Implementation of FFT Algorithm Based on Vedic Maths Using FPGA

[1]Vishal Panchal, [2]Prof. Milind Shah

[1]Student, [2]Associate Professor
Electronics & Communication Department,
Vishwakarma Government Engineering College, Chandkheda, Gujarat,India

***Abstract*—This Fast Fourier transform (FFT) is an efficient algorithm to compute the N point DFT. The FFT is a computationally intensive digital signal processing function, but the Implementation of FFT requires large number of complex multiplications, so to make this process rapid and simple, it is necessary for a multiplier to be fast and power efficient. To solve this problem, UrdhvaTiryagbhyamsutra in Vedic mathematics is used.It is based on a concept through which the generation of all partial products can be done and then, concurrent addition of these partial products can be done. The conventional multiplication method requires more time & area than Vedic algorithms. In this paper, the algorithm for FFT using Vedic maths is proposed.**

***IndexTerms*— FFT, Vedic Mathematics, Vedic Multiplier, UrthvaTiryagbhyam,Vertically and Crosswise Algorithm.**

I. INTRODUCTION

With the advancement of VLSI, Fast Fourier Transform is applied to wide field of digital signal processing and communication system applications. It is mainly used in wireless local area network, digital audio broadcasting, digital video broadcasting-terrestrial and digital video broadcasting-handheld. Due to such diverse application of FFT, it is desirable to develop efficient FFT to meet the requirement of various OFDM communication standards.

The FFT is a faster version of the Discrete Fourier Transform (DFT) and calculates Discrete Fourier Transform efficiently by reducing the computational complexity & number of multipliers required. Since multipliers are very power hungry elements in VLSI designs, they result in significant power consumption. So, the complex multiplication operations are realized using UrdhvaTirvagbhyam. In Ancient Indian vedicmaths, it is an efficient method of multiplication. It literally means “Vertically and crosswise”. This Sutra shows how to handle multiplication of a larger number (N x N, of N bits each) by breaking it into smaller numbers of size (N/2= n) and these smaller numbers can again be broken into smaller numbers (n/2 each) till we reach multiplicand size of (2 x 2). Thus, it simplifythe whole multiplication process. Due to its regular structure, it can be easily layout in a silicon chip. The Multiplier has the advantage that as the number of bits increases, gate delay and area increases very slowly as compared to other multipliers.

In this paper, Vedic algorithm is proposed for the implementation of FFT and multipliers to be used in the FFT. Fast Fourier Transform design methodology using Vedic mathematics algorithm provides a fast and a reliable approach to compute the N point DFT.

II. FFT: FAST FOURIER TRANSFORM

The Cooley–Tukey algorithm, named after J.W. Cooley and John Tukey, is the most common fast Fourier transform (FFT) algorithm. It re-expresses the discrete Fourier transform (DFT) of an arbitrary composite size N = N1*N2 in terms of smaller DFTs of sizes N1 and N2, recursively, to reduce the computation time. The best known use of the Cooley–Tukey algorithm is to divide the transform into two pieces of size N/2 at each step, and is therefore limited to power-of-two sizes, but any factorization can be used in general. These are called the radix-2 and mixed-radix cases, respectively (and other variants such as the split-radix FFT have their own names as well). Here we use radix-2 algorithm.There are two techniques for calculating FFT.

1) Decimation in Time
2) Decimation in Frequency

Here we use decimation in time technique which is shown in figure-1. An N-pointDiscrete Fourier transform (DFT) performs the conversion of time domain data into frequency domain data. The DFT function ofX(k), which is an N-point sequence of x(n), is defined in equation 1.

$$X_k = \sum_{n=0}^{N-1} (x_n) e^{-\frac{2\pi i}{N} nk} \qquad \text{for } 0 \leqslant k \leqslant N\text{-}1 \qquad (1)$$

The more common and simplified notation for the DFT can be seen in Eq. 2,

$$X_k = \sum_{n=0}^{N-1} (x_n) {W_N}^{kn} \qquad (2)$$

W_N represents the twiddle factor or “Nth root of unity” of a complex multiplier.

As seen in figure-1, basic block for calculating FFT is butterfly unit & to implement butterfly unit, multiplier is required. So first of all, multiplier will be implemented based on vedic mathematics. After implementing multiplier, butterfly unit will be developed. By repetition of butterfly unit, FFT can be computed.

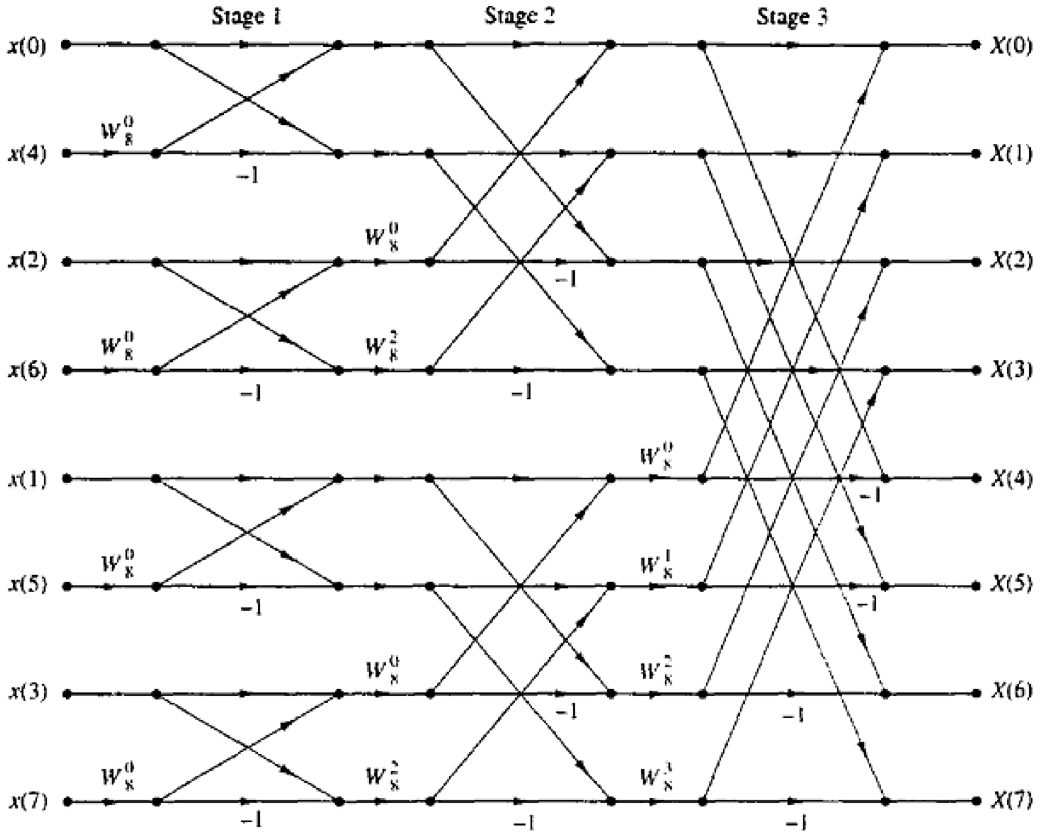


Figure-1 : DIT algorithm for calculating FFT[10]

III. Vedic Mathematics

TheVedic mathematics - a gift given to this world by the ancient sages of Indiais a system which is far simpler and more enjoyable than modern mathematics. The simplicity of Vedic Mathematics means that calculations can be carried out mentally though the methods can also be written down. There are many advantages in using a flexible, mental system. Vedic Mathematics refers to the technique of Calculations based on a set of 16 Sutras, or aphorisms, as algorithms and their upa-sutras or corollaries derived from these Sutras. Any mathematical problems (algebra, arithmetic, geometry or trigonometry) can be solved mentally with these sutras. Vedic Mathematics is more coherent than modern mathematics.

Vedic Mathematics offers a fresh and highly efficient approach to mathematics covering a wide range - starts with elementary multiplication and concludes with a relatively advanced topic, the solution of non-linear partial differential equations. But the Vedic scheme is not simply a collection of rapid methods; it is a system, a unified approach. Vedic Mathematics extensively exploits the properties of numbers in every practical application.

Vedic Mathematics is the ancient system of mathematics which has a unique technique of calculations based on 16 Sutras. Urdhvatiryakbhyam, being a general multiplication formula, is equally applicable to all cases of multiplication. Nikhilam algorithm has the compatibility to handle different data types.

UrdhvaTriyagbhyam Sutra is employed in this project for Multiplication.UrdhvaTiryagbhyam Sutra is a general multiplication formula applicable to all cases of multiplication. It literally means “Vertically and crosswise”.It is based on a novel concept through which the generation of all partial products can be done and then, concurrent addition of these partial products can be done. Thus parallelism in generation of partial products and their summation is obtained using UrdhvaTiryagbhyam sutra.

IV. Architecture of Vedic multiplier:

The multiplier is based on UrdhvaTriyagbhyamsutra. This sutra shows how to handle multiplication of larger number (N X N bits) by breaking it into smaller sizes. For multiplier, first the basic blocks, that 2x2 multiplier is made and then, 4x4 block , 8x8 block have been made. A 4x4 multiplication is simplified into 4, 2x2 multiplication that can be performed in parallel. This reduces the number of partial product and thus reduces the delay of the multiplier. This is parallel implementation style of UrdhvaTriyagbhyam sutra.

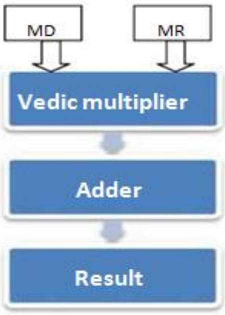


Figure 2: Architecture of Multiplier

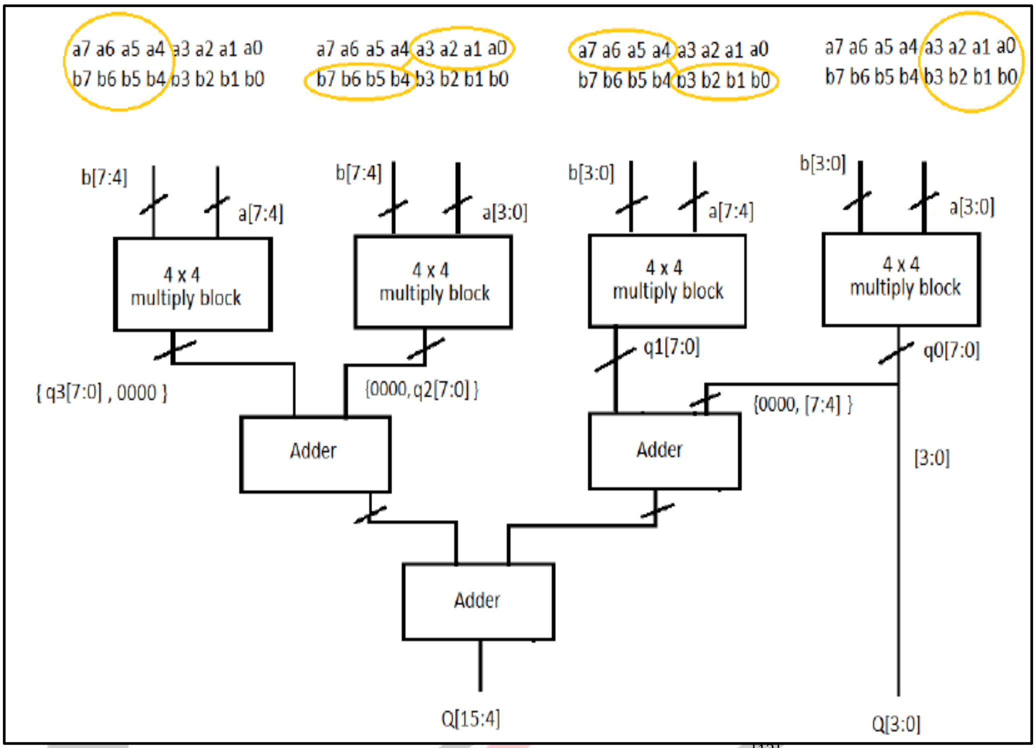


Figure 3: Block diagram of 8 bit Vedic Multiplier[12]

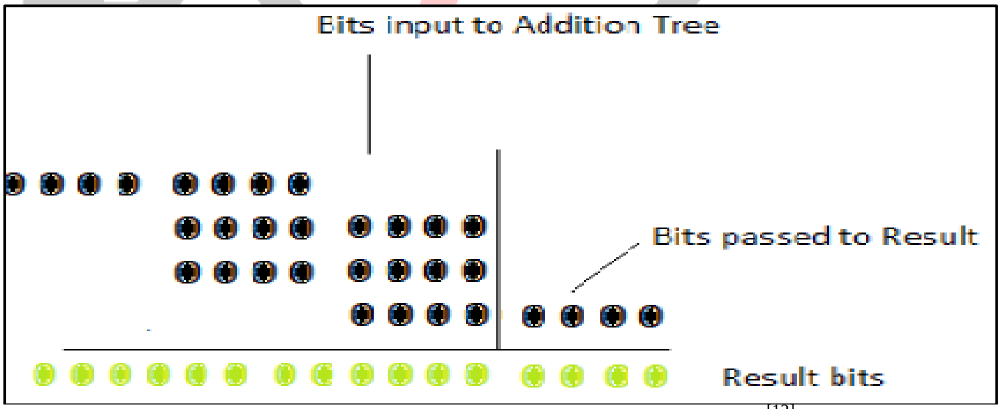


Figure 4: Wallace tree diagram for 8 bit Vedic Multiplier[12]

Using 4 such 4x4 multipliers and 3 adders we can built 8x8 bit multipliers as shown in figure 3. We have to first write code for 8-bit and 12-bit adders. This architecture follows wallace tree which reduces the addition levels from 3 to just 2 stages as shown in figure 4. Last 4 LSB bits are directly taken from qo for the final result.After padding 4 zeros to qo in MSB side, qo& q1 are added using 8-bit adder and result will be stored in q4.For adding q2 and q3, 4 zeros are padded into q2 on left side, and, 4 zeros are padded on right into q3. After this, q2 and q3 are added using 12-bit adder. Result of this addition is stored in q5.Now, to get final result, q4 and q5 are added using 12-bit adder. As both have same alignment, no need to pad any zero. Result is stored in q6.Now, Most significant 12 bits of result is stored in q6 and 4 LSB is in q0.To generate final result, these bits are taken from q6 and q0.Hence, to implement 8x8 vedic multiplier, four 4x4 multipliers, one 8-bit adder and two 12-bit adders are required.

V. IMPLEMENTATION & RESULTS

Figure 5 shows RTL design of FFT implementation and Figure 6 shows waveforms of input and output for different values.

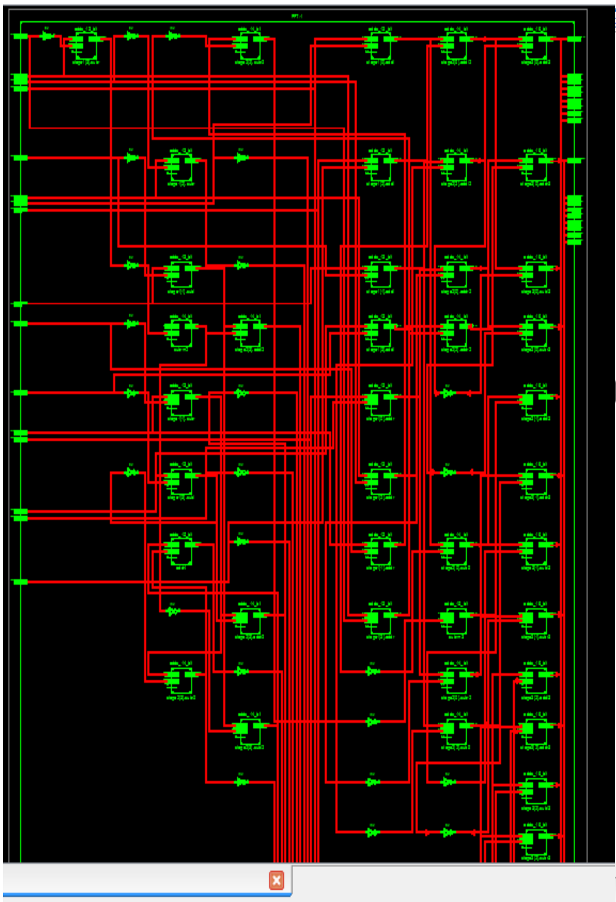
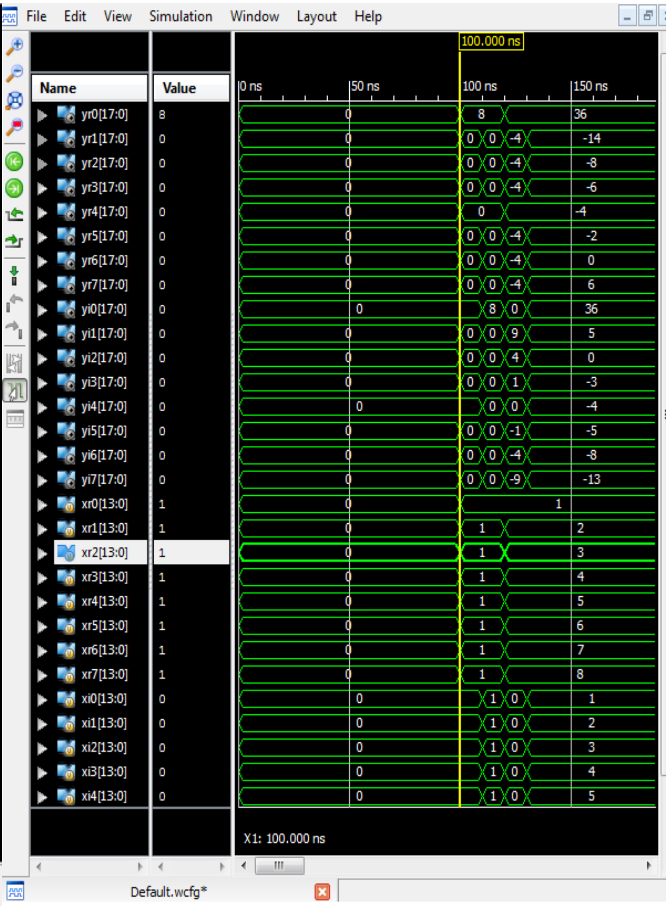


Figure 5: RTL Design of FFT Implementation



Name	Value
yr0[17:0]	8
yr1[17:0]	0
yr2[17:0]	0
yr3[17:0]	0
yr4[17:0]	0
yr5[17:0]	0
yr6[17:0]	0
yr7[17:0]	0
yi0[17:0]	0
yi1[17:0]	0
yi2[17:0]	0
yi3[17:0]	0
yi4[17:0]	0
yi5[17:0]	0
yi6[17:0]	0
yi7[17:0]	0
xr0[13:0]	1
xr1[13:0]	1
xr2[13:0]	1
xr3[13:0]	1
xr4[13:0]	1
xr5[13:0]	1
xr6[13:0]	1
xr7[13:0]	1
xi0[13:0]	0
xi1[13:0]	0
xi2[13:0]	0
xi3[13:0]	0
xi4[13:0]	0

Figure 6: Waveform of input & output

VI. CONCLUSION

It can be concluded that Vedic FFT is better in all respect likespeed, delay, area, complexity, power consumption.Vedic maths decreases number of computation,thus speed increases and hardwarerequired is reduced. So power consumption is also reduced. This alsogives chances for modular design where smaller block can beused to design the bigger one. So the design complexity getsreduced for inputs of large no of bits and modularity getsincreased. General multiplication can be implemented by UrdhvaTiryagbhyam sutraand, in case of multiplication of large numbers,Nikhilam Sutra is used which reduces themultiplication of two large numbers to the multiplication of twosmall numbers. Combine approach of FFT with VedicMathematics create the new advancement in various fields ofengineering.

VII. ACKNOWLEDGMENT

I would like to express my deep and sincere gratitude toProf. Milind Shahfor their constant encouragement, valuable guidance and constructive suggestions.

REFERENCES

[1] Nidhi Mittal, Abhijeet Kumar, “Hardware Implementation of FFT using vertically and Crosswise Algorithm”, International Journal of Computer Applications (0975 – 8887), Volume 35– No.1, December 2011.

[2] Ashish Raman, Anvesh Kumar, R.K.Sarin, “High Speed Reconfigurable FFT Design by Vedic Mathematics”, Journal of computer science and engineering, volume 1, issue 1, may 2010.

[3] Anuj Kumar Varshney, Vrinda Gupta, “Power-Time Efficient Algorithm for Computing Reconfigurable FFT in Wireless Sensor Network”, ISSN : 2229-3345 Vol. 2 No. 3.

[4] Anveshkumar, Ashishraman, Dr.R.K.sarin, Dr.ArunKhosla, “Small area Reconfigurable FFT Design by Vedic Mathematics”, IEEE , volume 5, year 2010.

[5] Anveshkumar, Ashishraman, “Low Power ALU Design by Ancient Mathematics”, IEEE, volume 5, year 2010.

[6] Prof J M Rudagi, Vishwanath Amble, VishwanathMunavalli, RavindraPatil, VinaykumarSajjan, “Design And Implementation Of Efficient Multiplier Using Vedic Mathematics”, IET, Conference on Advances in Recent Technologies in Communication and Computing, year 2011.

[7] S. S. Kerur, PrakashNarchi, Jayashree C N, Harish M Kittur, Girish V A, “Implementation of Vedic Multiplier for Digital Signal Processing”, International Conference on VLSI, Communication & Instrumentation (ICVCI), year 2011.

[8] TusharV.More, AshishR.Panat, “FPGA Implementation of FFT Processor Using Vedic Algorithm”, IEEE International Conference on Computational Intelligence and Computing Research, 2013

[9] DevikaJaina, KabirajSethi, Rutuparna Panda, “Vedic Mathematics Based Multiply Accumulate Unit”, International conference on Computational Intelligence and Communication systems, year 2011.

[10] John G Proakis and Dimitris G. Manolakis, “Digital Signal Processing : Principles, Algorithms, and Applications”, Prentice-Hall, Inc., 1996.

[11] Samir Palnitkar, “Verilog HDL: A Guide to Digital Design and Synthesis”, Sunsoft press, 1996.

[12] http://verilog-code.blogspot.in/2014/01/design-and-implementation-of-16-bit.html