International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-8 Issue-10, August 2019

FFT using Power Efficient Vedic Multiplier

Nidhi Gaur, Anu Mehra, Pradeep Kumar

Abstract*: *Modern communication systems rely on Digital Signal Processing (DSP) more than ever before. Improving the speed of FFT computation using high speed multipliers will help to enhance the performance of DSP systems. In this paper a DIT FFT architecture using high performance Modified Vedic multipliers is proposed. Vedic Multipliers offer a more efficient way to perform multiplication on large numbers occupying less area and consuming low power and delay The adders used in the Vedic multipliers are Brent Kung based and multiplexer based adders. The right utilization of these adders at different word lengths helps to achieve an architecture with minimal area and power. Comparative analysis of modified 24×24 Vedic Multiplier with existing Vedic Multiplier shows the improvement in performance with respect to power and area. Proposed FFT design is compared with existing designs for dynamic power consumption and an improvement of 46.93% compared to Tsai's FFT Design and 59.37% compared to Coelho's FFT Design is achieved. The entire architecture is implemented on Virtex 7 FPGA and simulated using Xilinx Vivado 2017.4.

Index Terms*: *Digital Signal Processing (DSP), Decimation in Time , Fast Fourier Transform , Vedic Multipliers.

I. INTRODUCTION

Present era of signal processing owes to discrete fourier transform (DFT) and frequency division multiplexing (FDM) based communication systems. DFT is a known technique to find frequency components of a signal from time domain [1]. The time complexity of DFT is defined as 'n^2', where 'n' is the length of transform. Cooley and Tukey projected a new algorithm termed as Fast Fourier Transform (FFT) to reduce the delay and computation time of DFT calculations [2]. The number of stages in FFT calculations is (n/2) $\log_r$ n, provided 'r' is the radix of FFT. This algorithm simplifies calculation by dividing a FFT of size n=n1n2 into FFT's of lower sizes i.e. of n1 and n2 respectively. The FFT is generally divided into size N/2. This is known as radix-2 FFT. Calculation of FFT requires a lot of complex multiplication on floating point numbers. Binary Floating Point IEEE 754 is the standard which is used worldwide for floating point computation in binary format. Multiplier is one of the inevitable component of any FFT design. A number of multipliers exist in literature. Vedic multipliers is a class of multipliers based on Vedic mathematics and are known to offer the benefits like high efficiency, enhanced delay, low power and reduced area compared to conventional multipliers [3, 4].

Nowadays FFT processors are employed in DSP, wireless applications and MIMO-OFDM systems [5, 6] to achieve faster speeds, low power consumption and efficient area utilization. Pipelining has been used as a processing enhancement method for 64 point FFT using 8 point FFT's [7]. In [8] proposed FFT and Inverse FFT processor is 64 point which uses pipeline processing. Multiplication is done in parallel fashion to reduce the hardware cost of the processor. The processor is designed using radix-2 algorithm for 64 point FFTs in OFDM applications. A 256 bit 64 point FFT architecture is proposed in [9] using radix-4 algorithm to calculate the transforms. The design aimed to reduce the delay by reducing the computational complexity of the transform. A Radix 4 booth multiplier is designed in [10] for DSP applications. A 32 point FFT architecture is proposed in [11]. The design is simulated and implemented on Vertex 6 FPGA. Since multiplication is most desired operation in 'FFT processor', the power optimized multipliers, well embedded in FFT, reduces overall power consumption of processor datapath [12]. In [13] a comparison of the performance of FFT architectures designed using conventional multipliers and Vedic multipliers are discussed. It is observed that the conventional multipliers require more time and area than Vedic multipliers and the processing speed increases with the bit length. A 24x24 vedic multiplier using structural modeling is designed for delay optimization by Athira and Renjith in 2017 [14]. Vedic Multipliers are found to be faster in comparison to conventional multipliers [15].

This paper implements a radix-2 8 point FFT processor for DSP applications designed using modified 24×24 floating point Vedic multipliers. The design is optimized for power. Low power modified Vedic multiplier unit for FFT processor is designed using brent kung adders and multiplexers by replacing ripple carry adders in conventional vedic multipliers. Percentage power improvement in new design is almost 60% which is significant.

The paper is presented in six sections including this section. IInd section describes a review of FFT algorithm. Floating Point multiplier used to design FFT using Vedic mathematics is explained in IIIrd Section. Section IV highlights proposed design and its implementation. Vth section presents a discussion over experimental results while the conclusion is presented in last section followed by references.

II. RADIX-2 FFT ALGORITHM

The 'Discrete Fourier Transform (DFT)' transforms a sequence of N complex number, x[n], into another sequence of complex numbers, X[n] which is defined by [16]:

Manuscript published on 30 August 2019.
***Correspondence Author(s)
Nidhi Gaur, Department of ECE, Amity University, Uttar Pradesh, India.
Anu Mehra, Department of ECE, Amity University, Uttar Pradesh, India.
Pradeep Kumar, Department of ECE, Amity University, Uttar Pradesh, India.

$$X(n) = \sum_{n=0}^{N-1} x[n] \cdot e^{-\frac{2\pi nx}{N}} \quad (1)$$

$$X(n) = \sum_{n=0}^{N-1} x[n] \cdot \left[\cos\frac{2\pi nx}{N} - i \cdot \sin\frac{2\pi nx}{N}\right] \quad (2)$$

This employs N^2 complex multiplications which increases drastically as the number of elements in the sequence increase [7]. FFT was proposed to decrease the complexity of the calculation involved.

The Radix-2 DIT FFT algorithm for an 8 bit sequence is presented in Figure 1. X[n] represents output sequence of frequency components, x[n] symbolizes the input sequence and W_N^r is a factor known as the twiddle factor. The Twiddle factor can be calculated as follows:

$$W_N^r = e^{-j\left(\frac{2\pi}{N}\right)r} \quad (3)$$

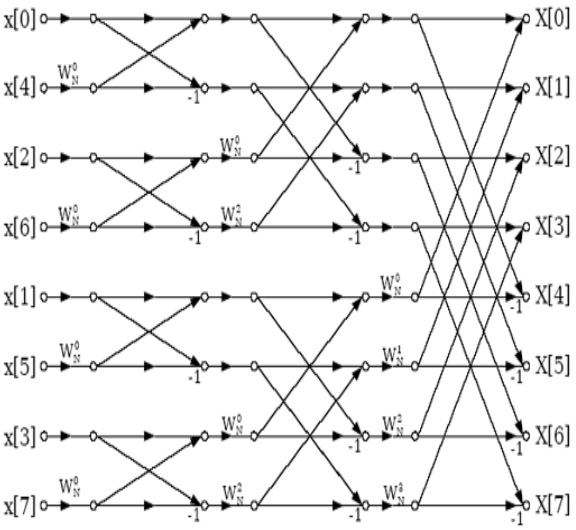

Fig 1. Radix-2 eight point DIT FFT algorithm representation

The algorithm is made up of 12 butterfly structures each containing 1 complex multiplication unit. Thus the complexity of an FFT algorithm can be generalized as (n/2) log2n. A basic 2 point butterfly unit is shown in Fig.2 along with calculation of its output.

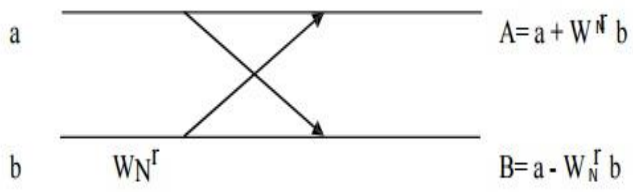

Fig 2. DIT Butterfly Structure

III. VEDIC MULTIPLICATION

Multiplication is the key component in the FFT calculations. Vedic multiplication over the years has emerged as the fastest and low power technique over traditional array multiplier and booth multiplier techniques [13] [14]. Urdhva-Triyakbhyam Sutra is the most commonly used Sutra in Vedic mathematics for multiplication of higher bit numbers. Urdhva stands for vertically and Triyakbhyam stand for crosswise. Vedic multipliers generally divide large numbers into smaller numbers to ease calculations. Initially a 3x3 multiplier is implemented using Urdhva-Triyakbhyam Sutra. The 3x3 multiplier comprises of 2 half adders, 2 full adders and two 2 bit binary adders (Figure 3).

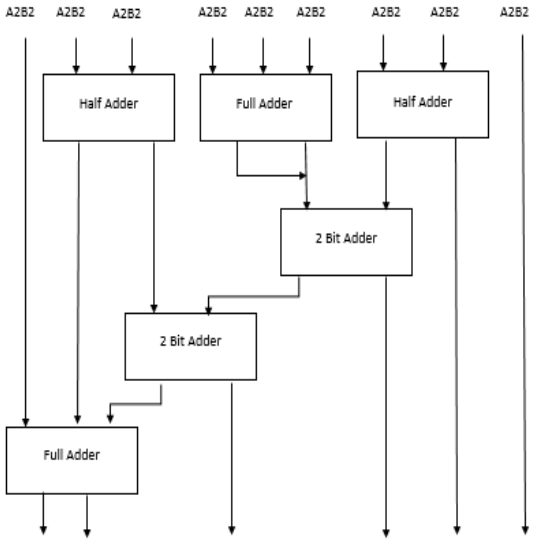

Fig 3. 3x3 Vedic Multiplier Architecture

A 6x6 multiplier is then designed using four 3x3 multipliers and three adders. Then, a 12x12 is generated using four 6x6 multipliers and three adders. Similarly a 24x24 multiplier is designed using four 12x12 multipliers and three adders.

IV. IMPLEMENTATION OF PROPOSED DESIGN

The conventional Vedic multiplier uses Ripple Carry Adders. Ripple Carry Adder has a limitations that it takes excess time to propagate the carry which effects its delay performance adversely. On comparison with other adders, we see that Brent Kung Adders are fast but requires lot of area. As the size of adder increases the area increases drastically. On the other hand, Adders using multiplexers have comparatively low area but the delay is comparatively high for small adders. Thus to obtain feasible results in terms of area and delay, Brent Kung adders are used to design smaller Vedic multipliers and Multiplexer based adders are used to design 24x24 Vedic Multipliers. The architecture of the modified Vedic Multiplier is depicted in Figure4.

24x24 Vedic Multiplier is then used to design a Floating Point multiplier. The single precision Floating point representation is a 32 bit representation with 23 bits for the mantissa, 8 bits for exponent and 1 bit for denoting the sign of the number. The multiplication of a floating point number can be summarized in three steps:

- Addition of the exponential components.
- Multiplication of the mantissa components. The mantissa is stored discarding the most significant 1. Thus concatenation of this bit will result in a 24 bit mantissa.
- Ex-oring of the sign bits.

Retrieval Number: I8922078919/19©BEIESP
DOI: 10.35940/ijitee.I8922.0881019
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

The Floating Point Multiplier is then used to design a simple Butterfly Unit. The butterfly unit also comprises of a floating point adder and floating point subtractor as shown in Fig 5. The 8 point DIT FFT design is implemented using the butterfly units as shown in Fig 6. Design is further extended to 16 point FFT unit.

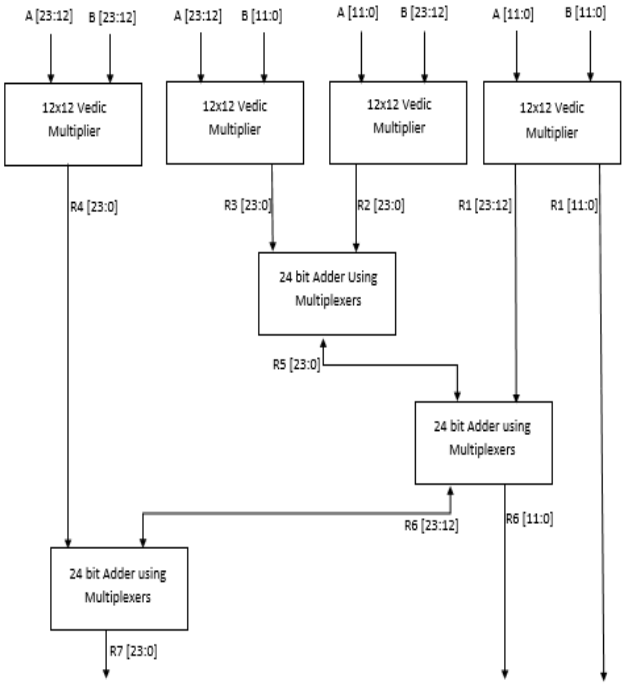

Fig 4. Modified 24x24 Vedic Multiplier

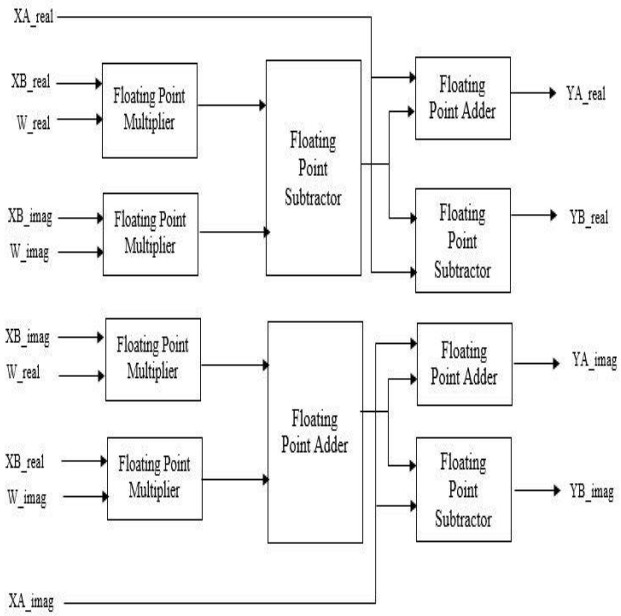

Fig 5. Butterfly Unit Block Diagram

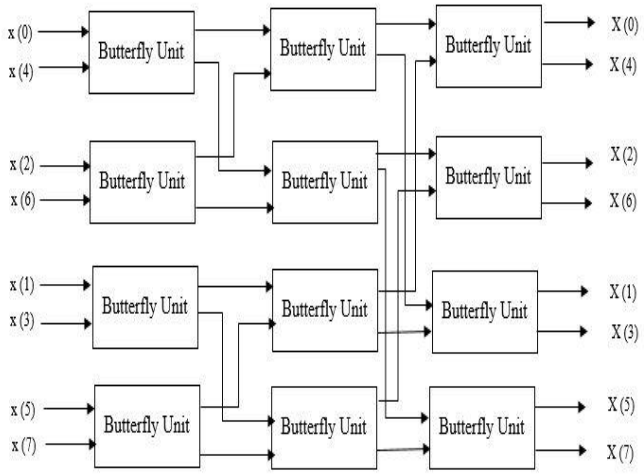

Fig 6. 8 point FFT Processor Block Diagram

V. EXPERIMENTAL RESULTS

Figure 7 depicts RTL schematic of modified Vedic multiplier. As mentioned earlier, 24x24 Vedic Multiplier uses Brent Kung Adders for smaller bit additions and Adders using multiplexers as the 24 bit adder.

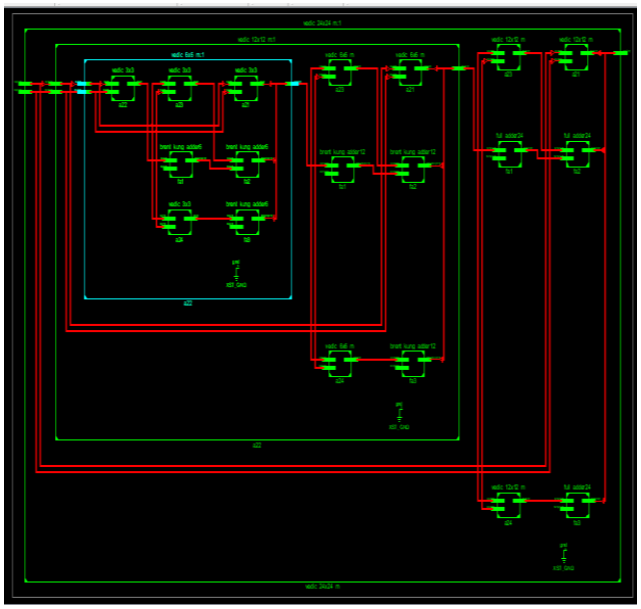

Fig 7. RTL Schematic of Modified Vedic Multiplier

The simulation results of the modified 24x24 Vedic multiplier implemented on Virtex 7 are shown in Fig.8. 24x24 Base Vedic Multiplier with RCA[5] is also implemented for comparison and comparative analysis of the device utilization, power and delay of the modified Vedic multiplier with base Vedic Multiplier is presented in Table 1. The design is found to be low power and low area with delay penalty.

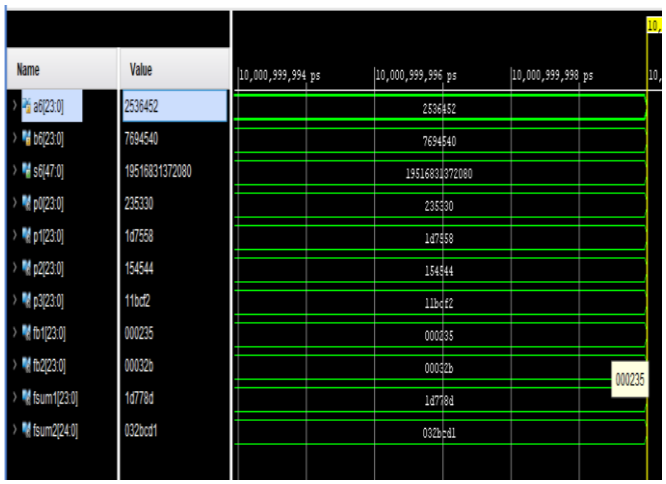

Fig 8. Simulated waveform output of 24x24 Vedic Multiplier

TABLE 1.Comparative analysis of base and proposed Vedic multipliers.

Features	Conventional 24x24 Vedic Multiplier [12 implemented on Virtex 7]	Modified 24x24 Vedic Multiplier
Slice LUTs	984	960
IOBS	96	96
Delay	14.937	18.458
Dynamic Power	23mW	19mW
Static Power	243mW	243mW

The RTL schematic of the proposed 8 point FFT architecture is depicted in Figure9. The design consists of 12 butterfly units. Each butterfly unit consists of four floating-point Vedic multipliers, 4 floating-point adders and 4 floating-point subtractor units. The simulation results of FFT design are shown in Fig.10. The input sequence is read from Block RAM of FPGA. The input sequence consists of eight elements 64 bits long, each containing a 32 bit real and 32 bit imaginary part. The output sequence similarly has 8 elements 64 bits long. It is stored in a file for verification. Similarly the schematic of proposed 16 point FFT design is depicted in Figure 11 and simulation results are demonstrated in Figure12.

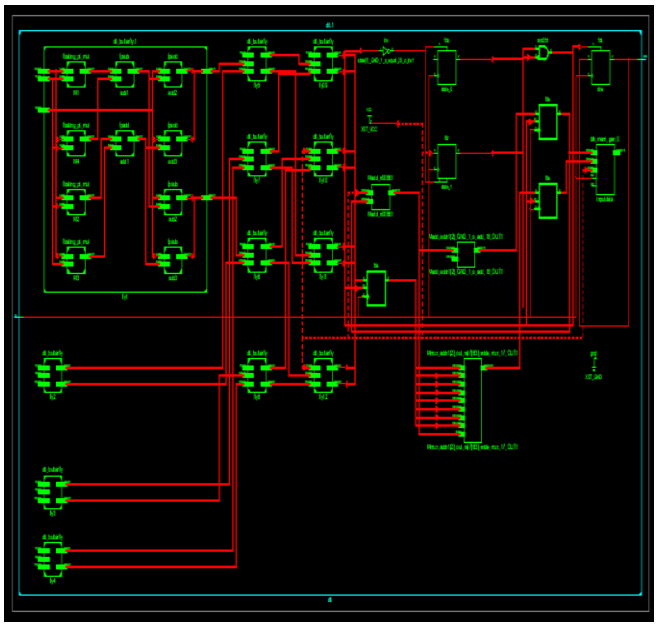

Fig 9. RTL Schematic of Proposed 8 point FFT design

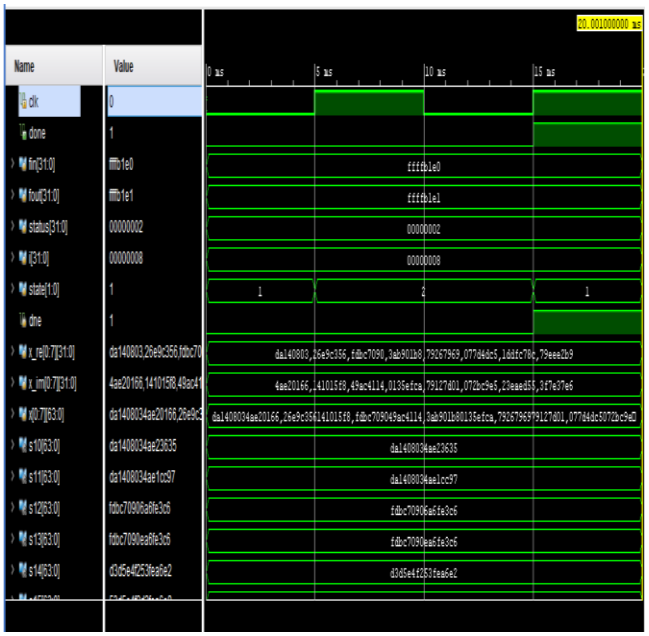

Fig 10. Simulation results of proposed 8 point FFT design

Retrieval Number: I8922078919/19©BEIESP
DOI: 10.35940/ijitee.I8922.0881019
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

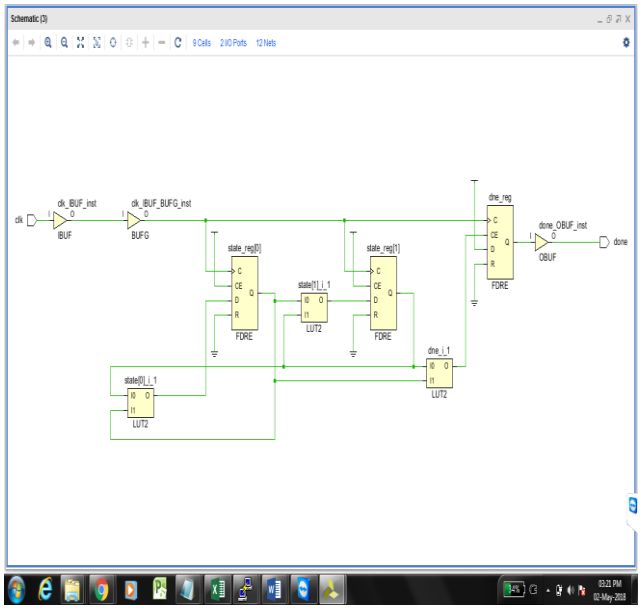

Fig 11. Schematic of proposed 16 point FFT design

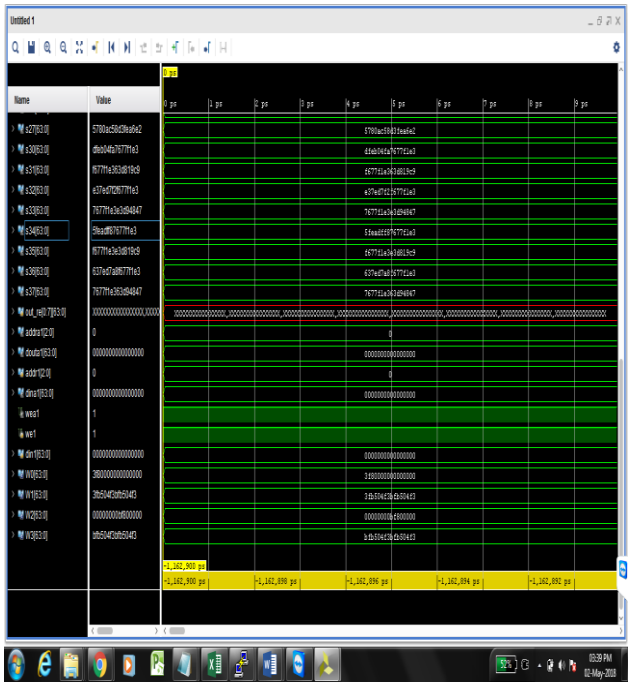

Fig 12. Simulation Results of proposed 16 point FFT design

The design has been implemented on Virtex 7 7vx485tffg1157 FPGA device and simulated using Xilinx Vivado 2017.4. The device utilization and dynamic power utilization of the proposed FFT design is compared with previous designs in literature in Table 2. The entire architecture is designed and implemented using Verilog HDL.

TABLE 2. Comparative analysis of Proposed FFT design

Features	Tsai's FFT Design [5]	Saponara's FFT design [6]	Coelho's FFT Design [1]	Proposed 8 point FFT Design	Proposed 16 point FFT Design
Size of FFT	64	128	12	8	16
Technology	Virtex 4	Virtex 4	Virtex 6	Virtex 7	Virtex 7
Slice LUTs	7549	16124	2924	10942	19485
Slice Registers	NA	NA	NA	131	131
No. of DSP48	16	152	0	0	0
RAM Blocks	NA	152	NA	2	2
IOBs	NA	NA	NA	2	2
Dynamic Power	147mW	NA	192mW	78mW	86mW

VI. CONCLUSION

In this paper, eight and sixteen point DIT FFT Architecture is proposed using modified Vedic Multipliers. Vedic multipliers have lower power, delay and area compared to traditional array multipliers and Booth Multipliers. And multiplication being a key aspect of FFT calculations, the use of the Vedic multipliers helps to improve the performance and the time taken to perform the calculations. A 24x24 low power Vedic multiplier is proposed in present paper for FFT calculations. The proposed FFT consumes almost 60% lesser power in comparison to previous designs.

REFERENCES

1. Diego F. G. Coelho, Renato J. Cintra, Nilanka Rajapaksha, Gihan J. Mendis, Arjuna Madanayake, Vassil S. Dimitrov: 'DFT Computation Using Gauss-Eisenstein Basis: FFT Algorithms and VLSI Architectures', *IEEE Transactions on Computers*, 2017, 66 (8), pp. 1442-1448, doi: 10.1109/TC.2017.2677427
2. J.W. Cooley, J.W. Tukey: 'An algorithm for the machine calculation of complex Fourier series', *Mathematics of Computation*, 1965, 19 (90) pp. 297-301.
3. Yogita Bansal, Charu Madhu: 'A novel high-speed approach for 16x16 Vedic Multiplication with compressor adders' *Computers and Electrical Engineering*, 2016, 49, pp39–49.
4. A.Ronisha Prakash & S.Kirubaveni: 'Performance Evaluation of FFT Processor Using Conventional and Vedic Algorithm', Proc. International Conference Emerging Trends in Computing, Communication and Nanotechnology, 2013, pp.89 – 94.
5. P.-Y. Tsai, C.-W. Chen, and M.-Y. Huang: 'Automatic IP generation of FFT/IFFT processors with word-length optimization for MIMO-OFDM systems', *EURASIP Journal on Advances in Signal Processing*, Jan. 2011, 2011, pp. 1–15, doi:10.1155/2011/136319.
6. S. Saponara, M. Rovini, L. Fanucci, A. Karachalios, G. Lentaris, and D. Reisis: 'Design and comparison of FFT VLSI architectures for SoC telecom applications with different flexibility, speed and complexity trade-offs', *Circuits, Systems, and Signal Processing*, 2012, 31 (2), pp.627–649, DOI 10.1007/s00034-011-9332-
7. J. M. Rudagi, R. Lobo, P. Patil, N. Biraj and N. Nesaragi: 'An efficient 64-point pipeline FFT engine', Proc. IEEE ARTCC Conf., Kottayam, India, Oct. 2010, DOI: 10.1109/ARTCom.2010.31.
8. C. Yu, M. H. Yen, P. A. Hsiung, and S.-J. Chen: 'A low-power 64-point pipeline FFT/IFFT processor for OFDM applications', *IEEE Trans. Consumer Electron.*, Feb. 2011, 57 (1), pp. 40-45.
9. Sudha kiran G, brundavani p: 'FPGA Implementation of 256-Bit, 64-Point DIT-FFT Using Radix-4 Algorithm', *International Journal of Advanced Research in Computer Science and Software Engineering*, September 2013, 3 (9), pp. 126-133.
10. Nagamani A N, Manish Nagaraj, Nikhil R, Vinod Kr. Agrawa: 'Reversible Radix-4 Booth Multiplier for DSP Applications' 2016 International Conference on Signal Processing and Communications (SPCOM), Bangalore, India, 12-15 June 2016, doi: 10.1109/SPCOM.2016.7746687.

11. Asmitha Haveliya: 'Design and Simulation Of 32-Point FFT Using Radix-2.Algorithm For FPGA Implementation', 2012 Second International Conference on Advanced Computing & Communication Technologies, Rohtak, Haryana, India, 7-8 Jan. 2012, pp. 167-171, doi: 10.1109/ACCT.2012.43.
12. Avinash Kumar Singh, Ashutosh Nandi: 'Design of Radix 2 Butterfly Structure using Vedic multiplier and CLA on Xilinx', Conference on Emerging Devices and Smart Systems (ICEDSS), Tiruchengode, India, March 2017.
13. Laxman P.Thakre, Suresh Balpande, Umesh Akare, Sudhir Lande: 'Performance Evaluation and Synthesis of Multiplier used in FFT operation using Conventional and Vedic algorithms', Third International Conference on Emerging Trends in Engineering and Technology (ICETET), Goa, India, Novemeber, 2010.
14. M.S Athira Menon, R.J Renjith: 'Implementation of 24 bit high speed floating point Vedic multiplier', International Conference Networks and Advances in Computational Technologies (NetACT), Thiruvanthapuram, India, July 2017.
15. Swapnil Badar, D.R Dandekar: 'High speed FFT processor design using radix −4 pipelined architecture', International Conference on Industrial Instrumentation and Control, May 2015.
16. Tapsi Gupta, Janki Ballabh Sharma: 'A High-Speed Single-path Delay Feedback Pipeline FFT Processor using Vedic-Multiplier', International Conference on Information, Communication, Instrumentation and Control (ICICIC), Indore, India, August 2017.

AUTHORS PROFILE

Ms. Nidhi Gaur is currently working as Assistant Professor in Department of Electronics and Communication Engineering, ASET at Amity University, Uttar Pradesh. She has done her M.Tech in VLSI Design from Banasthali University Rajasthan. Her area of research is Front end VLSI design, communication and networking. Currently she is pursuing PhD from Amity University, uttar Pradesh.

Dr. Anu Mehra received her Masters in engineering from K.U.Leuven, Belgium in the year 1998. She received Ph. D in "High Energy physics" from Jamia Millia Islamia, Delhi ,India in 2004. After the completion of PhD she went for a post Doc to IMEC, Belgium. She also did a complementary Masters in Engineering there. She joined Amity University, Noida, India in the year 2004. Currently she is working as Professor in Amity School of Engineering and Technology in the Department of Electronics and Communications Engineering.

Dr Pradeep Kumar is currently working as an Associate Professor in the Department of Electronics and Communication Engineering, Amity University Uttar Pradesh, Noida, India. He has received his Ph. D. degree from Garhwal University Srinagar (Garhwal) Uttaranchal, India, in 2006. Dr Kumar's has 14 years of teaching and research experience, most recently focusing on VLSI, microelectronics, device modelling and simulation, design and verification using Verilog and systemverilog.