SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V13I1P101

Volume 13 Issue 1, 1-10, January 2026
© 2026 Seventh Sense Research Group®

Original Article

A High Performance 128 to 2048-Point Pipeline SDF-FFT Architecture based on DPRAM for LTE and WiMAX Systems using Modified Low-Power Carry Select Adder

Nancharaiah Vejendla[1*], R. Ramana Reddy[2], N. Balaji[3]

[1]*Department of Electronics & Communication Engineering, JNTU Kakinada, A.P., India.*
[2]*Department of Electronics & Communication Engineering, JNTUACEK(A), Kalikiri, A.P, India.*
[3]*Department of Electronics & Communication Engineering, UCOEK(A), JNTU Kakinada, A.P, India.*

[*]*Corresponding Author : nanch84@gmail.com*

Received: 01 November 2025 Revised: 03 December 2025 Accepted: 02 January 2026 Published: 14 January 2026

Abstract* - Fast Fourier Transforms (FFTs) are essential in communications, signal processing, computer vision, measurements, and instrumentation. The time-domain signals are mapped into frequency-domain representations by FFT for the number of points specified. In telecommunications, FFT is widely used in Orthogonal Frequency Division Multiplexing (OFDM) to map various frequency components to perform the modulation and demodulation processes before transmission or reception. Owing to the high computational complexity, the FFT implementation for real-time applications is quite difficult. The FFT can be efficiently implemented in Field-Programmable Gate Array Logic (FPGA) for superior performance compared to embedded applications. This paper presents a Novel Semi-Pipelined FFT Architecture based on Single-Path Delay Feedback (SP-FFT-SPDT-DPRAM) with DP Random-Access Memory (RAM) for variable-point FFT. Using DP-RAM, the read and write operations for two locations can be performed instead of using single-port RAM. A low-power, low-area-based carry-select adder is utilized to minimize the power consumption and area in complex adders and multipliers. The projected design, implemented using 180 nm CMOS technology, demonstrates improved performance compared to existing architectures.

Keywords* - Discrete Fourier Transform (DFT), TFs (TF), Complex Number Multiplication, Fast Fourier Transform (FFT), Butterfly Structure. Single-path Delay Feedback (SDF), Semi-Pipelined Architecture, Processing Element (PE), and Digital Signal Processing (DSP).

1. Introduction

To calculate the frequency spectrum of finite-duration signals, the Discrete Fourier Transform (DFT) is an essential tool in modern Digital Signal Processing (DSP). It is also essential to modern telecommunication technologies like Long-Term Evolution (LTE), Digital Video Broadcasting, and Worldwide Interoperability for Microwave Access (WiMAX) [1]. With a temporal complexity of $O(N^2)$ for an N-point transform, DFT is computationally demanding and time-consuming for big datasets. J.W. Tukey and J.W. Cooley presented the FFT as a solution to these problems. Although there are numerous varieties, their work was first distributed in 1965, and it is believed that Gauss portrayed the method as quite a while in the past, as far back as 1805 [2]. Now, FFT is a vital component in DSP, sensor signal processing, Digital Communication, and radar signal processing; basically, it is an algorithmic optimization of DFT. The FFT algorithm computes the DFT with a diminished count of arithmetical activities from $O\ (N^2)$ to $N/2\ (log2\ N)3$ by exploiting the symmetry and periodicity of TFs. The primary advantage of the FFT technique over traditional DFT methods is its significantly faster calculation speed. The FFT technique consistently reduces calculation time significantly. There are a few kinds of FFT calculations. Radix-2 is a useful calculation for determining the FFT using the DIT strategy [3]. Traditionally, DSP calculations are actualized through generic programmable DSP chips as an alternative for more applications, special-purpose mounted DSP chipsets, or FPGAs, which are primarily reconfigurable semiconductor chips. FFT computing involves both addition and multiplication. Since multipliers are relatively sluggish hardware components, their performance impacts the effectiveness of the FFT hardware. Frequently used hardware multipliers include the Array Multiplier, Booth Multiplier, Serial Multiplier, Booth-Encoded Wallace Tree Multiplier, and Wallace Tree Multiplier [4]. For fixed-size FFT processors, several models have been proposed during the last 20 years. These models fall into two categories: pipelined and memory-based. The memory-based design provides a low-power arrangement; however, this methodology experiences extensive latency and requires additional buffer space for framework synchronization. Pipelined engineering, together with SDF pipeline FFT design and Multipath Delay Commutator (MDC) design, yields high throughput; however, it requires many hardware assets simultaneously [5]. FFT is a computationally intensive algorithmic rule within the physical layer of an OFDM system that converts information between the time and frequency domains. Several OFDM systems, such as 4G LTE/LTE-A, 5G, and Wireless Native Space Networks (WLAN), require FFT of size power-of-two. LTE transmission precoding requires non-power-of-two DFTs from 12 to 2400. Within the upcoming 5G communication, FFT remains a critical component for all waveform candidates; its computation speed must be sufficiently high to meet the data rate requirements for 5G. Consequently, future multimode base stations will require FFT processors capable of supporting various DFT sizes and high-speed FFT operations [6]. The existing architectures focus on optimizing either the speed or the power. The proposed research work focuses on improving the speed and power dissipation by utilizing the pipelined architecture with DP-RAM and a low-power adder.

This paper presents an FFT processor optimized for area, efficiency, and processing speed. This paper proposes a novel design aimed at enhancing performance in terms of Power, Performance, and Area (PPA), including reduced area, lower power consumption, and increased speed. The paper is organized as follows. Section 1 focuses on an introduction to the FFT architecture and applications. Section 2 reviews the work contributed by various researchers. Section 3 presents the proposed FFT architecture in a detailed manner. Section 4 discusses the implementation results of the proposed architecture. Finally, Section 5 concludes the paper.

2. Prior Work

Numerous studies have been conducted to implement FFT computations in real-time applications. All studies focused on reducing the area power and speeding up the processor by providing the maximum frequency. More research has been conducted to implement the FFT algorithm in FPGAs and ASICs instead of embedded implementations because of the speed of FPGAs and ASICs compared to embedded and DSP devices.

Specifically designed to support 16 to 4096-point FFTs and 12 to 2400-point DFTs targeting high throughput requirements, Shaohan Liu et.al. proposed a high-flexibility, low-latency, memory-dependent, high-throughput programmable FFT processor for 4G, WLAN, and future 5G systems. To balance performance and hardware cost, the authors adopted a 16-way parallel data-path memory-based architecture. To enhance hardware efficiency, several optimizations were introduced. A reconfigurable butterfly unit was designed to maximize resource reuse, which is capable of executing eight radix-2, four radix-3 (or) radix-4, two radix-5 (or) radix-8, or one radix-16 computation per clock cycle. In addition, Twiddle Factor (TF) multipliers were optimized using various strategies, including a modified Coordinate Rotation Digital Computer (CORDIC) scheme, which significantly reduced hardware complexity while supporting both FFT and DFT operations. The processor, which was created using 65 nm CMOS technology, takes approximately 1.46 mm² area, 68.640 mW of power consumption, and a Signal-To-Quantization Noise Ratio (SQNR) of 66.10 dB It attains a 972.0 MS/s throughput at 250MHz for a 4096-point FFT. The proposed design outperforms existing solutions in terms of normalized throughput per area, making it a strong candidate for future multimode base station implementations [6].

An area-efficient pipelined FFT processor by Chu Yu and Mao-Hsu Yen that enables 128–2048–/1536-point computations for LTE and WiMAX systems. A unique Single-Path Delay Feedback (SDF) architecture in their design can effectively handle FFT computations with 128, 256, 512, 1024, 1536, and 2048 points. Similar to force utilization, the suggested strategy engages the 1536-point FFT using a less complicated exertion calculation plan, which drastically lowers hardware expenses. Related to the recently referenced 1536-point FFT calculation plot, the projected arrangement incorporated a capable three-sort-out SDF pipeline engineering on which to execute a radix-3 base FFT. The novel radix-3 FFT processor with an SDF pipeline reduces the eccentricities of the resulting hardware compared to conventional designs, streamlines its data flow, and is simple to control. To reduce the memory space needed for a 1536-point FFT calculation, this proposal also included a hardware-sharing component. To lower the memory space requirements of the suggested 1536-point FFT calculation, this proposal also incorporates a hardware-sharing scheme. 90 nm CMOS technology was used to implement the suggested structure [7].

For mobile WiMAX applications, Chu Yu presented a 128/512/1024/2048-point pipeline FFT/IFFT architecture with a low-power radix-2 pipeline FFT design. For an effective hardware implementation, the architecture uses a Single-Path Delay Feedback (SDF) structure. To minimize power consumption, the design employs low-power single-port registers for delay buffers in the embedded memory, enabling significant energy savings. The overall design occupied approximately 140 K logic gates and consumed approximately 18 mW at 20 MHz. The architecture also includes reconfigurable complex multipliers and bit-parallel complex multipliers to cut down on the amount of ROM needed to store Twiddle Factors (TFs). The design achieved an 87.5% reduction in ROM size by effectively optimizing both area and power consumption, which is particularly beneficial for long-point FFT/IFFT processors [8].

By utilizing Twiddle Factor (TF) scaling, Harsha Keerthan et.al. proposed a concurrent and pipelined VLSI implementation of a unique radix-2 Decimation-in-Time (DIT) FFT method that dramatically lowers computational complexity. Recognizing the need for high-performance FFT processors in signal processing applications, the authors designed a fully parallel and pipelined architecture for a 64-point FFT optimized for Application-Specific Integrated Circuit (ASIC) implementation. The proposed design reduced the number of multipliers and demonstrated hardware area reduction of 13.74% and a 16% decrease in power consumption compared to the conventional FFT architectures, offering a more efficient solution for real-time signal processing systems [9].

Gyuseong Kang, Woong Choi, and Jongsun Park proposed a low-cost FFT processor design based on embedded DRAM-based memory customization. The fact that FFT processors display regular and predictable memory access patterns that can be successfully utilized for eDRAM-based memory optimization is the basis for the main concept. Both pipelined and memory-based FFT architectures were subjected to the suggested customization methods.

The pipelined architecture reduces unnecessary RWL activity and word line driving power in column-interleaved memory arrays by using data-packing approaches and Read-Word Line (RWL) coupling write-assist techniques. The proposed eDRAM-based pipelined architecture demonstrates significant improvements in cost efficiency and power optimization for FFT processor design, as evidenced by implementation results for a 2048-point FFT using 0.11-μm CMOS technology [10]. Previous studies have focused on minimizing power and area while enhancing clock frequency, but have encountered significant challenges in achieving an optimal balance among these parameters. High-speed operation requires full pipelining. It increases the area and power. This paper proposes a semi-pipelined architecture to achieve an optimal balance among power, area, and speed. The implementation utilizes Dual-Port RAM (DP-RAM) and a Low-Power Carry-Select Adder (CSLA).

3. Semi Pipelined SDF FFT Design using Low Power Carry Select Adder

The implementation of FFT in an FPGA can be performed in two ways: fully parallel and serial implementation. The basic element of the FFT was implemented using the SDF architecture, as shown in Figure 1.

Both adders and subtractors are used to perform addition and multiplication operations. The architecture can perform one complex butterfly operation in a single clock cycle. Here, mux is used to select the direct or calculated input based on the selection line, which is used while performing the FFT operation. Figure 2 shows the butterfly element used for FFT computation.

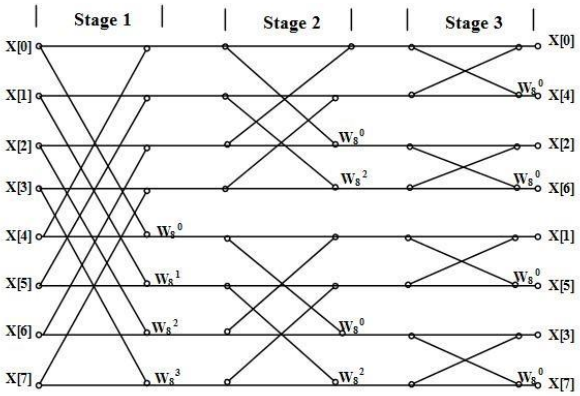

Fig. 1 Radix-2/4/8 DIF-FFT signal-flow graph of length 8

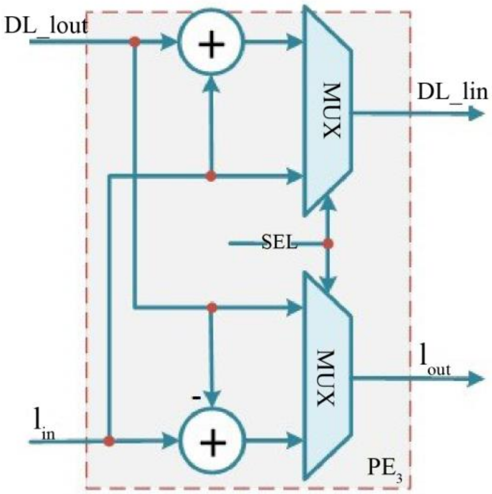

Fig. 2 Basic butterfly element used for FFT

3.1. Existing 8 Point FFT Architecture using Basic Processing Elements

The PE3 stage of the FFT implementation includes two multiplexers, adders, and a delay. This is the common element in each stage of the FFT computation. The data for the PE3 block were input as two blocks of data, and each block contained an equal amount of data. The data block that initially arrived will be stored in delay in order, and the next blocks of data that arrive will be added to or subtracted from the initially saved data block, respectively.

For example, in the first stage of computation of an 8-point FFT, the initial 4 numbers will be stored in delay, and the other 4 numbers will be introduced to the system. The two data blocks will undergo addition and subtraction as per the FFT algorithm. The twiddle factor will get multiplied by the output of the PE3 stage as per the need. The input and output of the PE3 stage may vary according to the stage of the FFT in which the PE3 is used. Also, the number of delays registered varies according to the FFT logic. Each PE3 stage is arranged to reap the maximum advantages of pipelined architecture.

3.2. Working Principle

Figure 3 shows the Butterfly element used for the 8-point FFT. The FFT architecture follows a pipelined data flow to minimize the delay. The 8-point FFT consists of three Processing Element (PE3) stages: PE3 Stage 1, PE3 Stage 2, and PE3 Stage 3, incorporating 4-bit, 2-bit, and 1-bit registers, respectively. Data are stored and retrieved from each register serially.

By the 4th Clock Cycle (CLK), the first four registers in PE3 Stage 1 are filled with input data. Starting from the 5th CLK Cycle, the system performs addition and subtraction operations using the direct input and data retrieved from the PE3 Stage 1 registers. The addition result is stored back in the same stage, whereas the subtraction result is forwarded to the register in the next stage. This process is repeated during the 6th CLK cycle. From the 7th CLK Cycle, the outputs from PE3 Stage 1 are combined (added and subtracted) with the data stored in the PE3 Stage 2 registers. The addition result is stored in PE3 Stage 2, and the subtraction result is passed to PE3 Stage 3 registers.

This process continues in subsequent clock cycles. Finally, X_3 and X_7 outputs are obtained at the last PE3 stage. In the 9th CLK cycle, the data from the first PE3 stage (the stored sum results) will move to the 2nd PE3 stage and be stored in Reg.; meanwhile, the data stored on the 2nd PE3 stage reg will pass to the 3rd PE3 Reg. and be stored. On the next CLK cycle, the same process is repeated. This results in outputs X1 and X5. In the next two CLK cycles, the data stored in the first PE3 stage Reg. and the second PE3 stage Reg. undergo addition and subtraction; the addition result is stored in PE3 stage 2 Registers, and the subtraction result passes to PE3 stage 3 Reg. and is stored. In the 12th CLK cycle, the stored data from the PE3 3rd stage and the result from the PE3 stage 2 will be added and subtracted to give the outputs X2 and X6. Finally, during the 13th clock cycle, the remaining data in the PE3 stage are passed forward and stored, and in the 14th clock cycle, addition and subtraction operations produce the final outputs X0 and X4.

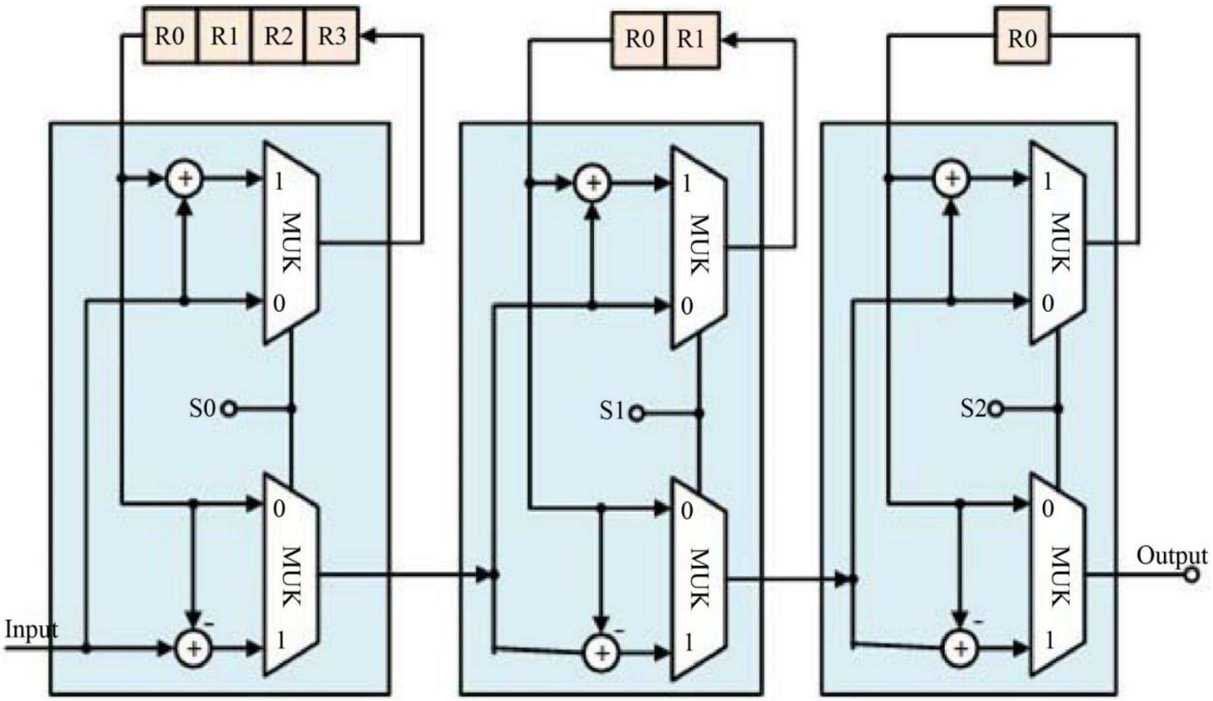

Fig. 3 Basic butterfly element used for FFT operation

3.3. Proposed FFT Architecture

In the proposed FFT architecture, a dual-ported RAM is introduced, enabling simultaneous read and write operations. To enable effective parallel access and processing, the input data was split in half and stored as two distinct portions. One sector at a time, data were extracted in line with the FFT execution technique.

The dual-port RAM is directly coupled to the main PE3 stage, allowing for simultaneous data input, retrieval, and output writing. The value of "N," or the number of FFT points, determines how many PE3 stages are used in the implementation. By enabling simultaneous read and write operations, the dual-port RAM integration greatly increases data throughput and boosts the FFT architecture's overall speed. The proposed FFT circuit architecture is shown in Figure 4. The dual-port RAM in the suggested system is where all of the input data is first kept. In the following step, Processing Element (PE3) blocks perform subsequent FFT computations. The architecture uses four parallel PE3 blocks to speed up an 8-point FFT. Data is retrieved from the dual-port RAM and passed into the first PE3 stage, where addition and subtraction operations are carried out between clock cycles. The 'n' value of the N-point fast Fourier transform (FFT) determines how many PE3 stages are needed. The input data was split into two halves and processed in parallel PE3 phases to increase computing speed further. Each FFT stage utilizes two concurrently operating PE3 blocks, enabling faster computation through pipelined executions. The integration of dual-port RAM not only facilitates simultaneous read and write operations but also eliminates unnecessary complexity during data transfer to the PE3 stage, thereby enhancing overall efficiency.

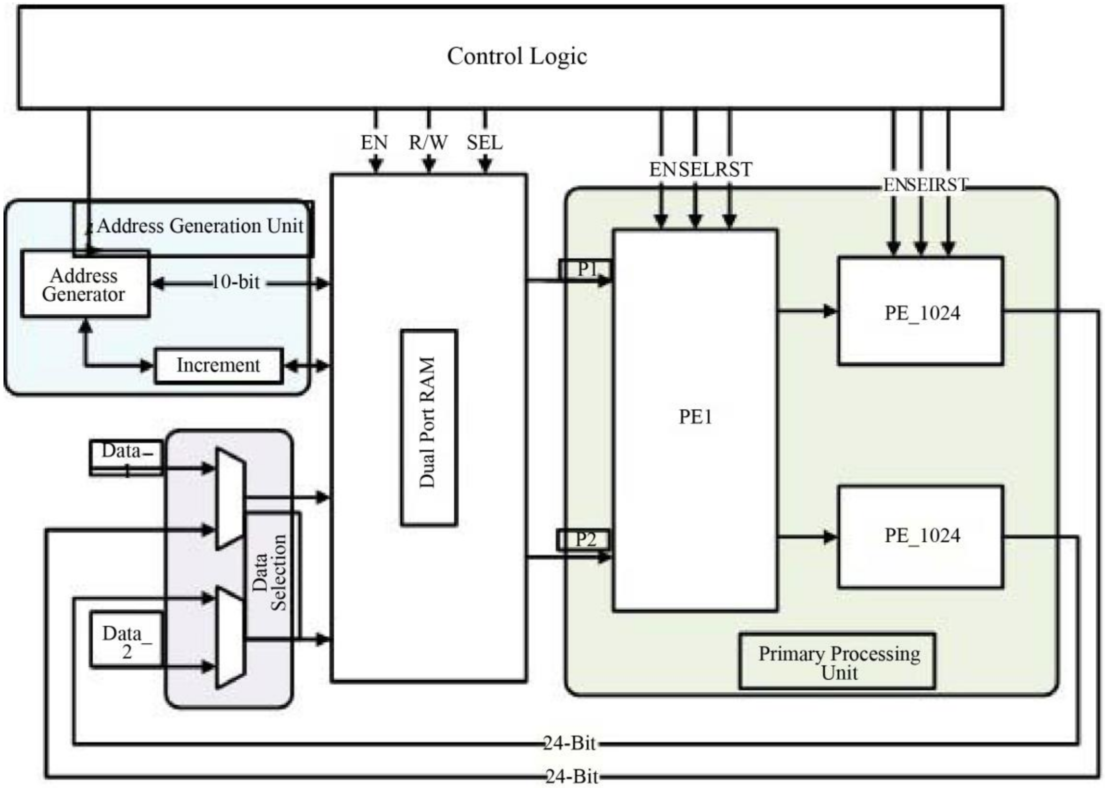

Fig. 4 Block diagram of proposed architecture for 2048-point FFT

3.4. Modified Low-Power, Low-Area Carry-Select Adder

The "N"-point size of the implemented FFT determines the number of adder blocks required, which are essential components of FFT architectures. An effective adder design is essential to improving the overall system performance because addition operations predominate in the FFT computation. A modified low-power Carry Select Adder (CSLA) is used in this work to maximize power consumption and area. By calculating a sum for both carry-in values (Cin = 0 and Cin = 1) in parallel and then utilizing multiplexers to pick the right result, the standard CSLA is frequently employed in high-speed arithmetic units to reduce the carry propagation time. However, because two Ripple Carry Adders (RCAs) are used for each operation, this method is not area-efficient. To solve this, the redesigned CSLA substitutes a Binary to Excess-1 Converter (BEC) for the RCA that corresponds to Cin = 1. Because the BEC uses fewer logic gates than an equivalent n-bit full adder layout, this swap drastically lowers the hardware size and power. Because the performance of the FFT depends on efficient addition operations, incorporating the modified CSLA improves speed and resource efficiency, which raises the overall efficacy and dependability of the FFT implementation. The redesigned Carry Select Adder’s structure is shown in Figure 5.

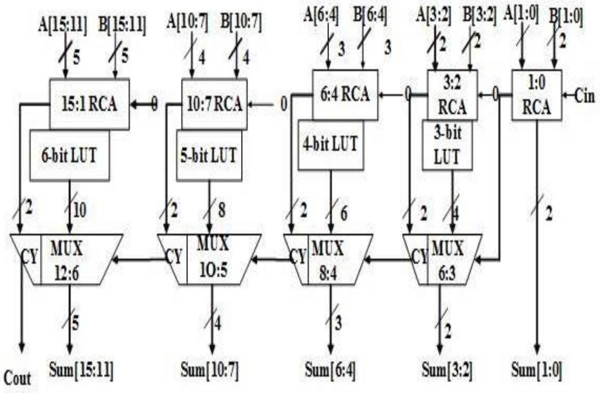

Fig. 5 Modified CSLA

4. Results and Discussions

The proposed FFT operation was implemented on an FPGA using Verilog HDL coding, and the design was synthesized using Xilinx ISE 14.2 and functionally verified with ModelSim DE 10.5a. The results obtained at various stages of the design and validation processes are systematically discussed. The proposed Processing Elements (PEs), Semi-Pipelined Radix-2 SDF (R2SDF) FFT Architecture, and Dual-Port Ram (DP-RAM) with an enhanced adder section (modified CSLA) were designed using Verilog HDL, simulated, and validated using ModelSim 10.5 (Mentor Graphics). Furthermore, the design was implemented targeting the Spartan-3 FPGA. Table 1 shows the FPGA specifications used for the proposed FFT synthesis. The primary objectives of the design are to achieve low power consumption, reduced slice and LUT utilization, and high speed and throughput, addressing the key performance metrics in the VLSI system design. The primary objective of this paper is to reduce the delay and area consumption in the frequency transformation process used in OFDM systems. To efficiently convert time-domain signals into frequency-domain representations, a pipelined Radix-2 Single-Path Delay Feedback (R2SDF) architecture was developed in this paper.

Table 1. FPGA specifications

FFT size	2048
Speed Grade	-2
Target Device	Xc6vcx75t-2ff484
Data Width	12
RAM’s	2
Registers	2048
Multiplexers	80
Flipflops/Latches	1812
Complex multipliers	10
Shift Registers	1584

4.1. Synthesis of SP-FFT-SPDT-DPRAM in FPGA

To increase the processing speed of the FFT module, the existing PE structures were modified by adding a dual-port RAM unit at the beginning. The dual-port RAM unit of the PE structures facilitates fast access to data in the main modules. The effects of the dual-port RAM unit and the enhanced adder section among all outputs result in a high-speed operation in frequency transformation computation. The principal objective of this paper was to design an FFT architecture that achieves low power consumption and reduced area utilization. Key performance parameters, (i) Number Of Slice Look-Up Tables (LUTs), (ii) Number Of Occupied Slices, (iii) Number Of Slice Registers, (iv) Delay (ns), (v) Operating Frequency (MHz), and (vi) Power Consumption (W), were carefully evaluated to compare the proposed FFT design with existing implementations.

The comparison shows that the proposed FFT has an advantage in terms of area reduction and resource management. Comparing the values obtained for the parameters considered in the paper, obtained a clear view of the improvement in performance. Comparing Tables 2 and 3, there is a visible improvement in reducing hardware resource usage in the proposed FFT architecture.

Table 2. FPGA performance for convetional FFT

Size	**8**	**16**	**32**	**64**	**128**	**256**	**512**	**1024**	**2048**
Number of Slice LUT’s	446	667	777	1003	1263	1531	1830	2520	3419
Number of Occupied Slices	145	233	217	273	325	400	504	676	940
Number of Slice Registers	152	348	492	644	804	980	1140	1460	1812
Delay’s (in ns)	6.91	12.77	3.85	4.71	5.48	6.44	7.45	8.17	9.18
Freq (MHz)	67.82	48.55	87.05	80.95	75.65	71.01	71.01	63.24	63.24
Power(W)	1.29	1.29	1.29	1.29	1.29	1.29	1.29	1.29	1.29

Table 3. FPGA performance for SP-FFT-SPDT-DPRAM

Size	**8**	**16**	**32**	**64**	**128**	**256**	**512**	**1024**	**2048**
Number of Slice LUT’s	174	342	456	583	737	911	1307	2008	3122
Number of Occupied Slices	111	111	156	198	242	320	490	605	1045
Number of Slice Registers	174	246	332	434	568	742	1092	1650	2696

Delay’s (in ns)	13.59	4.479	5.345	6.21	7.075	8.083	8.805	9.813	9.563
Freq (MHz)	74.11	227.61	190.17	163.305	143.091	114.697	114.697	114.697	114.697
Power(W)	1.293	1.293	1.293	1.293	1.293	1.293	1.293	1.293	1.293

Table 4. Comparison with other FFT designs

Design	Proposed	[11]	[5]	[7]
FFT Size, K	8 ~2048	4 ~64	8 ~32	128 ~2048
Word length bit	12	16	Oct-13	16
SQNR, dB	-	51	53.28	-
Technology	180nm	65nm	0.13nm	0.18 μm
Voltage, V	1.2	1.2	1.2	1.6
Maximum frequency, MHz	227.61	125	400	80
Throughout	1	1	1	-
Execution time, μs	13.59	102.4	205.83	-
Chip area, mm2	1.852	1.05	2.7	2.0164
Normalised area,mm2	-	4.2	2.7	-
Power, mW	1.293 @ 20 MHz	33.5 @ 40 MHz 4 K	35.7 @ 40 MHz 8 K	9.79 @20 MHz
Normalised power, mW	1.293	69.79	47.91	2.4

4.1.1. Hardware Resource Utilization

SP-FFT-SPDT-DPRAM uses fewer Slice LUTs, occupied Slices, and slice registers than the existing FFT implementation method.

This was reflected in the decreased chip area. Table 2 and 3 collectively show the hardware resource utilization of the proposed FFT.

Figure 6 shows the synthesis performance results of the conventional FFT. Figure 7 shows the synthesis performance of the proposed FFT architecture. Comparisons of the proposed chip with other FFT processors [5, 7, 11], including FFT size, supply voltage, power consumption, execution time, chip area, and maximum frequency, are listed in Table 4.

4.1.2. Frequency Analysis

The proposed FFT processor can operate up to 227.61 MHz and dissipates 1.293 mW at 20 MHz with an execution time of 13.59 µs. Table 4 compares the frequency data with other FFT methodologies.

The results show the improved performance of the proposed architecture. Figure 8 shows the comparison of the proposed FFT with the conventional FFT.

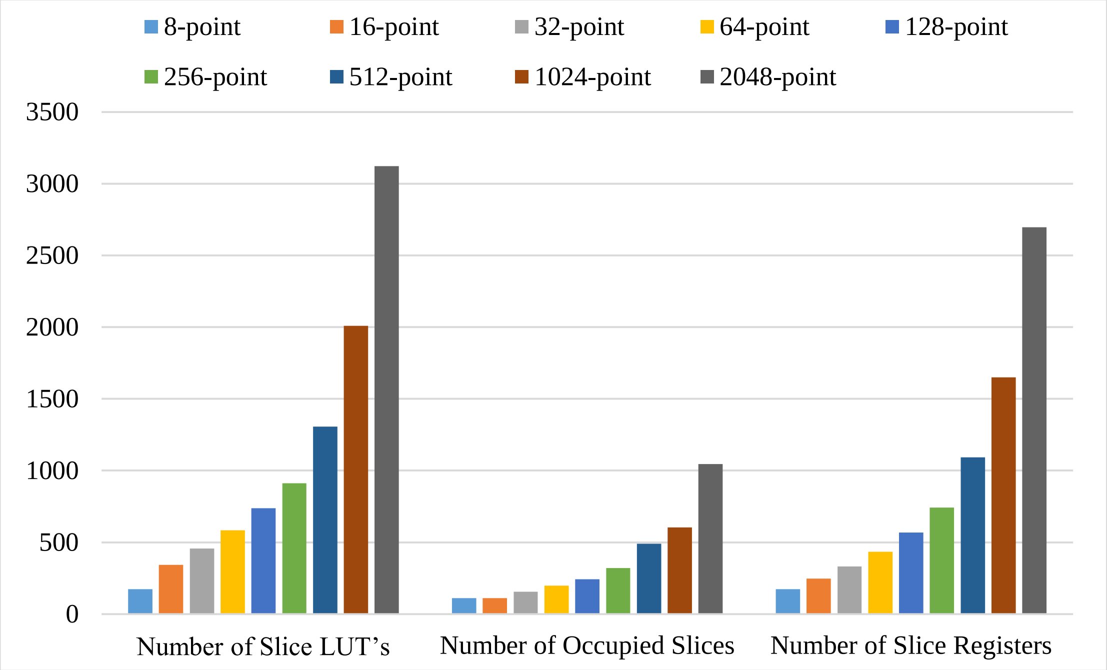

Fig. 6 FPGA performance chart for conventional FFT implementation

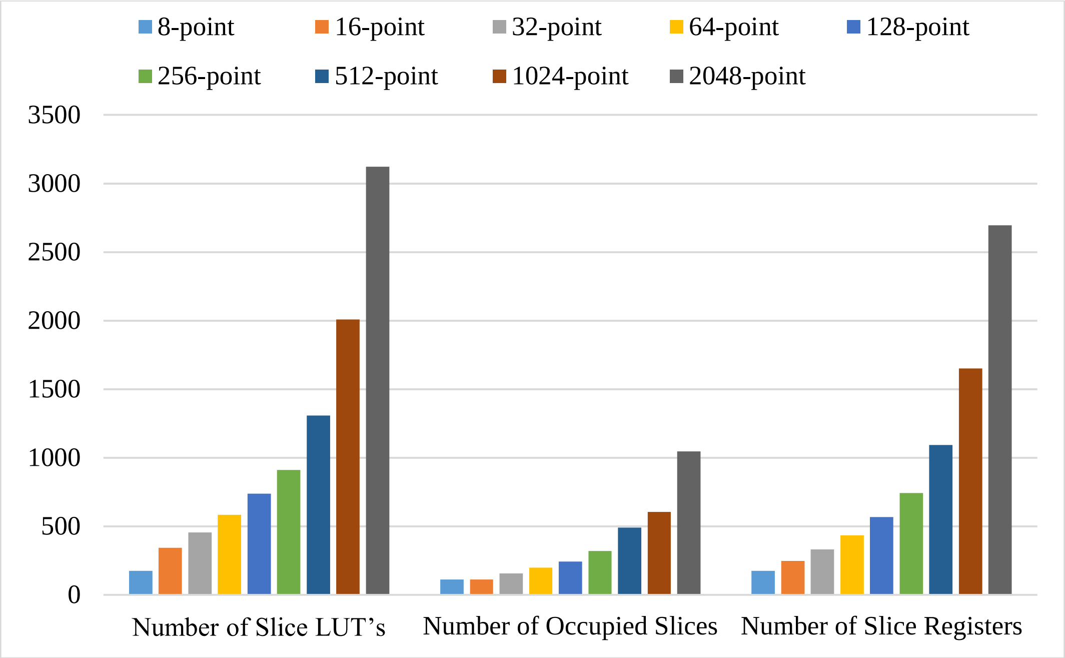

Fig. 7 FPGA performance chart for SP-FFT-SPDT-DPRAM implementation

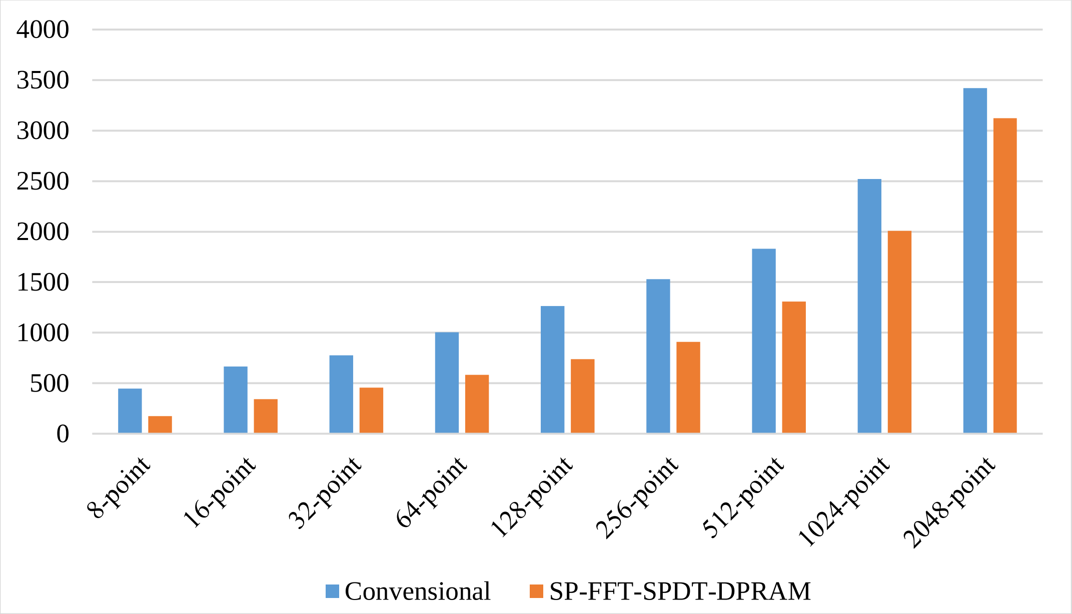

Fig. 8 FPGA performance chart for SP-FFT-SPDT-DPRAM implementation

4.2. Implementation of SP-FFT-SPDT-DPRAM in ASIC (180 nm CMOS Technology)

The proposed FFT architecture was implemented using 180 nm technology. Table 4 summarizes the key hardware implementation parameters and provides a comparison with existing methodologies. The entire FFT processor was designed using Verilog HDL, with dual-port RAM employed for intermediate data storage. Although the compared FFT processors differ in both CMOS technology nodes and FFT sizes, the proposed design demonstrates a competitive performance. It supports a maximum operating frequency of 227.61 MHz and achieves a throughput of 1 MSample/s. The chip area occupied 1.852 mm², and the execution time for a 2048-point FFT was 13.59 µs. The post-layout simulation results indicated a core power consumption of only 1.293 mW. Figure 9 shows the proposed FFT ASIC implementation comparison with the conventional FFT.

4.2.1. Power Analysis

Power consumption analysis is crucial for hardware implementation. A comparison between the proposed FFT architecture and the conventional FFT method clearly demonstrates the power efficiency of the proposed design. According to the Xilinx analysis report, the proposed method exhibited a notable reduction in power consumption, highlighting its advantage over existing approaches in terms of energy efficiency. Figure 10 shows the RTL schematic of the proposed FFT.

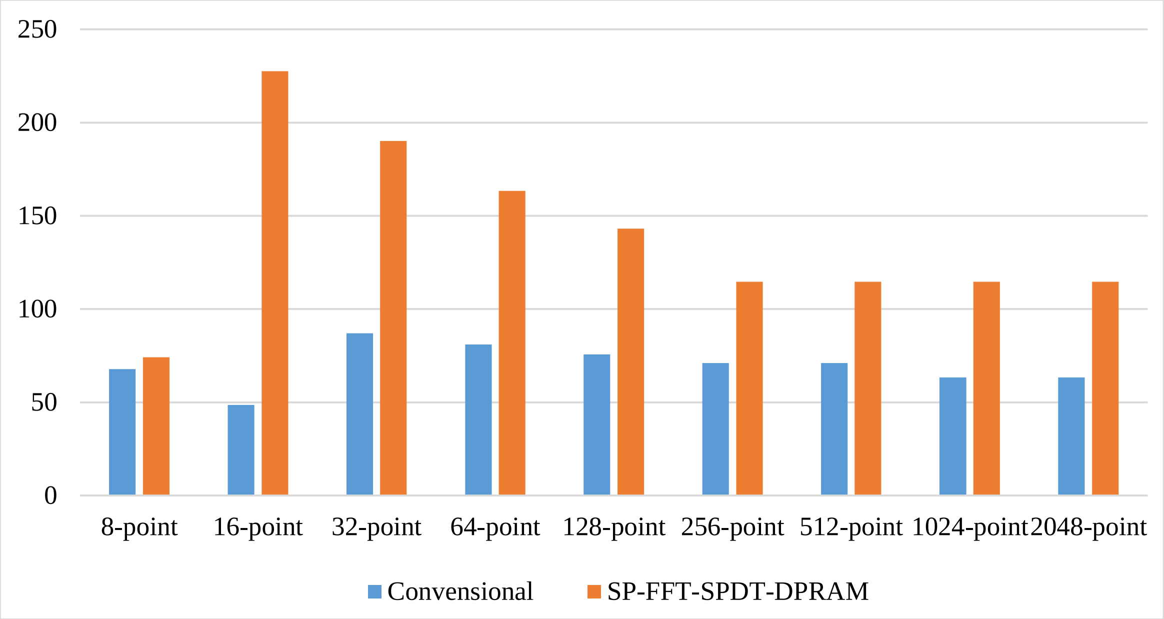

Fig. 9 Performance of frequency chart for SP-FFT-SPDT-DPRAM

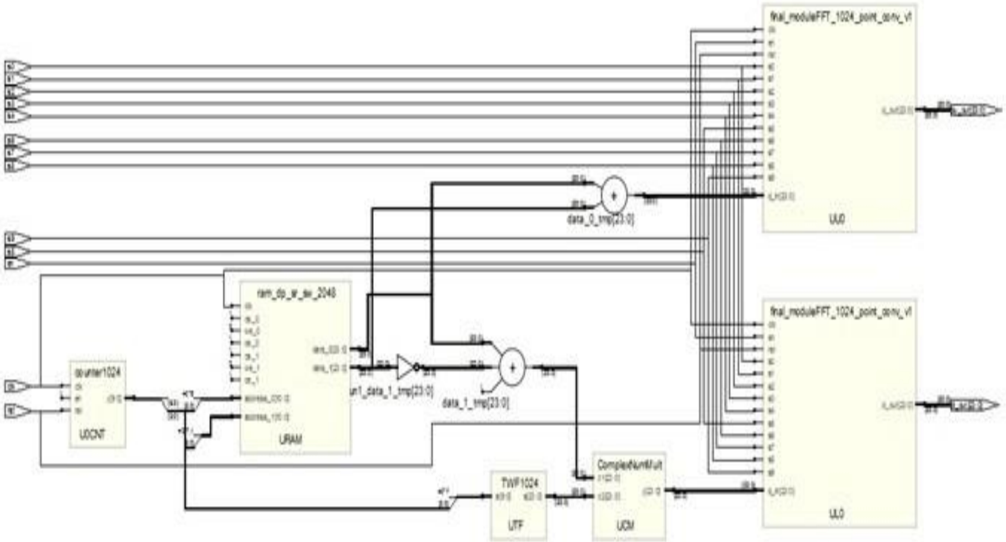

Fig. 10 Proposed FFT architecture synplify RTL schematic for 2048-point FFT

5. Conclusion

In this paper, a high-performance 128–2048-point pipelined FFT architecture suitable for WiMAX and LTE applications is proposed. The FFT operation is executed using (Processing Element) PE3 stages based on the FFT algorithm. To enhance the performance, a dual-port RAM was integrated into the design, enabling an efficient and organized data flow. This addition reduces the number of pipeline stages and clock cycles, resulting in significant improvements in the area efficiency and processing speed. The architecture also employs pipelining techniques to cascade the butterfly elements, which effectively reduces the silicon area required for high-radix butterfly computations. Furthermore, the adder section is implemented using an enhanced Carry-Select Adder (CSLA), contributing to improved overall performance in terms of resource utilization and speed. The key advantages of the proposed architecture include high-speed computation and area efficiency, making it well-suited for use in communication systems such as LTE devices, wireless modems, and portable phones.

References

[1] Chu Yu et al., "A Low-Power 64-Point Pipeline FFT/IFFT Processor for OFDM Applications," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 1, pp. 40-40, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[2] M. Vinay Kumar, A. David Selvakumar, and P.M. Sobha, "Area and Frequency Optimized 1024 Point Radix-2 FFT Processor on FPGA," *2015 International Conference on VLSI Systems, Architecture, Technology and Applications (VLSI-SATA)*, Bengaluru, India, pp. 1-6, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[3] Nooshin Mahdavi, R. Teymourzadeh, and Masuri Bin Othman, "VLSI Implementation of High Speed and High Resolution FFT Algorithm based on Radix 2 for DSP Application," *2007 5th Student Conference on Research and Development*, Selangor, Malaysia, pp. 1-4, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[4] Arathi Ajay, and R. Mary Lourde, "VLSI Implementation of an Improved Multiplier for FFT Computation in Biomedical Applications," *2015 IEEE Computer Society Annual Symposium on VLSI*, Montpellier, France, pp. 68-73, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[5] Hao Xiao et al., "VLSI Design of Low-Cost, High-Precision Fixed-Point Reconfigurable FFT Processors," *IET Computers and Digital Techniques*, vol. 12, no. 3, pp. 105-110, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[6] Shaohan Liu, and Dake Liu, "A High-Flexible Low-Latency Memory-Based FFT Processor for 4G, WLAN, and Future 5G," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 3, pp. 511-523, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[7] Chu Yu, and Mao-Hsu Yen, "Area-Efficient 128- to 2048/1536-Point Pipeline FFT Processor for LTE and Mobile WiMAX Systems," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 9, pp. 1793-1800, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[8] Chu Yu, "A 128/512/1024/2048-Point Pipeline FFT/IFFT Architecture for Mobile WiMAX," *2013 IEEE 2nd Global Conference on Consumer Electronics (GCCE)*, Tokyo, Japan, pp. 243-244, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[9] Harsha Keerthan Samudrala et al., "Parallel and Pipelined VLSI Implementation of the New Radix-2 DIT FFT Algorithm," *2018 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS)*, Hyderabad, India, pp. 21-26, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[10] Gyuseong Kang, Woong Choi, and Jongsun Park, "Embedded DRAM-Based Memory Customization for Low-Cost FFT Processor Design," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 12, pp. 3484-3494, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[11] S. Kala et al., "Energy Efficient, Scalable, and Dynamically Reconfigurable FFT Architecture for OFDM Systems," *2014 Fifth International Symposium on Electronic System Design (ISED)*, Surathkal, Mangalore, India, pp. 20-24, 2014. [CrossRef] [Google Scholar] [Publisher Link]