

MATHEMATICAL FOUNDATIONS OF FAST FOURIER TRANSFORM DESIGN AND ANALYSIS: A COMPUTATIONAL PERSPECTIVE AT THE PHYSICAL LEVEL

¹ Leena Maria S, ² Ganavi S B

¹ Assistant Professor, Department of Mathematics, Government Engineering College; Hassan,
Karnataka, 573201, India

¹ leenamaria668@gmail.com, ² manjuganavi2017@gmail.com

Abstract:

This paper presents an attempt is made to combine the ideas of Vedic mathematics with the technology of very large-scale integration (VLSI) 90nm in order to create and evaluate a physical level device for the Fast Fourier Transform (FFT). The application of Vedic mathematics has the ability to increase the speed at which computations are performed by providing computational advantages over traditional methods in some domains. Through the utilization of Vedic Mathematics in the development of a Fast Fourier Transform (FFT) chip, it is possible to enhance the efficiency of computation, hence reducing the amount of time that is required. Considering the importance of FFT in a wide variety of digital signal processing applications, time optimization is considered to be of the utmost importance. Hardware Description Language (HDL) code is utilized in the implementation of the Vedic Mathematics method, and the design compiler tool developed by Synopsys, Inc. is utilized to assess the results of the implementation.

Key words: VLSI, nanotechnology, mathematics, binary sequences, computational efficiency.

I. Introduction:

Through the use of discrete Fourier transforms, a discrete signal that is in the time domain can be transformed into a discrete signal that is stored in the frequency domain that corresponds to it [1]. In comparison to Discrete Fourier Transforms, the Fast Fourier Transform is a more efficient version of the latter [1]. Multiplication and addition of numbers are two of the essential processes that are required for the creation of a Fast Fourier Transform device. A decrease in the complexity of the mathematical operation of multiplication might lead to a reduction in the complexity of the circuit as a whole. Vedic mathematics is an ancient mathematical system that originated in India. It offers a wide variety of procedures and techniques that can be utilised for computational task completion [3]. These procedures and techniques can be traced back to the Vedas, which are a collection of ancient Indian scriptures [3]. The Urdhva Tiryakbhyam Sutra and the Nikhilam Sutra are two examples of the many diverse multiplication techniques that are included in Vedic mathematics [3]. For the purpose of multiplying two binary values, this work makes use of the Nikhilam Sutra approach, which is derived from Vedic mathematics. Following the completion of the multiplication procedure, the module is then incorporated into the addition module in order to enable the execution of Fast Fourier Transforms. The HDL programming language is used to develop the entire module's code, and

the design compiler tool supplied by Synopsys Inc. is utilised for the purpose of designing and analysing the circuit.

In the realm of signal processing, the transformation of signals from the time domain to the frequency domain is a fundamental task with widespread applications. This transformation enables the analysis and manipulation of signals in various domains, ranging from telecommunications to image processing. At the heart of this transformation lie the discrete Fourier transform (DFT) and its computationally efficient counterpart, the Fast Fourier Transform (FFT). While FFT algorithms have revolutionized signal processing by drastically reducing computational complexity, further optimizations are continually sought to enhance efficiency and performance.

In recent years, researchers have begun exploring alternative methodologies to streamline FFT circuit designs and improve computational efficiency. One such approach involves the integration of techniques derived from ancient mathematical systems, such as Vedic mathematics. Originating from ancient India, Vedic mathematics encompasses a rich array of mathematical principles and techniques aimed at rapid and efficient computation. Among these techniques, the Nikhilam Sutra stands out as a systematic method for multiplication, promising to simplify complex computational tasks.

This research paper delves into the integration of Vedic mathematics techniques, particularly the Nikhilam Sutra, into FFT circuit design. By leveraging the principles of Vedic mathematics, this study seeks to optimize binary multiplication operations within FFT circuits, ultimately aiming to reduce computational complexity and enhance overall efficiency. Through a comprehensive exploration of this integration, the paper aims to contribute to the ongoing efforts to advance FFT circuitry and signal processing methodologies.

The significance of FFT algorithms in signal processing has been well-established in the literature, with numerous studies focusing on optimizing FFT circuit designs for improved efficiency and performance. Cooley and Tukey's seminal work in 1965 introduced the FFT algorithm, laying the foundation for its widespread adoption in signal processing applications [9]. Subsequent research by Rader in 1968 further explored the implementation of DFT when the number of data samples is prime, providing valuable insights into FFT computation [10]. In parallel, the field of Vedic mathematics has garnered attention for its potential applications in computational tasks. Masuzawa and Kawamura (2005) investigated the utilization of Vedic multiplication architecture based on ancient Indian mathematics, demonstrating its effectiveness in streamlining multiplication operations [11]. The Nikhilam Sutra, in particular, has been identified as a promising approach to simplify binary multiplication tasks, offering potential reductions in computational complexity [11].

However, while previous research has examined the individual components of FFT circuitry and Vedic mathematics techniques, limited literature exists on the integration of these methodologies. This research paper aims to bridge this gap by exploring the synergies between FFT algorithms and Vedic mathematics principles, particularly the Nikhilam Sutra, in the context of FFT circuit design. By examining the potential benefits and challenges of this

integration, the paper seeks to contribute to a deeper understanding of efficient signal processing methodologies and pave the way for future advancements in FFT circuitry.

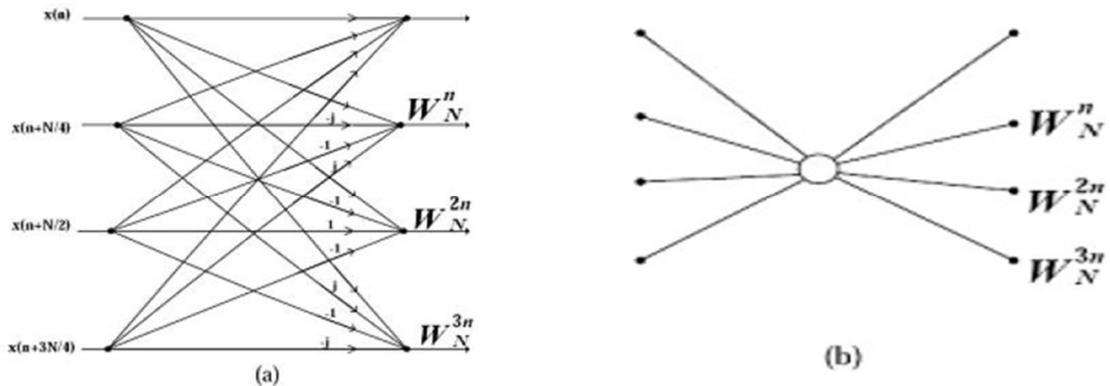
These days, a lot of application lines and products leverage the highly developed fields of audio and communications signal processing. The Discrete Fourier Transform (DFT) algorithm's arithmetic complexity has a major role in worldwide computational expenses since digital communications is a field with a lot of activity. The widely used radix-2 Fast Fourier Transform (FFT) technique was created by Cooley and Tukey [12] to lessen the DFT's computational burden. Given N points, the Discrete Fourier Transform (DFT) X(k) is given by

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk} \quad \{0 \leq k < N-1\}, W_N^{nk} = \exp(-j2\pi nk/N)$$

where the time-domain and frequency-domain sequences are represented by X(k) and X(n). Rather than directly implementing equation (1), the FFT algorithm recursively factors a large point DFT into several small point DFTs to minimize the overall functioning.

Decimation in Time (DIT) and Decimation in Frequency (DIF) FFT are two popular forms of decompositions. The bit reverse order input that DIT begins with and produces normal order output is the only distinction between these two methods. However, DIF produces output that is somewhat reversed order after beginning with normal order input. The DIF algorithm is used throughout this paper. The traditional Fast Fourier Transform FFT computation method requires N(N-1) complex additions and multiplications of N2. The same calculation using (N/2)log2N complex multiplications and (N)log2N complex additions is carried out by the radix-2 Cooley-Tukey method. However, using a radix-4 FFT technique rather than radix -2 logarithms is computationally more efficient.

Keep in mind that each N/4-pointDFT's input is a linear mixture of four scaled signal samples



The complete butterfly operation for Radix- 4 DIF is shown in figure 1 (a) and in a more compact form in figure 1(b) [13].

The ancient Indian mathematical discipline known as VEDIC mathematics [16] focuses mostly on Vedic mathematical equations and how to apply them to different areas of mathematics. The term "Vedic" comes from the word "Veda," which signifies the repository of all knowledge. After studying the Vedas for eight years, Sri Bharati Krisna Tirtha (1884–1960) rebuilt Vedic

sends read address signal. In the end, a little reverse scrambling is done once these twiddle factors are added to the butterflies' output. It is observed that memory remapping is required in the FFT implementation that remapping, though, is easily accomplished. For example, the entrance for the FFT radix-4 DIF must be written in memory addresses 0, 1, 2, 3, 4, 5, 6, and 7. Following the processing of a phase of scrambling, it must write in 0, 4, 2, 6, 1, 5, 3, and 7. That scurrying has a very clear sequence to it. When designing the 1024 point FFT processor, the entire architecture module is used five times. This can be explained using the formula $\text{Stage} = \log_4(\text{computing point}) \cdot 3$. Using the Sine-Cosine Lookup Table Algorithm, design an FFT processor.

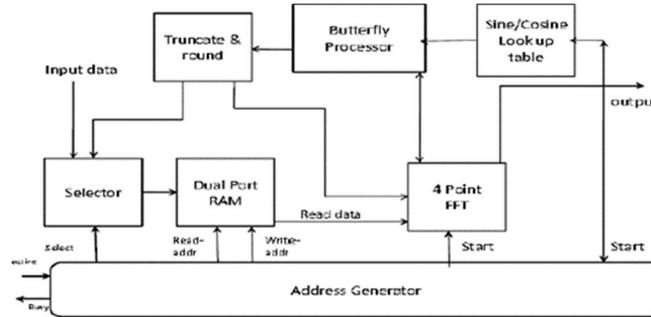


Figure 3 illustrates the FFT processor's design flow utilizing a sine-cosine lookup table.

Design of FFT Processor using Xilinx Core .The 1024 point FFT transform is computed in this part using the Xilinx FFT core. The Xilinx FFT core enables multiple arithmetic computations and provides many designs. Figure 4 displays the Core FFT architecture. Numerous initializations were made when implementing FFT Core. The FFT core in this solution uses a single radix-4 butterfly processing engine and has two processes, thus burst I/O architecture, for example, is used. Data loading and/or unloading is one procedure. The computation of the transform is the second step. Processing and data I/O do not happen at the same time. The data is loaded into the FFT at startup[18].

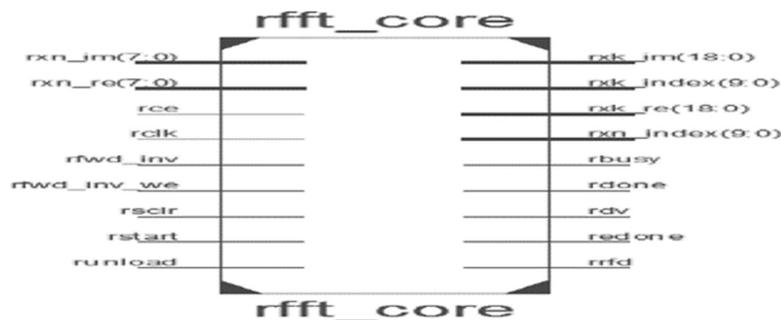


Figure 4: Architecture of Xilinx FFT core

The inputs are provided as 8-bits fixed-point data type. Coefficients are internally saved in the core and are also represented as 8-bit fixed-point data. We apply full-precision unscaled arithmetic, which takes into account the number of bit growth at each stage. In order to determine the necessary bits for correct representing the outputs, the core applied the formula [8]:

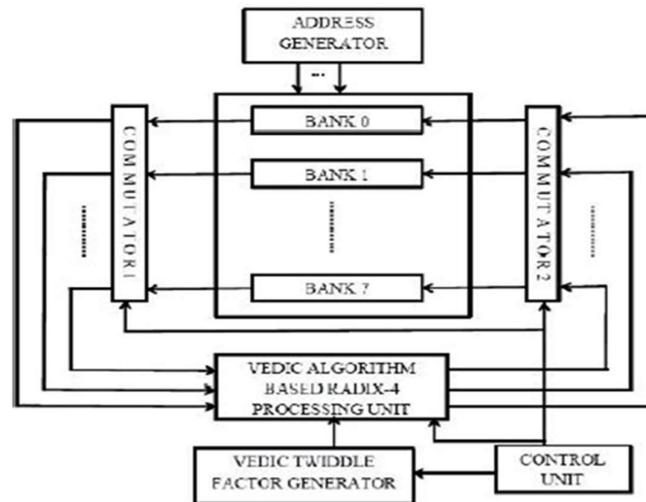


Figure 5: Architecture of 64-point FFT using Vedic algorithm.

The Vedic algorithm is a well-known and age-old method for doing mathematical operations. The multiplication technique known as "Urdhvatiryagbhyam" (vertical and across) is employed. This algorithm multiplies two supplied numbers in a vertical and crosswise fashion until it is left with only MSB bits. The suggested FFT performs complex twiddle factor multiplications using the Urdhvatiryagbhyam method of the Vedic methodology. algorithm [13]. Figure 8 shows the device utilization summary of 1024 point FFT processor using Sine- Cosine lookup table. Figure 9 shows the device utilization summary of 1024 point Xilinx Core FFT Processor [13]

II. Simplifying Binary Multiplication with Vedic Math:

In order to multiply two integers using the conventional method, it is necessary to do $O(n^2)$ piece operations [5]. In order to alleviate this level of complexity, Vedic mathematics provides a solution. When it comes to the construction of circuits for Fast Fourier Transforms, putting an emphasis on reducing the multiplication process will have important repercussions. The architecture of the circuit that performs the Fast Fourier Transform comprises modules that do addition as well as multiplication. The Nikhilam Sutra, a method of multiplication that originates from Vedic mathematics, is utilised in this investigation to perform the operation of multiplying two binary values.

A) Method of Nikhilam Sutra:

To begin the process of multiplying two numbers, X and Y, using the Nikhilam sutra, the first step involves selecting the smaller of the two numbers. The second step is to determine the base, which should be equal to the smallest number, represented as (10^N-1) . Next, subtract the chosen base from both X and Y. This subtraction constitutes the third stage of the process.

In the fourth step, repeat the preceding steps until either X or Y is evaluated to be either 1 or 0.

$$X \times Y = B \times (X + y) + (x \times y) \tag{1}$$

$$X \times Y = B \times (Y + x) + (x \times y) \tag{2}$$

$$x = X - B \tag{3}$$

$$y = Y - B \tag{4}$$

When X is greater than Y, equation 1 is utilised, and when Y is greater than X, equation 2 is utilised. Both equations are used to determine something. The letter N denotes the total amount of bits, while the letter B stands for the base number that has been chosen. According to the procedure described in Sitaramiah Venkataramana Srinivasan and Abdul Razak's Physical Level Design and Analysis of Fast Fourier Transforms using Vedic Mathematics, the process of multiplication continues to iterate until the value of either x or y reaches either 1 or 0. This single-bit determination brings the process to a close, with the number of steps varied according to the multiplicands that are involved.

III. Fast Fourier Transforms:

The amount of computations that are necessary for an n-point radix can be reduced from $2n^2$ to $2n \log_2 n$ with the use of the Fast Fourier Transform (FFT) [5]. The response of a fast Fourier transform (FFT) will be a sinc function [5] if the function that is to be converted does not have a harmonic relationship with the sampling frequency. An illustration of the butterfly structure of an

8-point radix of the Fast Fourier Transform may be found in picture 1. The construction of a circuit for FFT computations requires a less number of logical gates in comparison to that of Discrete Fourier Transforms.

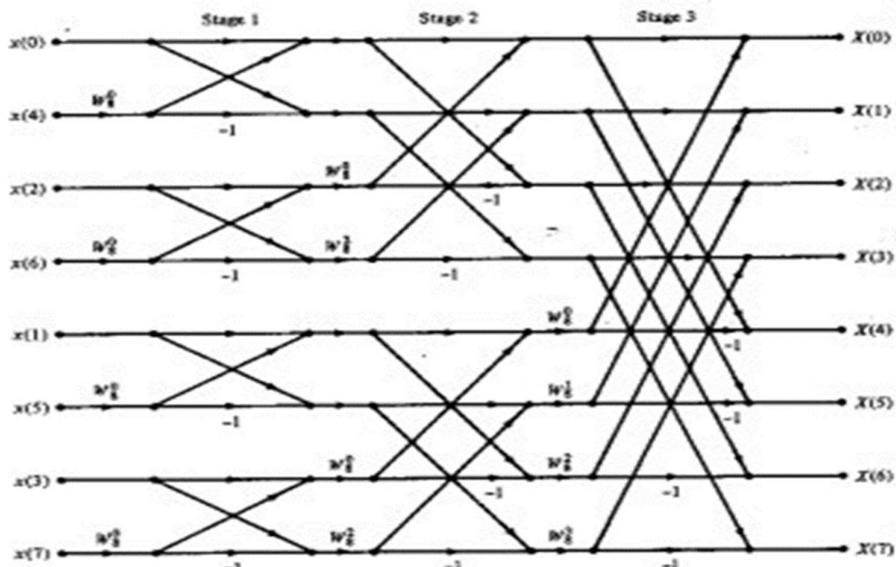


Fig. 1. 8 Point Fast Fourier Transform

IV. Design Concerning Effective Fourier Transformations:

Implementing modules for one-bit multiplication, addition, subtraction, and comparator circuitry, the Fast Fourier Transform circuit was effectively developed utilising a Vedic multiplier. The Fast Fourier Transforms circuit had the Nikhila algorithm integrated into it, which was used for multiplication. All of the circuitry for Fast Fourier Transforms had been written, including the code, to use the Nikhila algorithm's logic for multiplication, and other gate elements had been programmed to do different duties. The time and effort saved by using this method to apply Fast Fourier Transforms was substantial.

The VERILOG HDL language was used to code the programme for the four-point Fast Fourier Transform. The Fast Fourier Transform module, as well as the Nikhila Sutra algorithm's code, were both developed and modelled in the VERILOG language. At both the generic technology (GTECH) and gate levels, a Graphical User Interface (GUI) for synthesis, analysis, and design was created using the Design Vision tool from Synopsys, Inc.

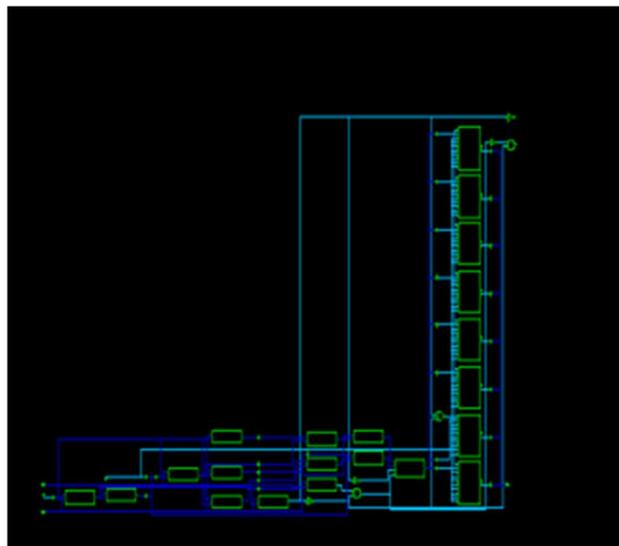


Fig. 2 Four Point Fast Fourier Transform circuit.

Figure 2 shows the top-level schematic diagram of the Fast Fourier Transform circuit generated using the design vision tool from Synopsys, Inc. A synthesis tool reads the Fast Fourier Transform hardware description in Verilog and processes it at the Register Transistor Level. In addition, the synthesis tool takes input from a typical cell library. The saed90nm_typ_ht.db library, which deals with 90 nanometer technology, is used in this paper. A fully structural description is provided as an output by the input in the form of a gate-level netlist. Steps like defining optimisation constraints, synthesising to gates, and creating different area reports are carried out by the Synopsys Inc. Design Compiler. The implementation of the Register Transistor Level makes use of a standard library that is compatible with 90nm technology. The result of the 90nm cell library is a basic netlist. A completely structured representation with just standard cells at the configuration's leaves is produced by this entry-level netlist. At its core, the programme performs a plethora of computations, such as an integrated circuit map, place, route, and floor design.

Synopsys Inc.'s IC Compiler is a tool for designing physical level chips for FFT circuits. As a single, focused, chip-level physical execution tool, it integrates different design stages to arrange transistors according to their final performance. This enables efficient execution through features such as low-power capabilities, tracking, mapping, location, route, floor planning, clock tree synthesis, and manufacturability. Figure 3 depicts the completed Fast Fourier Transform IC compilation.

V. Methodology:

The Fast Fourier Transform circuit is created and connected to the actual pins of the integrated circuit using a procedural process called mapping. By doing so, it establishes the circuit's input and output pins and allows for Boolean logic communication both internally and with external devices. Another important idea in chip design is "place and route," which guides the placement of transistors and the shortest path for miniature devices. The results of both the placement and routing operations are dependent on one another, and they take place simultaneously. To get the most efficient production, it is necessary to take this interdependence into account.

Input Bit Length for 4 point FFT radix	Number of calculations			
	Conventional		Vedic	
	Addition	1 bit Multiplier	Addition	1 bit Multiplier
2	16	64	8	8
3	56	324	40	18
4	120	1024	72	32
8	616	16384	424	128

Table – I -Comparative analysis of Vedic and conventional

The table above illustrates the demand of the multiplication circuit for the Vedic method compared to the conventional method in the context of four-point Fast Fourier Transforms. The difference in the need for the multiplier circuit compared to the standard procedure increases proportionally with the length of the input bit. In Vedic mathematics, the Nikhilam Sutra uses only one bit multiplier to multiply an n-bit number to another n-bit number, where n is an unsigned integer. This is in contrast to the traditional method, which requires the maximum required 1-bit multiplier to be n^2 . The region report generated by the integrated design vision tool.

Type of Area	Number of calculations	
	Conventional	Vedic
Combinational	4671	1254.297
Non Combinational	0	176.9472
Total cell area	0	1431.244
Total area	4680	1470.93

Table – II Table that compares the area of conventional and vedic medical

Conclusion:

This paper details the design process for a physical level semiconductor that uses VLSI 90 nm technology to execute Fast Fourier Transform operations. Using both traditional methods and the novel perspective of Vedic mathematics, they painstakingly investigated, contrasted, and analysed the Fast Fourier Transforms. The researchers set out to determine whether there would be any benefits or efficiency gains to using Vedic mathematics in FFT calculations by conducting this extensive analysis. Using Vedic Mathematics as a primary tool, the researchers painstakingly calculated and examined the outcomes relating to gate usage and area requirements in the following stages of their analysis. The results of their thorough investigation were comprehensive numerical tables that provided a solid basis for their subsequent judgements. Careful analysis of the data revealed that there were real advantages to applying FFT based on Vedic mathematics. In particular, it demonstrated a dramatic simplification of computations, especially with regard to timing and speed. This discovery demonstrated how combining current technology with old mathematical ideas might improve computing performance and efficiency.

REFERENCES

- [1] A. R. Prakash, S. Kirubaveni, "Performance evaluation of FFT processor using conventional and Vedic algorithm", IEEE International Conference on Emerging Trends in Computing, Communication and Nanotechnology (ICECCN), Tirunelveli, March 2013, pp. 89-94.
- [2] A. Kumar, A. Raman, "Low Power ALU Design by Ancient Mathematics", IEEE ICAAE, Singapore, Feb. 2010, pp. 862-865.
- [3] M. C. Hanumantharaju, H. Jayalaxmi, R. K. Renuka, M. Ravishankar, "A High Speed Block Convolution Using Ancient Indian Vedic Mathematics", IEEE International Conference on Computational Intelligence and Multimedia Applications, Sivakasi, Tamil Nadu, December 13-15, 2007, pp. 169-173.
- [4] Bansal, Y.; Madhu, C.; Kaur, P. "High speed vedic multiplier designs-A review", Engineering and Computational Sciences (RAECS), 2014 Recent Advances, On page(s): 1 -6.
- [5] Kumar, Anvesh, et al. "Small area reconfigurable FFT design by Vedic Mathematics", Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on. Vol. 5. IEEE, 2010.

- [6] K. Thanushkodi, K. Deena Dayalan, P. Dharani, "A Novel Time and Energy Efficient Cubing Circuit Using Vedic Mathematics for Finite Field Arithmetic," IEEE International Conference on Advances in Recent Technologies in Communication and Computing, Kottayam, Kerala, 27-28 Oct. 2009, pp. 873 – 875.
- [7] H. D. Tiwari, G. Gankhuyag, M. Kim, B. Cho, "Multiplier design based on ancient Indian Vedic Mathematics", IEEE Proc. International SoC Design Conference, ISOCC, Busan, 2008, pp. II-65 - II-68.
- [8] V. Kunchigi, L. Kulkarni, S. Kulkarni.: "High speed and area efficient Vedic multiplier," Proc. IEEE International Conference on Devices, Circuits and Systems (ICDCS), Coimbatore, 2012, pp. 360 – 364.
- [9] J. W. Cooley and J. W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," *Mathematics of Computation*, vol. 19, no. 90, pp. 297-301, 1965.
- [10] C. M. Rader, "Discrete Fourier Transforms when the Number of Data Samples Is Prime," *Proceedings of the IEEE*, vol. 56, no. 6, pp. 1107-1108, 1968.
- [11] S. Masuzawa and S. Kawamura, "Vedic multiplication architecture based on ancient Indian mathematics," *IEEE Transactions on Computers*, vol. 54, no. 7, pp. 839-849, 2005.
- [12]. António M. Grilo, Jaime Chen, Manuel Díaz, Daniel Garrido, and Augusto Casaca, "An Integrated WSA and SCADA System for Monitoring a Critical Infrastructure", *IEEE Transactions on Industrial Informatics*, Vol. 10, No. 3, pp. 1755-1764, August 2014.
- [13]. Debalina Ghosh, Depanwita Debnath, Dr. Amlan Chakrabarti "FPGA Based Implementation of FFT Processor Using Different Architectures", *IJAITI VOLUME 1 NUMBER 1*, PP 24-33, Jan/Feb 2012.
- [14]. Nisha John, Prof. Sadanandan G.K, "FPGA Implementation of a Novel Efficient Vedic FFT/IFFT Processor For OFDM", *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, Vol. 3, Issue 3, September 2014.
- [15]. More T.V.,PanatA.R. "FPGA implementation of FFT using vedic algorithm", *Computational intelligence and Computing Research(ICCIC)*,pp-1-5 , 2013.
- [16]. Harpreet Singh Dhillon, AbhijitMitra, "A Digital Multiplier Architecture using UrdhvaTiryakbhyam Sutra of Vedic Mathematics",*IITG* , pp-1-4 , 2010.
- [17]. AsmitaHaveliya, "FPGA implementation of a Vedic convolution algorithm", *International Journal of Engineering Research and Applications*, Vol. 2, Issue 1, pp.678-684,Jan-Feb 2012.
- [18]. A.Ronisha Prakash, S. Kirubaveni, "Performance Evaluation of FFT Processor Using Conventional and Vedic Algorithm", *IEEE International Conference on Emerging Trends in Computing, Communication and Nanotechnology*, pp-1-6, 2013.
- [19] Duhamel, P., & Vetterli, M. (1990). Fast Fourier transforms: a tutorial review and a state of the art. *Signal processing*, 19(4), 259-299.
- [20] Marks, R. J. (2009). *Handbook of Fourier analysis & its applications*. Oxford University Press.
- [21] Marks, R. J. (2009). *Handbook of Fourier analysis & its applications*. Oxford University Press.
- [22] Beylkin, G. (1995). On the fast Fourier transform of functions with singularities. *Applied and Computational Harmonic Analysis*, 2(4), 363-381.

