International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online) Volume 13, Issue 12, December 2024)

VLSI Architecture of DIF Fast Fourier Transform Filterbank for FPGA DSP Application

Siddhant Vishwakarma[1], Dr Vijay Yadav[2]
[1]M.Tech Scholar, Department of Electronics & Communication Engineering, LNCT, Bhopal, India
[2]Associate Professor, Department of Electronics & Communication Engineering, LNCT, Bhopal, India

***Abstract*— The Decimation-In-Frequency (DIF) Fast Fourier Transform (FFT) algorithm has become a cornerstone of digital signal processing (DSP) due to its computational efficiency and ability to handle real-time data processing tasks. In the context of Very Large-Scale Integration (VLSI) and Field-Programmable Gate Arrays (FPGAs), DIF FFT filterbanks provide a robust architecture for implementing high-performance DSP systems. This paper presents the VLSI architecture of DIF FFT filterbanks tailored for FPGA applications, emphasizing their significance in achieving low latency, high throughput, and efficient resource utilization. The simulation is performed using Xilinx ISE 14.7 software.**

***Keywords*— FPGA, DIF-FFT, DSP, VLSI, Xilinx.**

I. INTRODUCTION

The Digital Signal Processing (DSP) is an essential domain in modern technology, driving applications in telecommunications, image and audio processing, biomedical instrumentation, and multimedia systems. Among the critical operations in DSP, the Fast Fourier Transform (FFT) stands out as a fundamental algorithm for spectral analysis, filtering, and frequency-domain signal manipulation. Within FFT algorithms, the Decimation-In-Frequency (DIF) approach is particularly advantageous for hardware implementations due to its computational simplicity and reduced number of arithmetic operations. Its efficiency makes it a preferred choice for designing FFT filterbanks, which are integral components in many DSP systems.

The integration of FFT filterbanks into hardware platforms has become increasingly important with the rise of real-time DSP applications. Field-Programmable Gate Arrays (FPGAs) have emerged as a leading platform for such implementations due to their reconfigurability, parallel processing capabilities, and ability to achieve low-latency, high-speed computations. At the heart of these FPGA-based DSP systems lies the VLSI architecture, which enables the translation of complex mathematical algorithms like the DIF FFT into efficient, high-performance hardware designs. The synergy between VLSI architectures and FPGAs has enabled the development of sophisticated DSP solutions that meet the stringent demands of modern applications.

The VLSI architecture of DIF FFT filterbanks specifically tailored for FPGA platforms brings several advantages, including efficient resource utilization, high data throughput, and scalability. By leveraging the parallelism inherent in FPGAs, DIF FFT filterbanks can perform simultaneous computations, significantly enhancing processing speed. Additionally, the modular nature of VLSI design allows for pipelining and hierarchical processing, which are essential for achieving low-latency performance in real-time DSP systems. However, designing such architectures comes with challenges, such as managing power consumption, minimizing hardware resource usage, and ensuring numerical accuracy in fixed-point arithmetic.

This paper presents a comprehensive analysis of the VLSI architecture of DIF FFT filterbanks for FPGA-based DSP applications. It begins with an overview of the fundamentals of the DIF FFT algorithm and its implementation as a

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online) Volume 13, Issue 12, December 2024)

filterbank. The discussion then shifts to the architectural considerations unique to VLSI design, such as the use of pipelining, memory organization, and optimization techniques to maximize performance and minimize hardware costs. Key challenges in implementing these architectures are addressed, including resource constraints, trade-offs between speed and area, and the integration of FFT filterbanks into broader DSP systems.

Moreover, this work highlights the role of FPGA-specific features, such as hardware accelerators and configurable logic blocks, in enhancing the efficiency of DIF FFT filterbanks. The impact of design techniques such as parallelism and resource sharing on power consumption and computational throughput is also explored. To provide a comprehensive perspective, the paper discusses various applications of DIF FFT filterbanks in fields like telecommunications (e.g., OFDM systems), multimedia (e.g., audio equalization), and biomedical signal processing (e.g., EEG and ECG analysis).

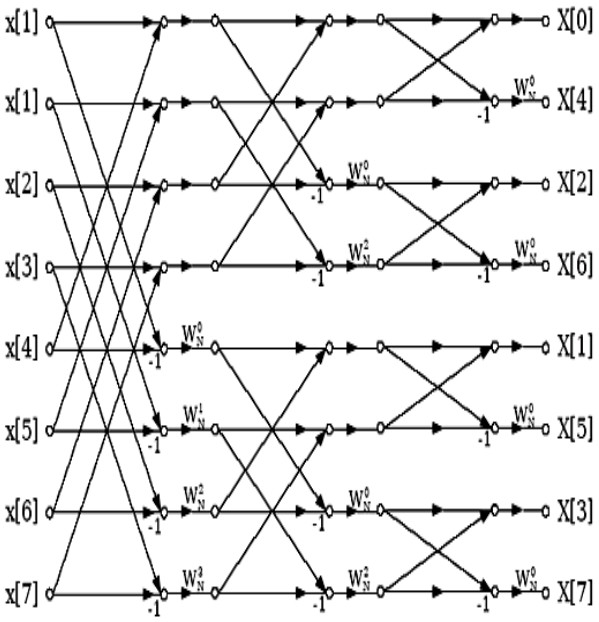

Figure 1: Radix-2 Decimation-in-Frequency FFT

II. PROPOSED METHODOLOGY

The proposed work is to implementation of the Decimation in frequency Fast Fourier Transform (DIF-FFT) algorithm using Xilinx software.

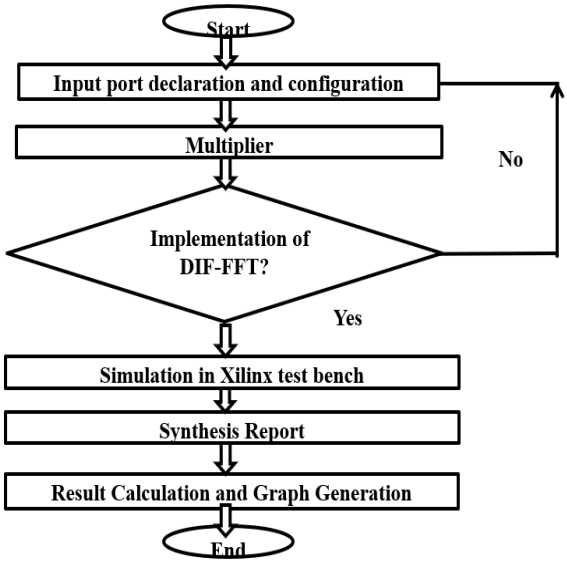

Figure 2: Flow Chart

The following steps is involved to complete the research work-

- The first step is to assign the input and output port in the Xilinx editor.
- Assign the supporting files, function and declaration of ports etc.
- Implementation of the multiplier to support the operation of DIF-FFT algorithm.
- Now run this algorithm and perform the implementation section or the view of register transfer level (RTL) view.
- Now check the result validation in the xilinx test bench using Isim simulator.
- Generate the synthesis report and check the results.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online) Volume 13, Issue 12, December 2024)

It has been shown that by re-indexing a subset of the output samples resulting from the conventional decompositions in the radix-4 FFT algorithms, a substantial reduction in the number of twiddle factor evaluations or accesses to the lookup table can easily be obtained. These savings have been achieved without imposing any additional computational or structural complexity in the algorithms. The basic idea for improved FFT algorithms introduced in this paper can also be applied to other higher radices DIT and DIF FFT algorithms as well as to multidimensional FFT algorithms.

When the number of data points N in the DFT is a power of 4 (i.e., $N = 4^v$), we can, of course, always use a radix-2 algorithm for the computation. However, for this case, it is more efficient computationally to employ a radix-r FFT algorithm. Let us begin by describing a radix-4 decimation-in-time FFT algorithm briefly. We split or decimate the N-point input sequence into four subsequence's,

$$x(4n), x(4n+1), x(4n+2), x(4n+3), n = 0, 1, \cdots, N/4-1$$

$$X(p,q) = \sum_{l=0}^{3} [W_N^{lq} F(l,q)] W_4^{lp}$$

$$F(l,q) = \sum_{m=0}^{(N/4)-1} x(l,m) W_{N/4}^{mq}$$

p=0, 1, 2, 3; l = 0, 1, 2, 3; q = 0,1,2, . . . , -1

and

$$x(l,m) = x(4m+1)$$

$$X(p,q) = X(\frac{N}{4}p + q)$$

Thus the four N/4-point DFTs $F(l,q)$ obtained from the above equation are combined to yield the N-point DFT. The expression for combining the N/4-point DFTs defines a radix-4 decimation-in-time butterfly, which can be expressed in matrix form as

$$\begin{bmatrix} X(0,q) \\ X(1,q) \\ X(2,q) \\ X(3,q) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} W_N^0 F(0,q) \\ W_N^4 F(1,q) \\ W_N^{24} F(2,q) \\ W_N^{34} F(3,q) \end{bmatrix}$$

Each butterfly involves three complex multiplications, since W_N^0=1 and 12 complex additions.

III. SIMULATION RESULTS

The simulation work is performed using the Xilinx ISE 14.7 software.

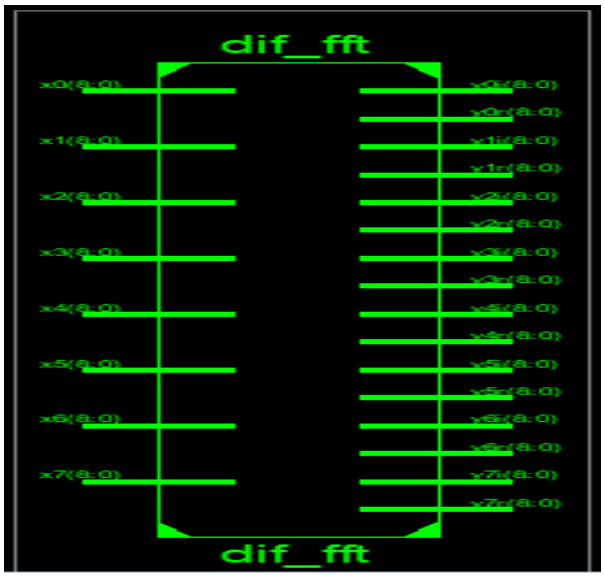

Figure 3: Top level View

Figure 3 is showing the top view of the proposed DIF-FFT algorithm, which includes the 8 inputs pins and 16 output pins.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online) Volume 13, Issue 12, December 2024)

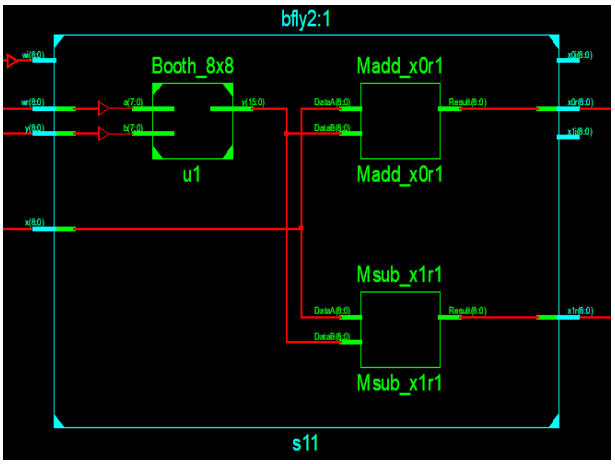

Figure 4: RTL view of the booth multiplier component

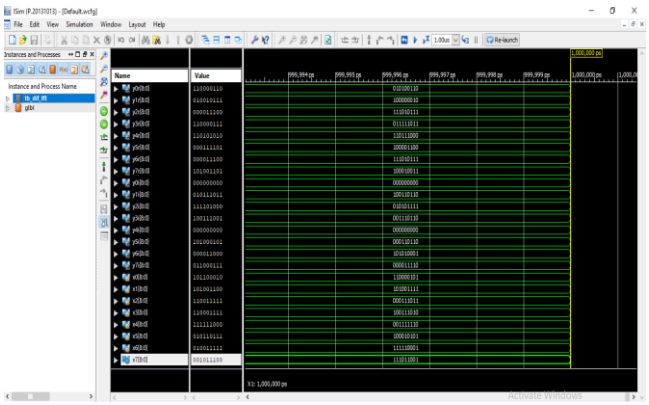

Figure 5: Result validation in Test bench

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	556	204000	0%
Number of fully used LUT-FF pairs	0	556	0%
Number of bonded IOBs	210	600	35%

Figure 6: Device utilization summary

Figure 6 is showing summary of components using duration proposed approximate multiplier implementation. Total number of slice look up table used 556 while availability is 204000. Look up table and flip flop pairs used 0 while availability is 556. Bonded input output block used 210 while availability is 600. Now total area is calculated from this utilization summary. Therefore, 11.66 % of area used for implementation of proposed DIF-FFT.

Table 1: Result Comparison

Sr No.	Parameter	Previous Work	Proposed Work
1	Method	Multi-stage multirate filterbank and FFT	DIF-FFT Filter bank
2	Total Component count	1572	766
3	Delay	4.35 ns	1.718 ns
4	Power	210 micro watt	128 micro watt
5	Frequency	436.94 MHz	582 MHz

IV. CONCLUSION

This paper presents VLSI implementation of the decimation-in-frequency- Fast Fourier Transform using the booth multiplier algorithm. Simulation is performed using the Xilinx ISE 14.7 software. Simulation results show that total component of the proposed work is 766 while the previous algorithm uses 1572 components. The delay is optimized by the proposed work is 1.718 ns while previous it is 4.35 ns. The power is 128 microwatt in the current design while previous it is 210 microwatt. The frequency or speed of the algorithm is 582 MHz while previous it is 436.94 MHz. Therefore proposed DIF-FFT algorithm gives better result in term of calculated parameters. So it can be used in high speed, low area and latency.

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 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.