Hardware Implementation of FFT using vertically and Crosswise Algorithm

Nidhi Mittal Abhijeet Kumar
ECE Deptt.ECE Deptt.
M M University, India

ABSTRACT
This paper is devoted for the implementation of FFT, which uses Vertical and Crosswise algorithms. Fast Fourier transform (FFT) is an efficient algorithm to compute the N point DFT. But the Implementation of FFT requires large number of complex multiplications, so to make this process rapid and simple it's necessary for a multiplier to be fast and power efficient. To tackle this problem UrdhvaTirvagbhyam in Vedic mathematics is an efficient method 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. It is one of the sutra of Vedic Mathematics equally applicable to all cases of multiplication. The conventional multiplication method requires more time & area on silicon than Vedic algorithms. In this paper the design for the architecture of FFT using Vertically and Crosswise is proposed and described using Verilog hardware description language. The code description is simulated using ModelSim SE 5.7f and synthesized using ISE Xilinx 9.2i for the FPGA device Spartan XC3S500e-fg320, Speed Grade-4. The results show how by combining these two approaches proposed design methodology is time, area and power efficient.

Keywords–FFT, UrthvaTirvagbhyam,Vertically and Crosswise Algorithm, Vedic Mathematics etc.

1. INTRODUCTION

Direct computation of Discrete Fourier Transform (DFT) requires of the order of N^2 complex multiplication operations, where N is the transform size. The FFT algorithm, started a new era in digital signal processing by reducing the order of complexity of DFT from N^2 to $N\log_2 N$, reduces the number of required complex multiplications compared to a normal DFT [1]. 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 Vedic mathematics 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, say) and these smaller numbers can again be broken into smaller numbers (n/2 each) till we reach multiplicand size of (2 x 2) Thus, simplifying the whole multiplication process. The processing power of this multiplier can easily be increased by increasing the input and output data bus widths since it has a quite regular structure. 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 [2]. Pipeline architecture based on the constant geometry of N point radix-2 FFT algorithm, which uses $N/2\log_2 N$ complex numbermultipliers and is capable of computing a full N-point FFT in N/2 clock cycles[3], has been proposed. Thus there is a need of innovative algorithms to improve the speed.In this paper, Vedic algorithm is proposed for the implementation of multipliers to be used in the FFT. Fast Fourier Transform (FFT) design methodology using Vedic mathematics algorithm provides a fast and a reliable approach to compute the N point DFT**.**This paper is organizedin the following way: Section 1 presents introduction of FFT.Section 2 provides the brief description of the Vedic Mathematics with Urdhva Tiryagbhyam sutra. Section 3 provides an overview on combine approach of Vedic with FFT. Section 4 demonstrates the results from the synthesis tool and the comparative study of the proposed architecture and other existing architecture. Section 5 concludes the paper.

1.2 Fast Fourier Transform (FFT)

An *N*-point Discrete Fourier transform (DFT) performs the conversion of time domain data into frequency domain data. The DFT function of X(k), which is an *N*-point sequence of x(n), is defined in Eq. 1.

$$X(k) = \sum_{n=0}^{N-1} x(n) \exp\{-j\left(\frac{2\pi}{N}\right) kn\},\ 0 \le k \le N-1 \quad (1)$$

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

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

Where $W_N = e^{-j2\pi/N}$, represents the twiddle factor or "Nth root of unity" of a complex multiplier.

The fast Fourier transform (FFT) is able to effectively decompose and compute a DFT by utilizing the symmetry and periodic property of the complex sequence, W_N. The properties are defined in Eq. 3 and 4.

Symmetry Property: $$W_N^{k+\frac{N}{2}} = -W_N^k \quad (3)$$

Periodicity Property: $$W_N^{k+N} = W_N^k \quad (4)$$

Thus reduces the complex number multiplication and addition from N^2 to $N/2\log_2 N$ and $N\log_2 N$ respectively. In FFT, where N is an integer power of 2, i.e. $N=2^L$, the no of stages of computation is L ($=\log_2 N$), then this algorithm is known as radix-2 FFT algorithm. For N=16, which consist of $L = \log_2 16 = 4$, four stages, the first stage computes the eight 2-point DFTs, the second stage computes the four 4-point DFTs, the third stage computes the two 8-point DFTs and finally the fourth stage computes the desired 16 point DFT. The number of complex multiplications are $N/2\log_2 N = 8\log_2 16 = 32$ and the number of complex additions are $N\log_2 N = 16\log_2 16 = 64$. The resulting *N*-point FFT will require *N*/2 complex multiplications per stage by some power of W_N, except for the final 2-point DFT stage where no

multiplication is needed. The radix-2 butterfly computation, seen in Fig. 1, is used to simplify the calculation.

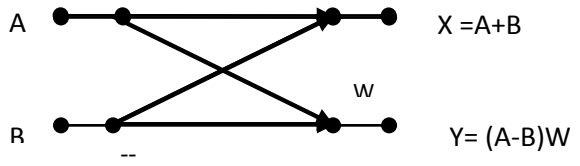

Fig. 1: Radix-2 DIF Butterfly Signal Flow

2. VEDIC MATHEMATICS

Vedic mathematics is an ancient mathematics concept that provides a fast and a reliable approach to perform arithmetic operation using sixteen sutras [5] which was rediscovered from the Vedas between 1911 and 1918 by Sri Bharati Krishna Tirthajicomprised all this work together and gave its mathematical explanation while discussing it for various applications. Swamiji constructed 16 sutras (formulae) and 16 upa sutras (sub formulae) after extensive research in Atharva Veda. Vedic mathematics is not only a mathematical wonder but also it is logical. That's why it has such a degree of eminence which cannot be disapproved. Due to these phenomenal characteristics, Vedic maths has already crossed the boundaries of India and has become an interesting topic of research abroad. One other advantage of Vedic mathematics is that it offers choices. The same calculation can be done by different methods. This way, Vedic mathematics actually helps in holistic development of the brain and children become more creative, inventing their own methods and understanding what they are doing. There is also often a choice about whether to calculate from left to right or from right to left. Vedic mathematics deals with several basic as well as complex mathematical operations. Especially, methods of basic arithmetic are extremely simple and powerful. The word "Vedic" is derived from the word **"Veda"** which means the store-house of all knowledge. The Vedic mathematics approach is totally different and considered very close to the way a human mind works. A large amount of work has been done in understanding various methodologies. The Sutras apply to cover each and every part of mathematics including arithmetic, algebra, geometry, trigonometry, astronomy, calculus etc. The beauty of Vedic mathematics lies in the fact that it reduces the otherwise cumbersome-looking calculations in conventional mathematics to a very simple one. This is so because the Vedic formulae are claimed to be based on the natural principles on which the human mind works. This is a very interesting field and presents some effective algorithms which can be applied to various branches of engineering such as computing and digital signal processing. In fact we can go further and say that if sixteen Sutras cover all of mathematics they must express universal principles and they must in some way form a complete set. With so many advantages, Vedic Mathematics provides with the possibility of solving the same problem in different alternative ways.

2.1 Description of UrdhvaTiryagbhyam Sutra

The multiplier is based on an algorithm UrdhvaTiryagbhyam of ancient Indian Vedic Mathematics. UrdhvaTiryagbhyam Sutra is a general multiplication formula applicable to all cases of multiplication [3]. 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. The Multiplier has the advantage that as the number of bits increases, gate delay and area increases very slowly as compared to other multipliers[9]. Therefore it is time, space and power efficient. 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, say) and these smaller numbers can again be broken into smaller numbers (n/2 each) till we reach multiplicand size of (2 x 2).Thus simplify the whole multiplication process. According to this algorithm, 4*4 bit multiplication can be carried out in the following way: Firstly, least significant bits are multiplied which gives the least significant bit of the product (vertical). Then, the LSB of the multiplicand is multiplied with the next higher bit of the multiplier and added with the product of LSB of multiplier and next higher bit of the multiplicand (crosswise). The sum gives second bit of the product and the carry is added in the output of next stage sum obtained by the crosswise and vertical multiplication and addition of three bits of the two numbers from least significant position. Next, all the four bits are processed with crosswise multiplication and addition to give the sum and carry. The sum is the corresponding bit of the product and the carry is again added to the next stage multiplication and addition of three bits except the LSB. The same operation continues until the multiplication of the two MSBs to give the MSB of the product. To make the methodology more clear, an alternate illustration is given with the help of line diagrams in fig. 2 where the dots represent bit '0' or '1'.

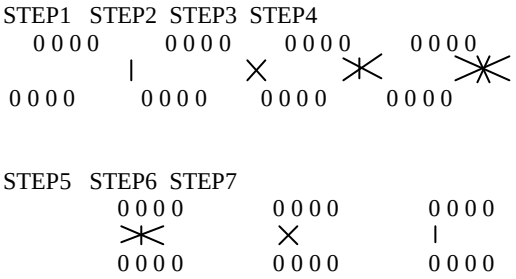

Fig.2: Line diagram for multiplication of two 4-bit numbers [6].

Hence this is the general mathematical formula applicable to all cases of multiplication. The hardware design for this algorithm will be very similar to that of the famous array multiplier where an array of adders is required to arrive at the final product. All the partial products are calculated in parallel and the delay associated is mainly the time taken by the carry to propagate through the adders which form the multiplication array. Clearly, this is not an efficient algorithm for the multiplication of large numbers as a lot of propagation delay is involved in such cases.

3. COMBINE APPROACH OF FFT WITH VEDIC MATHEMATICS

The FFT reduced the complex number multiplications and additions from N^2 to $N/2\log_2 N$ and $N\log_2 N$ respectively. For N=16, the number of complex multiplications are $N/2\log_2 N = 8\log_2 16 = 32$ and the number of complex additions are $N\log_2 N = 16\log_2 16 = 64$. The basic operation of DIT algorithm is the butterfly in which two inputs are combined to give the outputs. When the word length to be 16 bits. The single simple multiplier implementation needs 16 rows of partial product generation and each row containing 16 partial product bits. To accumulate these 16 partial product rows large hardware will be needed to get the result in sum and carry form. As implementation of 16 pointFFT requires large no of multiplication and these all multiplications are done

using Vedic mathematics reduces the time, area and power. The multiplication algorithm is then illustrated to show its computational efficiency by taking an example of reducing a 4x4 bit multiplication to a 2x2-bit multiplication operation. To analyse 4x4 multiplications, say X3X2X1X0 and Y3Y2Y1Y0. Following are the output line for the multiplication result, S7S6S5S4S3S2S1S0. Divide X and Y into two parts, say X3X2 & X1X0 for X and Y3Y2 & Y1Y0 for Y. Using the fundamental of Vedic multiplication, taking two bit at a time and using 2 bit multiplier block, this have the following structure for multiplication.

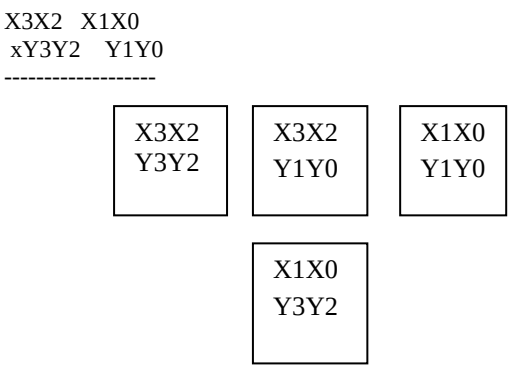

Fig. 3: Block diagram presentation for 4x4 multiplications

Each block as shown above is 2x2 multiplier. First 2x2 multiplier inputs are X1 X0 and Y1 Y0.The last block is 2x2 multiplier with inputs X3 X2 and Y3 Y2. The middle one shows two, 2x2 multiplier with inputs X3X2 & Y1Y0 and X1X0 & Y3Y2. So the final result of multiplication, which is of 8 bit, S7S6S5S4S3S2S1S0, X3X2 and Y3Y2 give multiplication result S33S32S31S30, X3X2 and Y1Y0 give multiplication result S23S22S21S20, X1X0 and Y3Y2 give multiplication result S13S12S11S10, X1X0 and Y1Y0 give multiplication result S03S02S01S00.

For the final result, add the middle product term along with the term shown below.

S33 S32 S31 S30 0 0 S01 S00
S23 S22 S21 S20
S13 S12 S11 S10
0 0 S03 S02

The first two outputs S0 and S1 are same as that of S00 and S01. Result of addition of the middle terms by using two, 4 bit full adders will forms output line from S5S4S3S2. One of the full adder will be used to add (S23 S22 S21 S20) and (S13 S12 S11 S10) and then the second full adder is required to add the result of 1st full adder with (S31 S30 S03 S02). The respective sum bit of the 2nd full adder will be S5S4S3S2. Now the carry generated during 1st full adder operation and that during 2nd full adder operation should be added using half adder so that the final carry and sum to be added with next stage i.e. with S33 S32 to get S7 S6. In the proposed architecture, the 4x4 bit multiplication operation is fragmented FFT modules. The 4x4 multiplication modules are implemented using small 2x2 bit multipliers. The structure of FFT has been designed, optimized and implemented on SPARTAN-3E FPGA (Field Programmable Gate Array). This FFT has the high speed and small area as compared to the conventional FFT. This particular FFT is to be designed by using Vedic multiplier. The delay produced by the Vedic FFT is smaller than the delay produced by the conventional FFT.An FFT circuit has been described that provides the high performance with Small areawhich has great applications incommunication, signal and image processing and instrumentation.

4. IMPLEMENTATION AND RESULTS

Table 1 shows the number of calculations required for various bit lengths. From this table it is clear that how the Vedic mathematics going to reduce the number of adder and multiplier as compare to the conventional method. As we are reducing the number of adder in each bit the delay provided by the particular circuit is also going to reduced hence which causes to increases the speed of FFT.

Table 1 Comparison between conventional Array

Input Bit Length	Numbers of calculations			
	Conventional		Vedic	
	M	A	M	A
2	4	2	4	1
3	9	7	9	5
4	16	15	16	9
8	64	77	64	53

Multiplier and Vedic Multiplier

M: Numbers of multiplication,A: Numbers of additions

Simulation and synthesis results show the comparison between conventional FFT and Vedic FFT in terms of delay, power and area. This is shown by bar-graph. The following bar-graph shows that there is some reduction in delay, power and area which means overall performance of Vedic FFT is better than the conventional FFT.

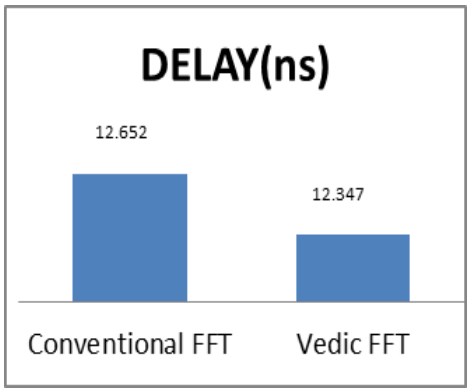

Fig. 4: Comparison by bar-graph for delay

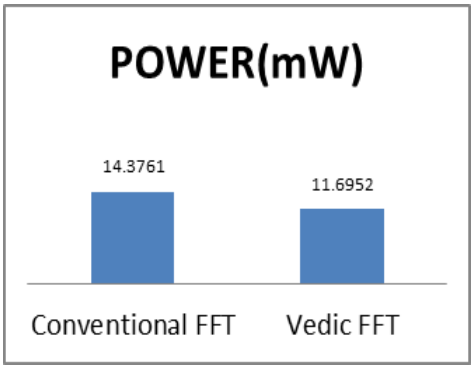

Fig. 5: Comparison by bar-graph for power

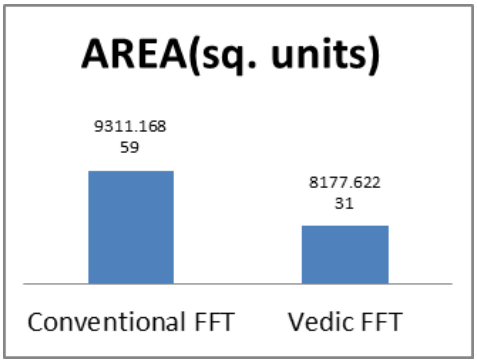

Fig. 6: Comparison by bar-graph for area

5. CONCLUSION

It can be concluded that Vedic FFT is superior in all respect like speed, delay, area, complexity, power consumption. Ancient Indian Vedic Mathematics gives efficient algorithms or formulae for multiplication which increase the speed of devices. This also gives chances for modular design where smaller block can be used to design the bigger one. So the design complexity gets reduced for inputs of large no of bits and modularity gets increased. UrdhvaTiryagbhyam, is general mathematical formula and equally applicable to all cases of multiplication. Also, the architecture based on this sutra is seen to be similar to the popular array multiplier where an array of adders is required to arrive at the final product. Due to its structure, it suffers from a high carry propagation delay in case of multiplication of large number. This problem can solve by Nikhilam Sutra which reduces the multiplication of two large numbers to the multiplication of two small numbers. The application of FFT algorithm includes linear filtering, Correlation, Spectrum Analysis which will further adds the field of Communication, signal & image processing and instrumentationthat can also benefit future needs of wireless communications systems. Combine approach of FFT with Vedic Mathematics create the new advancement in various fields of engineering.

6. ACKNOWLEDGMENTS

The authors would like to thank International Journal of Computer Applications (IJCA) for supporting this research.The authors would like to thank their colleagues and their family members for many fruitful discussions.

7. REFERENCES

[1] Ashish Raman, Anvesh Kumar, R.K.Sarin, ‘‘High Speed Reconfigurable FFT Design by Vedic Mathematics’’, journal of Computer Science and Engineering, vol.1, pp 59-63 May 2010.

[2] Anvesh Kumar, Ashish Raman, ‘‘Small Area Reconfigurable FFT Design by Vedic Mathematics”, vol 5, IEEE pp 836-838, 2010.

[3] Laxman P. Thakre, Suresh Balpande,UmaehAkare, SudhairLande, ‘‘Performance evaluation and Synthesis of Multiplier Used in FFT Operation Using conventional and Vedic Algorithm”, International Conference on emerging trends in Engineering and Technology, pp 614-619, 2010.

[4] M.E.Paramasivam, Dr.R.S.Sabeenian, ‘‘An Efficient Bit Reduction Binary Multiplication Algorithm using Vedic Methods”, IEEE pp 25-28, 2010.

[5] Anvesh Kumar, Ashish Raman, ‘‘Low Power ALU Design by Ancient Mathematics”, vol 5, IEEE pp 862-865, 2010.

[6] SumitVaidya, Deepak Dandekar, ‘‘Delay-Power Performance Comparison of Multipliers in VLSI Circuit Design”, International Journal of Computer Networks & Communications (IJCNC), Vol.2, No.4, pp 47-56 July 2010.

[7] Leonard Gibson Moses S, Thilagar M, ‘‘VLSI Implementation of High Speed DSP algorithms using Vedic Mathematics’’, International Journal of Computer Communication and Information System, Vol.2. pp 119-122 Jul –Dec 2010.

[8] Parth Mehta, DhanashriGawelli, “Conventional Versus Vedic Mathematical method for Hardware Implementation of a multiplier”, International Conference on emerging trends in Engineering and Technology, pp 640-642, 2009.

[9] M.Ramalatha, K.DeenaDayalan, S. Deborah Priya, P.Dharani, ‘‘High Speed Energy Efficient ALU Design using Vedic Multiplication Techniques”, ACTEA IEEE ZoukMosbeh, Lebanon, pp 600-603 July 15-17, 2009.

[10] Ramalatha M, Deena Dayalan, Thanushkodi K, Dharani P,‘‘A Novel Time and Energy Efficient Cubing Circuit using Vedic Mathematics for Finite Field Arithmetic”, International Conference on Advances in Recent Technologies in Communication and Computing , pp 873-875, 2009.

[11] Harpreet Singh Dhillon and AbhijitMitra, ‘‘A Reduced Bit Multiplication Algorithm for Digital Arithmetic’s”, International Journal of Computational and Mathematical Sciences Spring, 2008.

[12] Anthonyo Brien, Richard Conway, ‘‘Lifting Scheme Discrete Wavelet Transform using Vertical and Crosswise Multipliers”, ISSC, Galway June 18-19 pp 331-336, 2008.

[13] Honey DurgaTiwari, GanzorigGankhuyag, Chan Mo Kim, Yong Beom Cho, ‘‘Multiplier design based on ancient Indian Vedic Mathematics,” International SoC Design Conference, pp 65-68, 2008.

[14] Hanumantharaju M.C, Jayalaxmi H, Renuka R. K, Ravishankar M,‘‘A High Speed Block Convolution using Ancient Indian Vedic Mathematics,” International Conference on Computational Intelligence and Multimedia Application , pp 169-173, 2007.

[15] ShamainAkhter, ‘‘VHDL Implementation of Fast NxN Multiplier based on Vedic Mathematics”, Jaypee Institute of Information Technology University, Noida, 201307op, India, IEEE 2007.

[16]Dr. K. S.Gurumurthy, M. S. Prahalad, ‘‘Fast and Power Efficient 16x16 Array of Multiplier Using Vedic Multiplication”, International Conference on Computational Intelligence and Multimedia Application, 2006.

[17] Abhijeet Kumar, Dilip Kumar Siddhi, “Hardware Implementation of 16*16 bit Multiplier and Square using Vedic Mathematics”, Design Engineer, CDAC, Mohali.