ASIC Implementation of FFT Processor

Mr. M. Gnanesh Goud[1], M. Vaishnavi[2], M. Manga[3], P. Deepak[4], P. Pavan[5]
[1]*Assistant Professor, Dept. of ECE, TKR College of Engineering and Technology*
[2,3,4,5] *Student, Dept. of ECE, TKR College of Engineering and Technology*

***Abstract*—This project implements an 8-point Fast Fourier Transform (FFT) processor using ASIC design. In Stage-1, the design is developed in Verilog/VHDL using a radix-2 DIT architecture and verified through simulation. In Stage-2, it is synthesized and implemented through ASIC backend processes like placement, routing, and timing analysis. The final design achieves efficient performance, low area, and low power, making it suitable for real-time DSP applications.**

***Index Terms*—RTL Modelling, Backend Implementation**

I. INTRODUCTION

Digital Signal Processing (DSP) plays a major role in modern electronic systems such as communication, radar, biomedical equipment, and multimedia devices. Signals in real-world applications are usually processed in digital form for better accuracy, reliability, and flexibility.

One of the most important operations in DSP is converting a signal from the time domain to the frequency domain. This conversion helps in analysing the frequency components present in the signal. The mathematical tool used for this purpose is the Discrete Fourier Transform (DFT). However, direct implementation of DFT requires a large number of computations, making it inefficient for high-speed applications.

To overcome this limitation, the Fast Fourier Transform (FFT) algorithm was developed. FFT significantly reduces computational complexity, making real-time signal processing possible.

With the rapid growth of wireless communication systems, real-time signal processing has become essential. Applications such as OFDM systems, 5G communication, radar imaging, and biomedical monitoring require high-speed and low-power FFT processors.

Software-based FFT implementations are flexible but slower for high-performance systems. FPGA-based implementations provide flexibility but consume more area and power compared to ASIC designs.

ASIC (Application-Specific Integrated Circuit) implementation offers:

- Higher speed performance
- Lower power consumption
- Optimized silicon area
- Better reliability of mass production

II. METHODOLOGY

The methodology for implementing the 8-point FFT processor follows a structured ASIC design flow divided into two major stages, ensuring accuracy, efficiency, and optimization. In Stage-1, the design is described at the Register Transfer Level (RTL) using Verilog HDL/VHDL, where a radix-2 decimation-in-time (DIT) FFT architecture is selected for its simplicity and computational efficiency. The input sequence is decomposed into smaller parts using butterfly units, which perform complex addition, subtraction, and multiplication operations with twiddle factors. Proper pipelining and data flow control techniques are applied to improve throughput and reduce latency. Functional simulation is then carried out using test benches, and the obtained results are compared with mathematically computed FFT outputs to verify correctness.

In Stage-2, the verified RTL is synthesized into a gate-level netlist using standard cell libraries, focusing on optimizing area, timing, and power. The design then proceeds through physical design steps, including floor planning to define chip layout, placement of standard cells, and clock tree synthesis (CTS) to ensure balanced clock distribution with minimal skew. Routing is performed to establish interconnections between components while minimizing delays and

congestion. After routing, Static Timing Analysis (STA) is conducted to verify that all timing constraints such as setup and hold times are satisfied. Power analysis and optimization techniques are also applied to reduce dynamic and leakage power consumption. Finally, post-layout verification, including Design Rule Check (DRC) and Layout Versus Schematic (LVS), ensures that the fabricated design matches the intended schematic and adheres to manufacturing rules. This comprehensive methodology results in a high-performance, low-area, and low-power FFT processor suitable for real-time DSP applications.

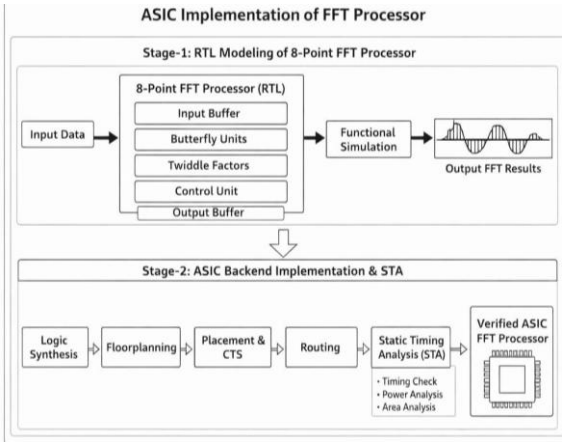

Figure:1: ASIC Implementation of 8-point FFT Processor

The implementation of the FFT processor follows a systematic ASIC design flow. Initially, the FFT algorithm (radix-2 DIT) is designed and modelled at the RTL level using Verilog/VHDL, incorporating butterfly units, twiddle factors, and control logic. The design is functionally verified through simulation by comparing outputs with theoretical FFT results. After verification, the RTL is synthesized into a gate-level netlist. The design then proceeds through physical design stages including floor planning, placement, clock tree synthesis (CTS), and routing. Finally, Static Timing Analysis (STA), power analysis, and physical verification (DRC/LVS) are performed to ensure the design meets timing, area, and power requirements.

III. IMPLEMENTATION DESIGN

The design of the 8-point FFT processor is based on a radix-2 decimation-in-time (DIT) architecture, which efficiently decomposes the Discrete Fourier Transform (DFT) into smaller computational stages. The overall system consists of three main stages, where each stage contains butterfly units that perform complex addition and multiplication operations.

In the design phase, the architecture is described at the Register Transfer Level (RTL) using Verilog HDL/VHDL. The core component of the design is the butterfly unit, which takes two complex inputs and produces two outputs using twiddle factor multiplication. Twiddle factors are precomputed and stored to reduce computational complexity. Pipeline registers are introduced between stages to improve throughput and enable high-speed operation. Control logic is also designed to manage data flow and synchronization across different stages of the FFT processor.

The input data is applied in a sequential manner, and bit-reversal logic is used to arrange the input samples in the correct order required by the DIT algorithm. Each stage progressively computes partial FFT results, and the final stage produces the complete frequency-domain output.

In the implementation phase, the RTL design is synthesized using standard cell libraries targeting ASIC technology. The synthesis process optimizes the design for area, power, and timing constraints. Following synthesis, physical design steps are carried out, including floor planning, placement of standard cells, clock tree synthesis (CTS), and routing. These steps ensure proper connectivity and timing closure of the design.

Static Timing Analysis (STA) is performed to verify that all timing constraints are met under different conditions. Post-layout simulation is conducted to validate the functional correctness after physical implementation. Power and area reports are generated to evaluate the efficiency of the design.

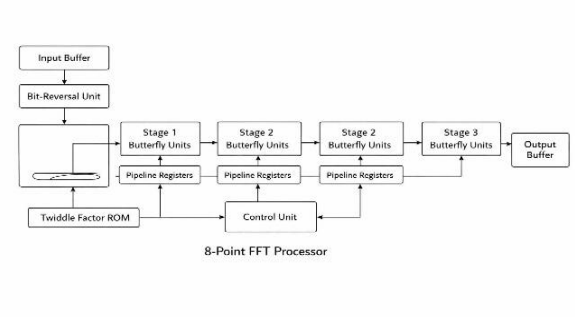

Figure:2: 8-point FFT processor (Radix-2 DIT Architecture)

The block diagram represents the architecture of an 8-point FFT processor based on the radix-2 decimation-in-time (DIT) algorithm. The input data is first stored in the input buffer and then rearranged using a bit-reversal unit to match the required processing order. The computation is performed in three stages of butterfly units, where each stage progressively transforms the time-domain input into frequency-domain output. Twiddle factors required for complex multiplication are stored in a ROM and accessed during processing. Pipeline registers are inserted between stages to improve speed and enable parallel data processing. A control unit manages the overall operation and synchronization of data flow. Finally, the processed data is collected in the output buffer, producing the FFT results.

IV. RESULT & DISCUSSION

The ASIC implementation of the 8-point FFT processor was successfully completed and verified across both design and backend stages. Functional simulation results confirm that the processor produces accurate frequency-domain outputs, closely matching the theoretical FFT values. This validates the correctness of the radix-2 decimation-in-time (DIT) architecture and butterfly computations.

Post-synthesis results indicate that the design meets the required timing constraints with efficient utilization of hardware resources. The incorporation of pipeline registers significantly improves the operating speed by enabling parallel processing across different stages. Static Timing Analysis (STA) confirms that there are no setup or hold violations, ensuring reliable high-speed operation.

From the physical design perspective, the layout is optimized to achieve a balance between area and performance. The reported core area and die area demonstrate a compact implementation suitable for ASIC realization. Power analysis shows low internal and switching power consumption, along with negligible leakage power, making the design energy-efficient.

The use of a Twiddle Factor ROM reduces computational complexity and contributes to area optimization. Additionally, proper floor planning and clock tree synthesis ensure uniform clock distribution and minimal delay variations across the chip.

Overall, the results demonstrate that the proposed FFT processor achieves high performance with low power and area overhead. This makes it well-suited for real-time digital signal processing applications such as wireless communicate on, image processing, and embedded systems.

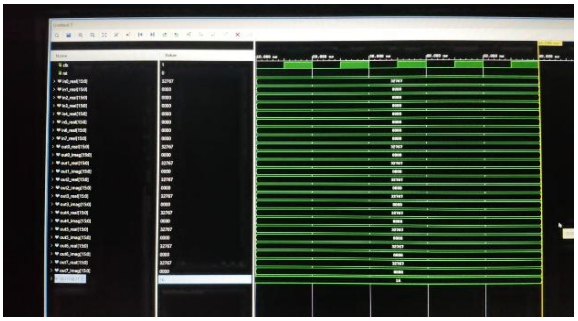

Figure:3: RTL Simulation Waveform of 8-point FFT Processor

The RTL simulation waveform of the 8-point FFT processor. The clock signal drives the sequential operation, while the reset initializes the system. The input signals consist of real-valued data, where one input has a maximum value (32767) and the remaining inputs are zero. The output signals display the corresponding FFT results, with real parts showing consistent values and imaginary parts remaining zero, indicating correct computation for the given input pattern. The stable and synchronized waveforms confirm proper data flow, correct functionality of butterfly operations, and successful verification of the FFT design at the RTL level.

Category	Parameter	Value
Area Report	Core Area	320107μm^2
	Die Area	0.339mm^2
Power Report	Internal power	0.875µW
	Switching Power	1.87 µW
	Leakage Power	7.68×10^{-8} µW
Physical Verification	DRC	0 Violations
	LVS	Matched

V. CONCLUSION

The ASIC implementation of the 8-point FFT processor has been successfully designed and verified

using a radix-2 decimation-in-time (DIT) architecture. The RTL design, developed using Verilog/VHDL, was functionally validated through simulation, confirming accurate FFT computation. The use of butterfly units and pipeline stages enabled efficient data processing and improved system throughput.

The backend ASIC implementation, including synthesis, floor planning, placement, clock tree synthesis, and routing, was completed successfully. Static Timing Analysis (STA) confirmed that all timing requirements were satisfied without violations. The design achieved a compact area, low power consumption, and clean physical verification results with zero DRC violations and successful LVS matching.

Overall, the proposed FFT processor demonstrates an optimized balance between performance, area, and power, making it suitable for real-time digital signal processing applications such as communication systems, image processing, and embedded platforms.

During the ASIC backend implementation, all major physical design steps—including synthesis, floor planning, placement, clock tree synthesis, and routing—were completed successfully. Static Timing Analysis (STA) confirmed that the design meets all timing constraints without setup or hold violations. The physical verification results, including zero Design Rule Check (DRC) violations and successful Layout Versus Schematic (LVS) matching, indicate a robust and manufacturable design.

The obtained area and power reports highlight that the processor is compact and energy-efficient, with low internal, switching, and leakage power consumption. This makes the design highly suitable for low-power and high-performance applications such as wireless communication systems, real-time signal processing, radar, and biomedical devices.

REFERENCES

[1] A. V. Oppenheim and R. W. Schafer, Discrete-Time Signal Processing, 3rd ed., Pearson, 2010.

[2] S. K. Mitra, Digital Signal Processing: A Computer-Based Approach, 4th ed., McGraw-Hill, 2011.

[3] J. G. Proakis and D. G. Manolakis, Digital Signal Processing: Principles, Algorithms, and Applications, 4th ed., Pearson, 2007.

[4] K. K. Parhi, VLSI Digital Signal Processing Systems: Design and Implementation, Wiley, 1999.

[5] B. Razavi, Design of Analog CMOS Integrated Circuits, McGraw-Hill, 2001.

[6] N. H. E. Weste and D. Harris, CMOS VLSI Design: A Circuits and Systems Perspective, 4th ed., Pearson, 2011.

[7] M. Keating et al., Low Power Methodology Manual for System-on-Chip Design, Springer, 2007.

[8] S. He and M. Torkelson, "A New Approach to Pipeline FFT Processor," IEEE Transactions on Signal Processing, vol. 44, no. 12, pp. 3081–3091, 1996.

[9] L. R. Rabiner and B. Gold, Theory and Application of Digital Signal Processing, Prentice Hall, 1975.

[10] P. Duhamel and M. Vetterli, "Fast Fourier Transforms: A Tutorial Review and a State of the Art," Signal Processing, vol. 19, no. 4, pp. 259–299, 1990.

[11] Xilinx Inc., "FFT v9.1 Logi CORE IP Product Guide," 2021.

[12] Synopsys, "Design Compiler and IC Compiler II User Guide," 2022.