Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 5, May : 2023

IMPLEMENTATION OF RADIX-2^2 FFT MDC ARCHITECTURE USING VEDIC MULTIPLIER FOR HIGH-SPEED APPLICATIONS

Mrs R.**Sravanthi,** PG student, dept. of ECE, Lendi Institute of Engineering and Technology: Vizianagaram, India
Dr.B.Sridhar, Professor, dept. of ECE, Lendi Institute of Engineering and Technology: Vizianagaram,India
Mr.B.Ramamohan, Professor, dept. of ECE, Lendi Institute of Engineering and Technology: Vizianagaram,India

ABSTRACT
Fast Fourier Transform (FFT) is an algorithm to calculate the DFT of a sequence which got many variations like radix-2, radix-4 and radix-2^k algorithms. The radix-2^k algorithm has a simple control structure, adaptive and requires less hardware making it suitable for low power design when compared to other algorithms. The proposed radix-2^k architecture utilizes multiple delay commutator (MDC) for attaining simple butterfly unit with less memory requirements. The input scheduling algorithm is used for shifting the data along the delay line which results in low power requirement. In this paper, we propose radix-2^2 FFT architecture with Vedic multiplier for high speed applications. The proposed architecture of 8-point, 16-point, 512 and 1024-point is implemented on Xilinx Vertex-5 FPGA board. The implementation of Radix 2^2 1024-point FFT enhances the speed of operation with reduction in area by using Vedic multiplier structure in the butterfly unit with delay of 42.102ns, 121 registers, 11629 LUTs and 5515 number of slices.

Keywords: FFT, MDC, DFT, Vedic Multiplier, Baugh Wooley Multiplier, LUT, operating frequency.

1. INTRODUCTION
FFT is proposed as an algorithm to calculate Discrete Fourier Transform which depends upon divide-conquer and recursion methods. In divide and conquer approach a larger problem is decomposed into smaller ones. In modern digital systems, the FFT processor plays an important role. For Ultra Wide Band (UWB) applications, high throughput can be achieved through complex multipath structures where data is processed parallely with a little penalty in area and memory as compared to single-path structures. The main research interest in FFT architecture implementation is to perform multiplication operation with less memory requirement, lower power utilization and high accuracy as suitable for today's applications. Many research works are reported in the open literature on the development of high-performance FFT architectures.
A historical overview of FFT algorithms is discussed [1] with a review of the state of art mechanism for different algorithms, statement of open problems and contribution of Gauss approach to computing DFT. In [2] radix-2 FFT architecture is developed which iterate over an array of Fourier series and delivers result in 2Nlog2N operations by utilizing the same data storage as the input array. In [3] prime factor algorithm for performing non two PowerPoint FFT using Winograd Transform and mixed-radix is developed. The FFT processor uses 3780 points and memory of 98280 bits for performing operations. The developed processors work for variable-size FFT/IFFT implemented in Chinese Television Broadcast (CTB). Novel Brunns FFT for software-defined radio [4] is implemented with successive stages to increment the number of bits. The developed architecture utilizes less hardware keeping changes in Noise to Signal Ratio (NSR) negligible when compared to conventional Brunns. The architecture is developed up to 14-bit precision where 14-bit FPGA version occupies 172 LUT and exhibits a delay of 11.3ns. Radix 22, 1024-point pipe-line processor in [5] is developed with 4 multipliers and 1024-word data memory for 1K FFT processor operating at 30 MHz frequency with

3.3 V power supply, require 40mm2 area. Baugh-Wooley multiplier [6] for signed multiplication is implemented with Verilog HDL utilizing 33 LUTs and occupies 18 slices.
Parallel pipeline FFT architectures [7] for radix 2 and 4 is developed. Radix-4 4096-point FFT is performed at 100 MHz signals using 16 computational elements per stage within a time of 0.5 microseconds. Classification of processors in [8] presents radix-2 and radix-4 architectures for single path and multipath delay architectures. The memory-based architecture shows that radix-2 requires 1 complex multiplier where radix 4 requires 3 multipliers and operates at N/2log2N+2 and N/4log4N cycles respectively. Parallel FFT is achieved through Folding Transformation in [9] which enables register minimization. Complex FFT is performed utilizing hardware in the serial architecture to derive L parallel architectures without increasing the complexity by L factor. The 64 point size operations utilize 2320 registers, 1916 LUT, 780 slices of FPGA. A pipelined FFT architecture in [10] is proposed for word sequential data. The proposed radix-4 FFT architecture uses 1og4 N-1 multipliers, 3log4N adders and occupies memory of 2N blocks. The radix-2K feedforward FFT architecture in [11] can be used for any power of two by applying both decimations in frequency and time. The feedforward architecture utilizes less hardware than parallel feedback ones. A 4096-point FFT with 16 samples parallel processor operating at 193 MHz utilizing 6423 slices with the latency of 1.516 micro seconds. FFT with radix-4 delay commutator [12] with 10 MHz clock rate and 40 MHz data processing rate is proposed. The commutator designed contains 108000 transistors, 12288 shift registers and 200 gates. Utilizing proposed commutator chip, the 40 MHz 4096-point FFT is reduced from 1375 commercial integrated circuits to 546 circuits which account to 60% reduction in complexity. For larger transforms like 16384 point FFT, the complexity is reduced from 1634 integrated circuits to 670 circuits. Parallel array multiplication with 2's complement in [13] performs m-bit by n-bit parallel multiplication. In algorithm 2's complement multiplication is considered as an equivalent parallel array addition where each partial product bit is the AND of a multiplier bit and a multiplicand bit where the signs of all the partial product bits are positive. Quantization errors faced by FFT are reported in [14]. Quantization errors occur due to finite word lengths in the digital system. Coefficient rounding and floating-point arithmetic quantization are reported where exact and truncated values for the coefficients 1 and -j are found to be roughly equivalent. The algorithms proposed are used to model quantization errors in convolution filters. Impact of Fixed-point accuracy performance on FFT is analysed in [15] by measuring signal to quantization noise ratio of 2N radix-2 FFT implementation thereby determining the trade-off between required hardware and output signal integrity. A 1024 point FFT with Vedic multiplier in butterfly architecture with optimum performance in terms of area and delay profile is reported in this paper.

2. METHODOLOGY

This work contains procedure to develop radix-2^2 FFT algorithm in combination with vedic multiplier to enhance the operating frequency and delay performance of FFT architecture.
The FFT algorithm was developed by Cooley and Tuckey in 1965. It could reduce the complexity of DFT significantly from $O(N^2)$ to $O(N \log_2 N)$. A basic N-point DFT can be represented as

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} \quad , K = 0, 1, 2 \ldots \ldots N-1. \qquad (1)$$

where $W_N^{kn} = e^{\frac{-j2\pi nk}{N}}$ is called as twiddle factor.
The radix-4 algorithm decomposes X(K) into 4 or more sub processes, which reduces the number of adders and multipliers. The implementation of radix-22 algorithm was chosen over the remaining algorithms as it uses the simple control structure of radix-2 and hardware saving technique of radix-4. In turn it also consumes less power which is inherent to its architecture. The radix-2^2 equation is represented as

$$X(k1 + 2k2 + 4k3) = \sum_{n3=0}^{\frac{N}{4}-1} [H(k1, k2, k3)] \, . \, W_{N/4}^{n3k3} \qquad (2)$$

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 5, May : 2023

where $H(k1,k2,k3) = \left[A + (-j)^{(k1+2k2)}B\right].W_N^{n3(k1+2k2)}$ (3)

$$A = x(n3) + (-1)^{k1}x\left[n3 + \frac{N}{2}\right] \quad (4)$$

$$B = x\left[n3 + \frac{n}{4}\right] + (-1)^{k1}x\left[n3\frac{3N}{4}\right] \quad (5)$$

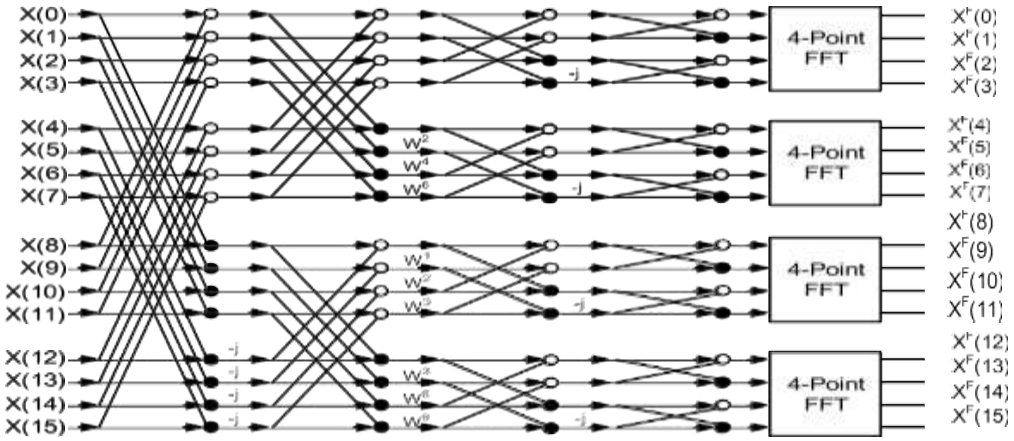


Figure.1: Radix-2^2 16 point FFT algorithm.

The A and B terms used in equation (3)

are the two conventional radix-2 FFT terms, and $W_N^{n3(k1+2k2)}$ is the twiddle factor in the complex multiplication. The radix-2^2 signal flow graph shown in Figure1 contains two cascaded radix-2 FFT architecture. Radix 2^2 decomposes two stages of radix-2 into cascaded radix-2 FFT shown in Figure 1 forming single stage thereby having same iterated stages of $\log_4 N-1$ as in radix-4. To achieve high throughput there is a need for separating inputs to each butterfly unit by optimum distance which can be attained by using delay feedback and commutator.

The delay feedback in Figure 2 shows that the butterfly unit outputs are delayed at pre-fixed distance to get pair with next stage inputs and finally fed back to butterfly unit for further operation cycles. Single delay feedback stores the data in shift registers, later to increase throughput multiple delay feedbacks are introduced with more delay paths which comes with hardware complexity.

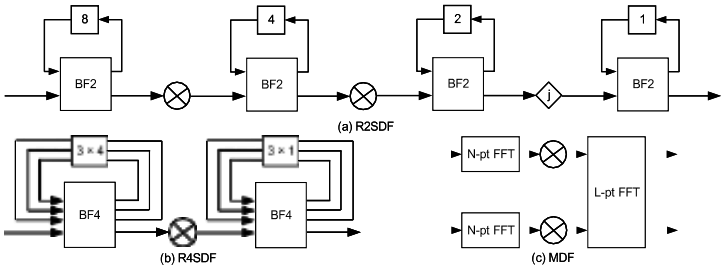


Figure 2. Delay feedback architecture: (a) Radix-2 single delay, (b) Radix-4 single delay, (c) Multiple delay.

The delay commutator make the output available immediately instead of feeding back butterfly units using various delay paths at correct placements. Figure 3 shows radix-4 multiple delay commutator which contains lower multiplier utilization rate but utilizes more memory.

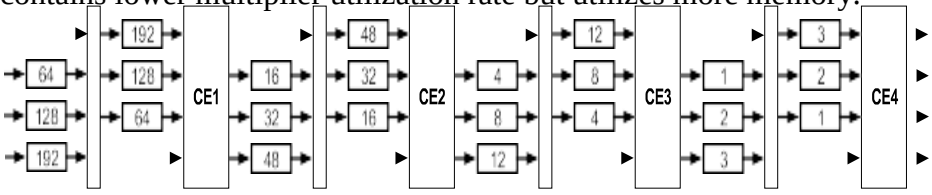


Figure. 3. Multiple delay commutator of radix 4

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 5, May : 2023

The developed 1024-point radix-2^2 processor shown in Figure 4 comes with two parallel data paths and contains nine processing levels. This structure drops shift registers present in conventional radix-2^2 algorithm which saves circuit area and reduces consumption of energy.

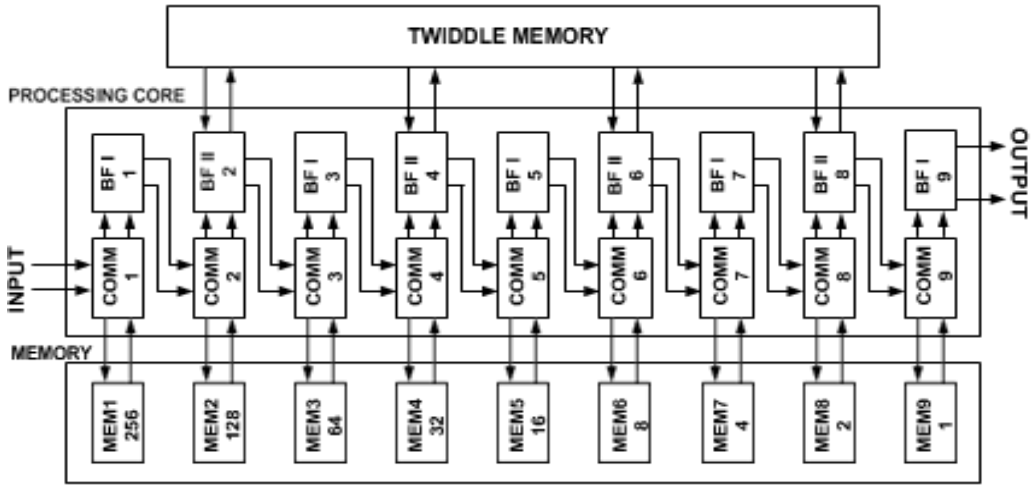


Figure.4: Proposed 1024-point Radix-2^2 FFT architecture containing two parallel data paths.

The butterfly type 1 in Figure 5a is similar to radix-2 containing two adders and subtractors for processing complex number inputs whereas butterfly type 2 is modified for processing H in equation (3) which process entire equation in two phases as shown in Figure 5b. Control signal and multiplexers are used in butterfly type 2 for switching between phases.

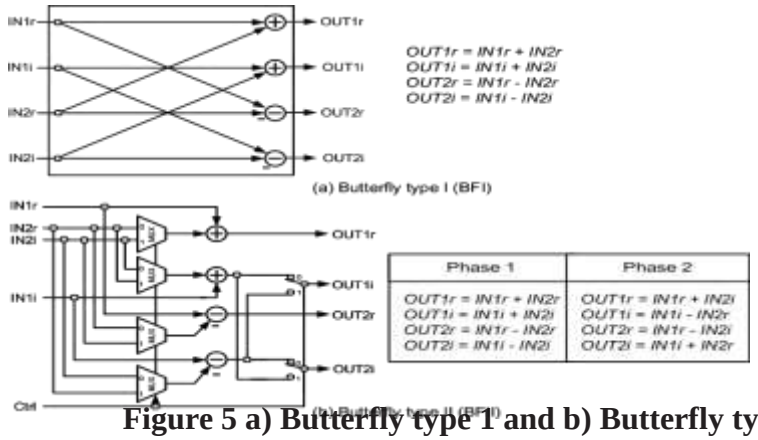


Figure 5 a) Butterfly type 1 and b) Butterfly type 2

The control signal for butterfly type 2 is generated from n bit counter which also acts as address generator for fetching twiddle factors from pre-fixed register memory. Figure 6 shows that the odd numbered butterflies are of type1 butterfly units and even numbered are of type 2 butterfly units.

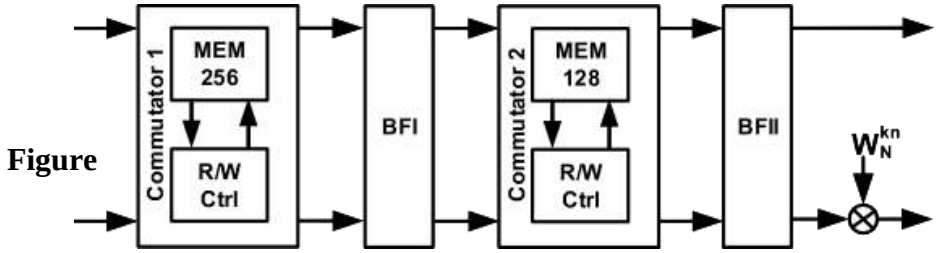


Figure 6 : Radix 2^2 as combination of type 1 and type 2 Butterfly units

The Radix-2^2 FFT algorithm developed in this work decreases the minimum requirement of speed and area when compared to other topologies. The proposed Radix-2^2 multipath delay commutator utilize least multipliers however adders are not small as Radix-2 butterfly structure. When multipliers and adders are compared adders consume low power and area. Considering 1024-point FFT, the proposed

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 5, May : 2023

design saves 50% of the multipliers from multiple delay feedback structures and 30% of memory from radix-2 multiple delay commutator or 60% of memory from radix-4 multiple delay commutators. In the butterfly unit, multiplication is implemented using Baugh wooley multiplier and Vedic multiplier to evaluate the performance of the proposed FFT. Comparison of their performance is presented.

Baugh wooley multiplier:

The Baugh wooley multiplier shown in figure 7 is implemented in the butterfly unit. The Baugh wooley multiplier design is employed in order to multiply the twiddle factor with the butterfly output and its main advantage is regularity in simple array architecture with less area.

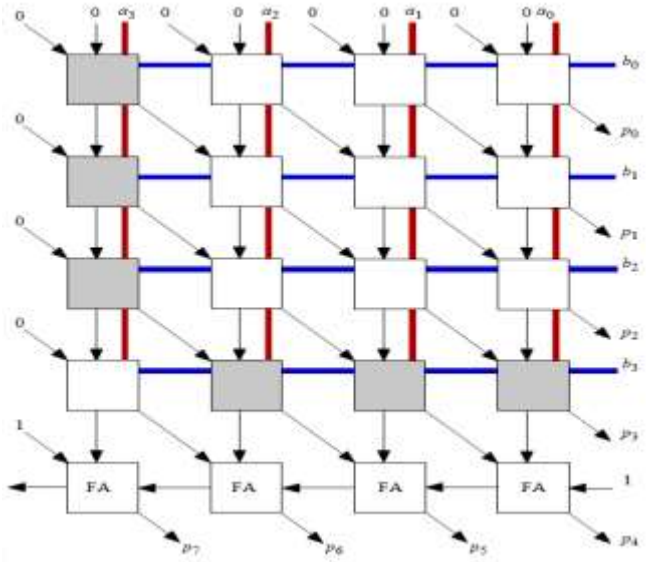
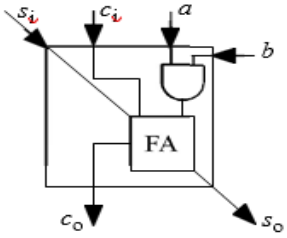


Figure7: Baugh Wooley Multiplier



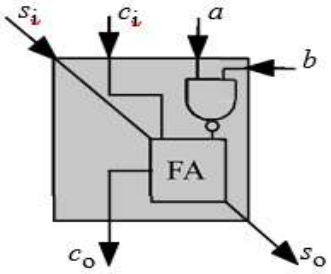


Figure 8 : Baugh Wooley Multiplier white cell **Figure 9: Baugh Wooley Multiplier gray cell**

Vedic multiplier:

In this design, the employment of Vedic multiplier is done and its operation is shown in Figure 10. Vedic multiplier is used to multiply Butterfly type two's output with twiddle factors as it is popular for its regularity in array multiplication with high speed operation.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 5, May : 2023

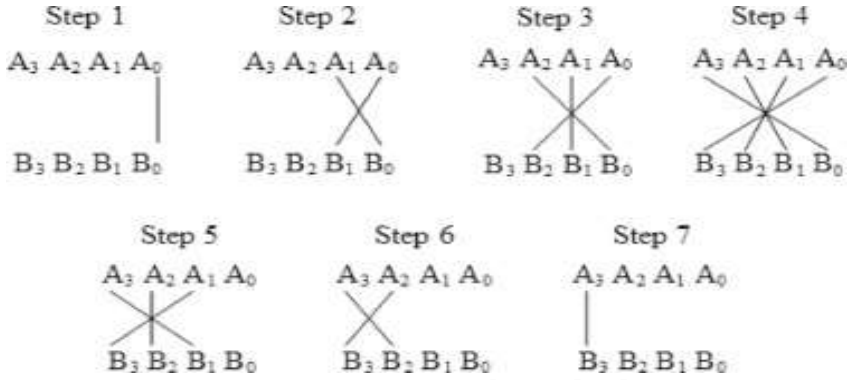


Figure 10: Vedic multiplier operation

Novel Input Scheduling Scheme:

Traditional scheduling approach in Figure 11 is with delay lines and switching networks in which data is shifted in delay network for every clock cycle which consumes notable amount of dynamic power.

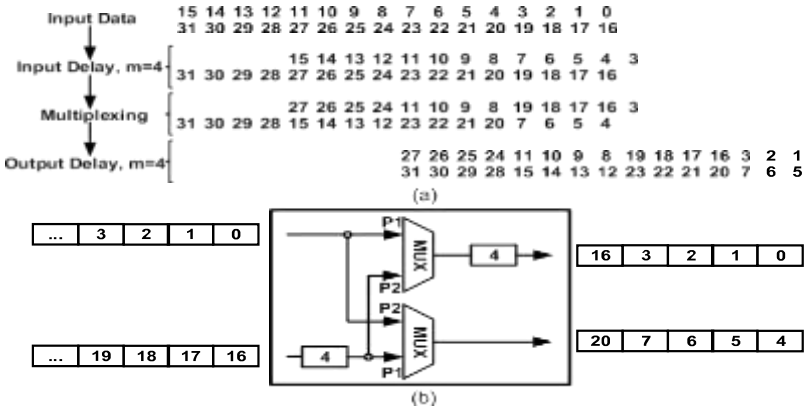


Figure 11: Traditional Input Scheduler

The developed input scheduler shown in Figure 12 performs read/write process in single cycle. It consists of register bank to memory's cell and decoder to convert input address to respective memory location. The input scheduler consists of two uniform memories equivalent to two delay lines of conventional design however here direct memory access is introduced which does not require shifting of data along register bank thereby reducing consumption of dynamic power.

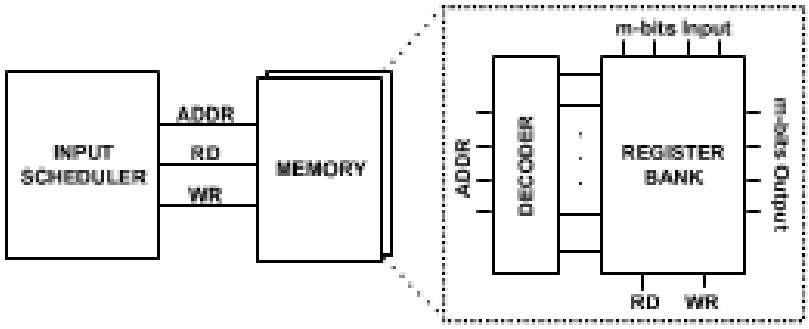


Figure 12 Proposed Input Scheduler Scheme

3. **Results and Discussion**

Radix-2^2 FFT architecture of 8-point, 16-point, 512 and 1024-point using Vedic multiplier and baugh wooley multiplier is implemented on Xilinx Vertx-5 FPGA board. The RTL schematic of Radix-2^2

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 5, May : 2023

1024-point pipeline architecture of MDC with Vedic multiplier is shown in Figure 13. The simulation results of the architecture is shown in the Figure 14.

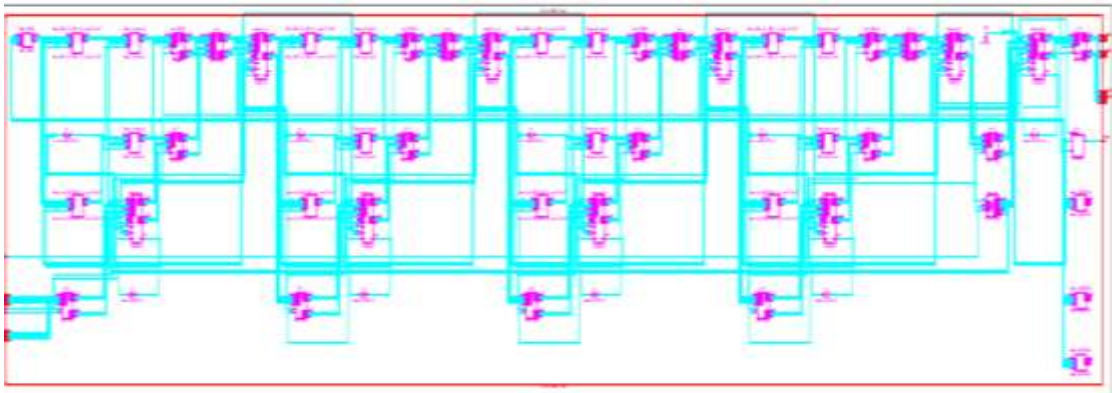


Figure13: RTL schematic of 1024-point radix-2^2 FFT MDC type pipeline architecture.

Figure 14: Timing Diagram

Table 1: Performance comparison of 8-point Radix-2^2 FFT architecture with Vedic multiplier and Baugh Wooley multiplier

Parameters	8-Point Radix-2^2 FFT architecture with	
	Vedic Multiplier	Baugh Wooley Multiplier
Maximum path delay	13.970 ns	20.939 ns
Maximum Frequency	66.006 MHz	38.04MHz
Static Power (mW)	177.32	177.32
No of occupied slices	772	1014
Total input LUTs	1761	2340
No of registers	28	29

Table 2: Performance comparison of 16-point Radix-2^2 FFT architecture with Vedic multiplier and Baugh Wooley multiplier

Parameters	16-Point Radix-2^2 FFT architecture with	
	Vedic Multiplier	Baugh Wooley Multiplier
Maximum path delay	14.66 ns	20.583 ns
Maximum Frequency	56.148 MHz	34.589MHz
Static Power (mW)	177.32	177.32
No of occupied slices	1041	1407
Total input LUTs	2176	2790
No of registers	67	74

Table 3: Performance comparison of 512-point Radix-2^2 FFT architecture with Vedic multiplier and Baugh Wooley multiplier

Parameters	512-Point Radix-2^2 FFT architecture with	
	Vedic Multiplier	Baugh Wooley Multiplier
Maximum path delay	25.712 ns	31.102 ns
Maximum Frequency	27.871 MHz	17.990MHz
Static Power (mW)	177.41	177.41
No of occupied slices	3062	3962
Total input LUTs	7084	7893
No of registers	92	101

Table 4: Performance comparison of 1024-point Radix-2^2 FFT architecture with Vedic multiplier and Baugh Wooley multiplier

Parameters	1024-Point Radix-2^2 FFT architecture with	
	Vedic Multiplier	Baugh Wooley Multiplier
Maximum path delay	42.102 ns	61.204 ns
Maximum Frequency	13.836 MHz	9.04MHz
Static Power(mW)	177.49	177.49
No of occupied slices	5155	5527
Total input LUTs	11629	13057
No of registers	121	122

Area and speed profile:
From the results tabulated in Table 1, implementation of Radix 2^2 architecture of 8-point using Vedic multiplier achieves maximum frequency of 66.006 MHz while with Baugh Wooley multiplier achieves only 38.04 MHz. The proposed architecture with Vedic multiplier achieves reduction of 31.34% in slices occupied and 32% in No of LUTs occupied comparing with Baugh Wooley multiplier. Table 2 shows the comparison results of FFT architecture of 16-point and achieves the maximum operating frequency of 56.148 MHz with Vedic multiplier and 34.589 MHz with Baugh Wooley multiplier. It achieves reduction of slices occupied and LUTs by 35.15% and 28.21%.
Table 3 and 4 shows the comparison results of 512 and 1024-point FFT. For 512-point the maximum operating frequency is 27.871 MHz and for 1024-point is 13.836 MHz. with Vedic multiplier the proposed architecture achieves reduction of 29.39% and 20% in slices occupied and No of LUTs for 512-point. For 1024-ponit FFT it is reduced by 11% and 12% when compared with Baugh Wooley multiplier. From the results, it evident that the proposed FFT architecture with Vedic multiplier achieves high speed and less area when compared with Baugh Wooley multiplier based architecture.

4. Conclusion

The radix-2k algorithm with adaptive and simple control structure is proposed which utilizes multiple delay commutator (MDC) for attaining simple butterfly units and less memory requirements. The architecture is implemented using Vedic multiplier and Baugh Wooley multiplier to perform complex multiplication operations. The proposed architecture with Vedic multiplier gives an advantaged of high speed of operation and occupies lesser area when compared to architecture with Baugh Wooley multiplier. Future scope involves reduction of dynamic power consumption by reducing unnecessary switching in architecture.

References

[1]. Heideman, M.T., Johnson, D.H. & Burrus, C.S. “Gauss and the history of the fast Fourier transform’’. Arch. Hist. Exact Sci.vol.34, pp.265–277, 1985.
[2]. Cooley, James W., and John W. Tukey. “An Algorithm for the Machine Calculation of Complex Fourier Series.” Mathematics of Computation, vol. 19, pp. 297–301, 1965.
[3]. L. Jiang, C. Liu and P. Zhang, "A novel overall in-place in-order prime factor FFT algorithm," 2012 5th International Congress on Image and Signal Processing, Chongqing, pp. 1500-1503, 2012.
[4]. S. Mittal, Z. A. Khan and M. B. Srinivas, "Area Efficient High Speed Architecture of Bruun's FFT for Software Defined Radio," IEEE GLOBECOM 2007 - IEEE Global Telecommunications Conference, Washington, DC, pp. 3118-3122, 2007.
[5]. Shousheng He and M. Torkelson, "Design and implementation of a 1024-point pipeline FFT processor," Proceedings of the IEEE 1998 Custom Integrated Circuits Conference (Cat. No.98CH36143), Santa Clara, CA, USA, pp. 131-134,1998.
[6]. Shruti D. Kale, Prof. Gauri N. Zade “Design of Baugh-wooley Multiplier using Verilog HDL” IOSR Journal of Engineering (IOSRJEN) ISSN (e): 2250-3021, ISSN (p): 2278-8719, Vol. 05, PP 25-29, Issue 10 ,October. 2015.
[7]. J. A. Johnston, "Parallel pipeline fast fourier transformer," IEEE Proceedings, French Communications, Radar and Signal Processing, vol. 130, no. 6, pp. 564-572, October 1983.
[8]. Classification of FFT architectures (online) //ir.nctu.edu.tw//bitstream /11536 /46157 /10/353010.pdf
[9]. M. Ayinala, M. Brown and K. K. Parhi, "Pipelined Parallel FFT Architectures via Folding Transformation," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 20, no. 6, pp. 1068-1081, June 2012.
[10]. G. Bi and E. V. Jones, "A pipelined FFT processor for word-sequential data," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 37, no. 12, pp. 1982-1985, Dec. 1989.

[11]. M. Garrido, J. Grajal, M. A. Sanchez and O. Gustafsson, "Pipelined Radix-2k Feedforward FFT Architectures," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 21, no. 1, pp. 23-32, Jan. 2013.
[12]. E. E. Swartzlander, W. K. W. Young and S. J. Joseph, "A radix 4 delay commutator for fast Fourier transform processor implementation," IEEE Journal of Solid-State Circuits, vol. 19, no. 5, pp. 702-709, Oct. 1984.
[13]. C. R. Baugh and B. A. Wooley, "A Two's Complement Parallel Array Multiplication Algorithm," IEEE Transactions on Computers, vol. C-22, no. 12, pp. 1045-1047, Dec. 1973.
[14]. D. James, "Quantization errors in the fast Fourier transform," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 23, no. 3, pp. 277-283, June 1975.

[15]. P. Gupta, "Accurate performance analysis of a fixed point FFT," 2016 Twenty Second National Conference on Communication (NCC), Guwahati, pp. 1-6.2016.