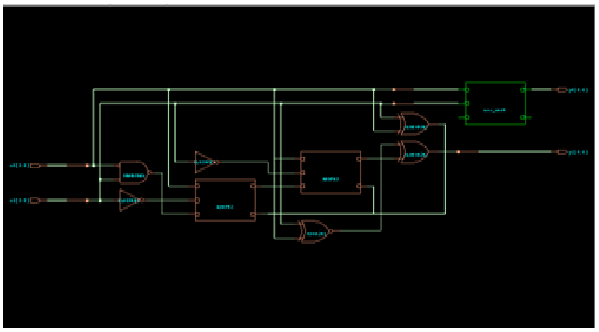

Fig.2 RTL Schematic of 2-point DIT-FFT architecture using CSA and Booth multiplier

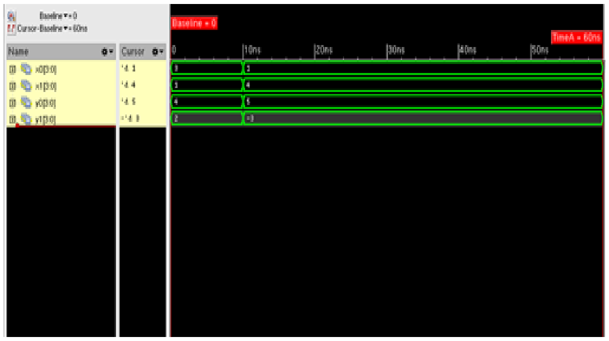

Fig.3 The waveform of 2-point

The circuit schematic shows a structural description of how inputs get processed to produce an output. It gives a temporal view of the states of the signals and thus helps in the validation of the functionality of the circuit. Specifically, it contributes to the understanding of changes over time in the output signals x [0], x [1], y [0], and y [1] with regard to inputs A, B, and C.

Parameters of the proposed model:

Parameters	Proposed Model
Power	91.09%
Cell count	171
Area	1019.099

The proposed model will have a power efficiency of 91.09%, a cell counts of 171, and an area of 1019.099 units, so at first glance, this will be a robust and efficient design. A high-power efficiency would guarantee that the model operations are effective; the cell count, at a medium level, ensures balanced complexity; and the area metric informs about the scalability of the model and, thus, the resources it would require.

Such deep analysis allows for insight into how further optimization may trade off against the model, making practical decisions around deploying the model to production, and ensuring its performance meets expectations without losing its resource efficiency.

V. CONCLUSION

In this case, a MAC unit using a Carry-Save Adder and a Booth multiplier will be a good selection for the design of the DIT-FFT (Decimation in Time Fast Fourier Transform) architecture. On one hand, the carry-save adder supports parallel additions that reduce delay, while the simple design enables compact integration. On the other hand, the booth multiplier reduces area consumption by efficiently handling partial products. It combines the Carry-Save Adder and Booth Multiplier in the MAC unit, balancing fast computations' speed and resource efficiency with a compact circuit to enhance the DIT-FFT architecture.

REFERENCE

[1] Qadeer, S., Keerthan, H., Azeemuddin, S., & Khan, M. Z. A. (2023). Low Power and Complexity Implementation of the Modified FFT with a New Bit-Slicing Scheme. Journal of Institution of Engineers (India) Series B, 104(6), 1285–1302.doi:10.1007/s40031-023-00923- x

[2] B. Meng, G. Shan, and Y. Zheng, “Design of Spectrum Processing Chiplet Based on FFT Algorithm,” Micromachines, vol. 14, no. 2, p. 402, Feb. 2023, doi: 10.3390/mi14020402.

[3] Elango, K., & Muniandi, K. (2022). A Novel Digital Logic for Bit Reversal and Address Generations in FFT Computations. Wireless Personal Communications, 128(3), 1827–1838.doi:10.1007/s11277-022-10021-8

[4] Priya, A. L., & Deepthi, V. M. (2021). Design and implementation of area-efficient pipelined FFT processor. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(11), 7245-7251.

[5] Garrido, M., & Malagon, P. (2021). The Constant Multiplier FFT. IEEE Transactions on Circuits and Systems. I, Regular Papers, 68(1), 322–335.doi:10.1109/tcsi.2020.3031688

[6] Padma, C., Jagadamba, P., & Reddy, P. R. (2020). Implementation of high-performance FFT architecture for DSP applications. International Journal of Advanced Science and Technology, 29(3).

[7] Elango, K., & Muniandi, K. (2019). VLSI implementation of an area and energy-efficient FFT/IFFT core for MIMO-OFDM applications. Annals of Telecommunications/Annales Des Télécommunications, 75(5–6), 215–227.doi:10.1007/s12243-019-00742-6

[8] Kavitha, M. S., & Rangarajan, P. (2020). An Efficient FPGA Architecture for Reconfigurable FFT Processor Incorporating an Integration of an Improved CORDIC and Radix-2r Algorithm. Circuits, Systems, and Signal Processing, 39(11), 5801–5829.doi:10.1007/s00034-020-01436-4

[9] D. Valencia and A. Alimohammad, “Compact and high-throughput parameterisable architectures for memory-based FFT algorithms,” IET Circuits, Devices & Systems, vol. 13, no. 5, pp. 696–703, Jul. 2019, doi: 10.1049/iet-cds.2018.5556.

[10] V. Sarada, T. Vigneswaran, and J. Selvakumar, “Low-power and high-throughput 128-point feedforward FFT processor,” Cluster Computing, vol. 22, no. S6, pp. 13397–13404, Feb. 2018. doi:10.1007/s10586-018-1918-4

[11] Xiao, H., Yin, X., Wu, N., Chen, X., Li, J., & Chen, X. (2018). VLSI design of low-cost and high-precision fixed-point reconfigurable FFT processors.

IET Computers & Digital Techniques, 12(3), 105–110.doi:10.1049/iet-cdt.2017.0060.

[12] H. K. Samudrala, S. Qadeer, S. Azeemuddin, and Z. Khan, “Parallel and Pipelined VLSI Implementation of the New Radix-2 DIT FFT Algorithm,” Dec. 2018, doi: 10.1109/ises.2018.00015.