

Implementation of Fast Fourier Transform and Vedic Algorithm for Image Enhancement Using Matlab

S. Arunachalam

Department of Mathematics
Rizvi College of Arts, Science and Commerce
University of Mumbai, Mumbai, India and
North Maharashtra University, Jalgaon, India

S. M. Khairnar

Department of Mathematics
MIT Academy of Engineering
University of Pune, Pune, India

B. S. Desale

Department of Mathematics
University of Mumbai, Mumbai, India

Copyright © 2014 S. Arunachalam, S. M. Khairnar and B. S. Desale. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

The basic problems in digital image processing are image signal representation and modelling, enhancement, restoration, reconstructions from projections, analysis and coding. Image enhancement includes method for contrast enhancement, edge detection, sharpening, linear filtering and non-linear filtering, zooming and noise removal. In this paper, we have implemented two-dimensional Fast Fourier Transform (FFT) and Vedic algorithm based on Urdhva Tiryakbhyam sutra. The algorithm is presented using MATLAB program. The input image is divided into blocks and two-dimensional FFT is applied to enhance

or filter the image. The proposed two-dimensional FFT design is based on using Urdhva Tiryakbhyam sutra. FFT computations using Vedic multiplication sutra gives a significant performance as compared to the conventional FFT.

Keywords: Fast Fourier Transform, Discrete Fourier Transform, Vedic Algorithm, Vedic Multiplier, Image Enhancement, Linear Filtering, Urdhva Tiryakbyham Sutra

1. Introduction

Image enhancement algorithms are used to emphasize specific image features to improve the quality of the image for visual perception or to aid in the analysis of image for feature extraction. An image is the two-dimensional signal whose intensity at any point is a function of two spatial variables. There has been research in the area of Very Large Scale Integration (VLSI) implementation, Block convolution, Field Programmable Gate Array (FPGA) implementation, Digital signal processing using Vedic algorithms. In [12], FPGA implementation of overlap-add algorithm for high speed block convolution using Vedic multiplication was presented. This method is very efficient in terms area and speed as compared to conventional architectures. In [20], using Vedic multiplication architecture new efficient method was presented for decomposing a perfect binary multiplication into smaller size which reduces the computation time and power consumption. In [13, 14], efficient method of elliptic curve encryption was introduced using ancient Indian Vedic mathematics. In [1], low power Arithmetic Logic Unit (ALU) design using Vedic multiplier was presented which has high speed, low power and consuming little area. In [23], multiplier design based on Vedic mathematics was proposed. The FPGA implementation result shows the delay and the area required in proposed design is far less than the conventional, booth and array multiplier designs. In [2], small area reconfigurable FFT design by Vedic mathematics was presented. The FFT design circuit provides the high performance, dynamic range and low power functionality. In [17], VLSI implementation of Ronald Rivest - Shamir Adi - Adleman Lonard (RSA) algorithm using Vedic mathematics was proposed. The RSA circuit implemented with Vedic algorithm overlay high speed multiplier algorithm exhibits improved efficiency in terms of speed. In [9], the implementation of Vedic algorithm in digital signal processing was presented. The algorithm implemented in 8085/8086 microprocessors clearly indicates the computational advantage as compared to the conventional methods. In [22], high speed energy efficient ALU design using Vedic algorithm was introduced. An efficient Vedic multiplier which has high speed, less complexity and consuming less area was designed by using Multiply Accumulate (MAC) unit. In [21], high speed Vedic multiplier for digital signal proce-

processors was proposed. The Vedic multiplier proves to be highly efficient in terms of speed. In [8], FPGA architecture for two-dimensional Discrete Fourier Transform (DFT) for large input sizes based on a novel two-dimensional decomposition algorithm was implemented. The evaluation of this architecture shows significant performance enhancements over existing two-dimensional DFT applications. In [18], the simulation of Vedic multiplier in two-dimensional DCT was implemented. The proposed two-dimensional DCT using Vedic algorithm and simulation with MATLAB shows significant performance as compared to the conventional DCT.

This paper proposes the implementation of two-dimensional FFT and Vedic algorithm using MATLAB. This paper is organized as follows. Section 2 describes Two-Dimensional DFT. Section 3 describes Two-Dimensional FFT. Section 4 focuses on Vedic mathematics. Section 5 describes Comparison between Vedic and conventional multiplier. Section 6 discusses Vedic multiplier and two-dimensional FFT. Section 7 shows the Two-Dimensional FFT results and discussion and finally the conclusion is given in Section 8.

2. Two-Dimensional Discrete Fourier Transform (DFT)

Let $f(x, y)$ for $x = 0, 1, 2, \dots, M - 1$ and $y = 0, 1, 2, \dots, N - 1$ denote a digital image of size $M \times N$ pixels. The two-dimensional DFT of $f(x, y)$ denoted by $F(u, v)$ and is given by the equation

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)} \quad (1)$$

for $u = 0, 1, 2, 3, \dots, M - 1$ and $v = 0, 1, 2, 3, \dots, N - 1$. We can expand the exponential into sine and cosine function with the variables u and v determining their frequencies x and y . The frequency domain is the coordinate system spanned by $F(u, v)$ with u and v as frequency variables. The $M \times N$ rectangular region defined by $u = 0, 1, 2, 3, \dots, M - 1$ and $v = 0, 1, 2, 3, \dots, N - 1$ is often referred to as the frequency rectangle [11]. Clearly, the frequency rectangle is of the same size as the input image. From the two-dimensional DFT Equation (1), we can show that the transformation property of DFT is given as

$$F(u - u_0, v - v_0) = f(x, y) e^{-j2\pi\left(\frac{u_0 x}{M} + \frac{v_0 y}{N}\right)}$$

when $u_0 = M/2$ and $v_0 = N/2$, we obtain

$$\begin{aligned} F\left(u - \left(\frac{M}{2}\right), v - \left(\frac{N}{2}\right)\right) &= f(x, y) e^{-j2\pi\left(\frac{x}{2} + \frac{y}{2}\right)} \\ &= f(x, y) (-1)^{(x+y)} \end{aligned} \quad (2)$$

Equation (2) shows that the Fourier spectrum can be shifted to the central point for better displaying by multiplying the original image by $(-1)^{(x+y)}$. The inverse DFT is given by the equation

$$f(x, y) = \left(\frac{1}{MN}\right) \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)} \quad (3)$$

for $x = 0, 1, 2, 3, \dots, M - 1$ and $y = 0, 1, 2, 3, \dots, N - 1$. Thus, given $F(u, v)$, we can obtain $f(x, y)$ back by means of inverse DFT. The value of $F(u, v)$ in Equation (3) is referred as the Fourier coefficients of the expansion. In some formulations of the DFT, the $(1/MN)$ term appears in front of the transform and in others it is used in front of inverse. The only difference between the direct and inverse Fourier transformations is the sign of exponent [16]. The principle for analyzing a transform visually is to compute spectrum. That is the magnitude of $F(u, v)$ which is a real function and display it as an image [10]. Let $R(u, v)$ and $I(u, v)$ be the two functions represent the real and imaginary components of $F(u, v)$. Then the Fourier spectrum is defined as

$$|F(u, v)| = [R^2(u, v) + I^2(u, v)]^{1/2} \quad (4)$$

The phase angle of the transform is defined as

$$\varphi(u, v) = \arctan \left[\frac{I(u, v)}{R(u, v)} \right] \quad (5)$$

3. Two-Dimensional Fast Fourier Transform (FFT)

For efficient computation of DFT several algorithms have been developed based on divide and conquer methods. The algorithm developed by J.W.Cooley and J.W.Tukey [6, 7] in 1965 is the most efficient one and is called FFT. The FFT algorithm is used as an efficient means to compute the DFT and inverse DFT. The

basic principle of FFT algorithm is to decompose DFT into successively smaller DFTs. Fast algorithms for computing the DFT commonly called the FFT algorithms is to decompose successively the N-point DFT computation into computations of smaller size DFTs and to take advantage of the periodicity and symmetry properties of the complex number W_N^{kn} . Such decompositions, if properly carried out, can result in a significant savings in the computational complexity given by the total number of multiplications and the total number of additions needed to compute all N-point DFT samples [3, 5]. In direct computation of N-point DFT, the total number of complex additions are $N(N-1)$ and the total number of complex multiplications are N^2 . In FFT, the total number of complex additions are reduced to $N(\log_2 N)$ and total number of complex multiplications are $(N/2)(\log_2 N)$.

4. Vedic Mathematics

The word 'Vedic' is derived from 'Veda' which means the store-house of all knowledge. Vedic mathematics was reconstructed from ancient Indian Vedas by Sri.Bharati Krsna Tirthaji [15] (1884-1960). Vedic mathematics is the name given to the ancient Indian system of mathematics based on simple rules and principles using which many mathematical problems can be solved [19]. Vedic formulae are claimed to be based on the natural principles on which the human mind works. This is very interesting field and presents some effective algorithms which can be applied to various branches of engineering such as computing, digital signal processing and digital image processing. Vedic mathematics is mainly based on sixteen sutras. In this paper, the proposed multiplier is based on Urdhva Tiryakbhyam Sutra. This is the general formula applicable to all cases of multiplication. It literally means "vertically and cross-wise". It is based on a novel concept through which the generation of all partial products can be done with the concurrent addition of these partial products. The parallelism in generation of partial products and their summation is obtained by using Urdhva Tiryakbhyam sutra for 4×4 bit multiplication. This algorithm can be generalised for the multiplication of a larger number $N \times N$. Multiplication is a fundamental operation in most signal processing applications. Fast multipliers are essential part of digital signal processing systems. The speed of multiplication operation is great importance in digital signal processing and image processing. The proposed Vedic multiplier significantly improves the computational speed involved in multiplication operations of the image processing [4].

5. Comparison between Vedic and Conventional Multiplier

In binary system, only 0 and 1 are used hence in multiplication in Urdhva Tiryakbhyam formula is replaced by AND logic. Each AND will be a bit wide and these bits are added together to generate cross product. Rules for vertically crosswise multiplication remains the same as starting from most significant bit (MSB) of both multiplicands considered for cross product. The increasing one bit in each further calculation with cross product taken for bits of multiplicands till all bits used. Further dropping bits from MSB process of cross product is continued till only least significant bit (LSB) is used for cross product. In this system, the maximum width of cross product depends on width of multiplicands. For instance, in 8-bit multiplication maximum cross product width will be $\log_2 8 + 1 = 4$. In 16-bit, it will be 5.

Conventional multiplier of width $N \times N$ will generate N number of partial products with each product containing N bits with 0 to 7 zeros added to the end. The partial products are generated by bitwise AND one multiplier with another multiplier. Hence the $N \times N$ bit multiplication uses $2N$ multiplication and N adders in the architecture of conventional multiplier. Vedic multiplier generates $2N - 1$ cross products of different widths which when combined forms $(\log_2 N + 1)$ partial products for N -bit multiplier. In case of number of partial products there is significant decrease in number for Vedic multiplier. But partial products generated in case of conventional multipliers are just by AND one multiplier by digits of another multiplier, whereas in case of Vedic, partial products are obtained after cross products are generated which requires some logic. Hence in Vedic multiplier delay in partial products is equal to adder delay. Critical path would consist of adders adding maximum number of bits in cross product. In all cases it will be the cross product in which all bits of multipliers will be considered. Different techniques are used to combine these partial products efficiently to reduce the total time required for multiplication. The multiplications and additions are significantly reduced in case of Vedic multiplier as compared to conventional multiplier. Table 1 gives the detail about the multiplication and addition in case of Vedic and conventional multiplier for 8×8 bit, 16×16 bit and 32×32 bit.

6. Vedic Multiplier and Two-Dimensional FFT

Transform coding algorithms start by partitioning the original image into blocks of small size, usually 8×8 block size. For each block the transform coefficients are calculated effectively by converting the original 8×8 array of pixel values into an

array of coefficients. MATLAB program is used for the computation of image pixels. The image is segmented into 8×8 block size. The time domain coefficients are converted into frequency domain coefficients. The computation of two-dimensional FFT is performed and its functional verification is carried out. The output of 8×8 block produces 64 FFT coefficients which is stored in a matrix. The inverse FFT is performed to recreate the input image. The design flow chart of Vedic multiplier is as shown in Figure 1. The Vedic multiplier is implemented along with the two-dimensional FFT. In practical applications, the Vedic multiplier must be implemented in hardware along with FFT for best efficiency. The multiplication used in the FFT algorithm is carried out using the Vedic multiplier instead of conventional multiplier as shown in flow chart in Figure 1. The design flow chart of two-dimensional FFT is as shown in Figure 2. The flow chart shows the computation of two-dimensional FFT of 8×8 blocks of input image.

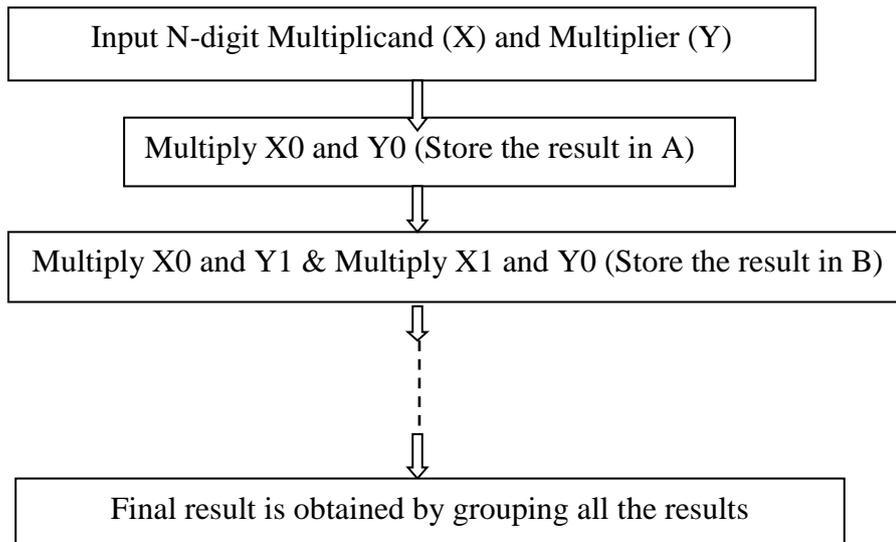
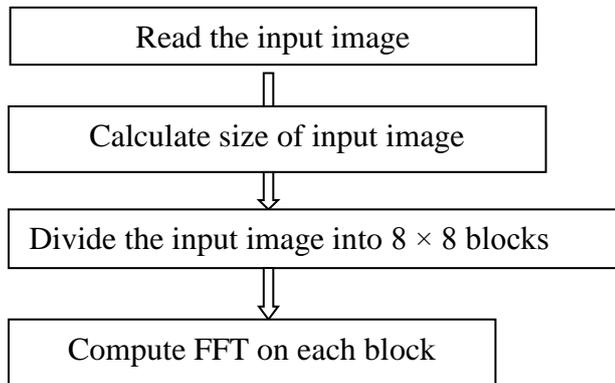


Figure 1: Flow Chart of Vedic Multiplier



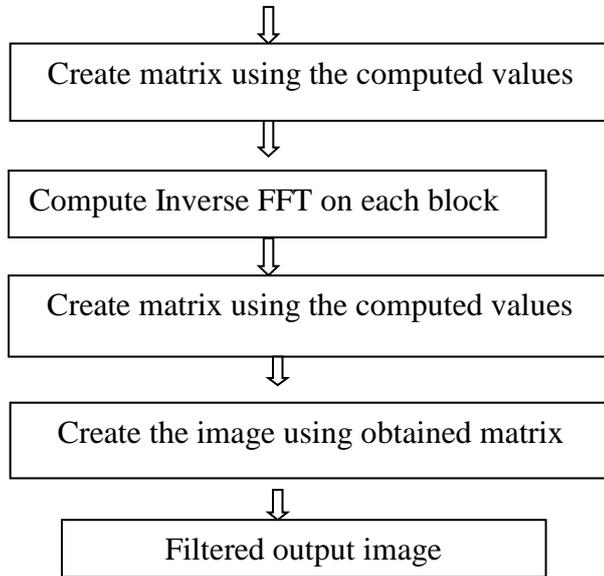


Figure 2: Flow Chart of two-dimensional FFT

7. Two-Dimensional FFT Results and Discussion

The number of additions and multiplications using Vedic multiplier and conventional multiplier is given in Table 1.

Table 1: Conventional and Vedic multipliers of different width

Bit length	Conventional		Vedic	
	Multiplication	Addition	Multiplication	Addition
8	64	56	8	4
16	256	240	16	8
32	1024	992	32	16

For two-dimensional FFT algorithm: Number of additions = $N (\log_2 N)$

Number of Multiplications = $(N/2) (\log_2 N)$

Table 2: Number of additions and multiplications for two-dimensional FFT

Bit Length	Two-Dimensional FFT	
	Multiplication	Addition
8	12	24
16	32	64
32	80	160

Table 3: Number of Additions and Multiplications for two-dimensional FFT with conventional and Vedic multiplier

Bit Length	FFT with conventional Multiplier		FFT with Vedic Multiplier	
	Multiplication	Addition	Multiplication	Addition
8	768	1344	96	96
16	8192	15360	512	512
32	81920	158720	2560	2560

From Table 3, we observe that the Vedic multiplier has a great advantage over the conventional multiplier when used with two-dimensional FFT for different block sizes. In the JPEG image, the input image is divided into 8×8 or 16×16 blocks, and the two-dimensional FFT is computed for each block and then putting the blocks back together into a single image. Equation (1) is used to calculate the FFT matrix. Then compute inverse FFT on each block and create the matrix using the computed values. The image is created using the obtained matrix. The input and output images of two-dimensional FFT process is shown below as Figures 3(a), 3(b), 3(c) and 3(d) and Figures 4(a), 4(b) 4(c) and 4(d). From the original gray scale input image Figure 3(a), we obtained the 'enhanced image' or 'Filtered image' shown in Figure 4(b) by implementing two-dimensional FFT with Vedic algorithm using MATLAB. As shown in Figure 2, first reading the input image and dividing the image matrix into the smaller matrices of size 8×8 . Implement the two-dimensional FFT and the matrix of size 8×8 is transformed using FFT. The multiplication used in the FFT algorithm is carried out using the Vedic multiplier instead of conventional Arithmetic Logic Unit (ALU) multiplier. The original input gray scale image is newspaper print of a car. Performing two-dimensional FFT, we obtain the real and imaginary parts of spectrum, log magnitude of spectrum, convolution matrix or kernel, phase spectrum of filtered image and filtered image.

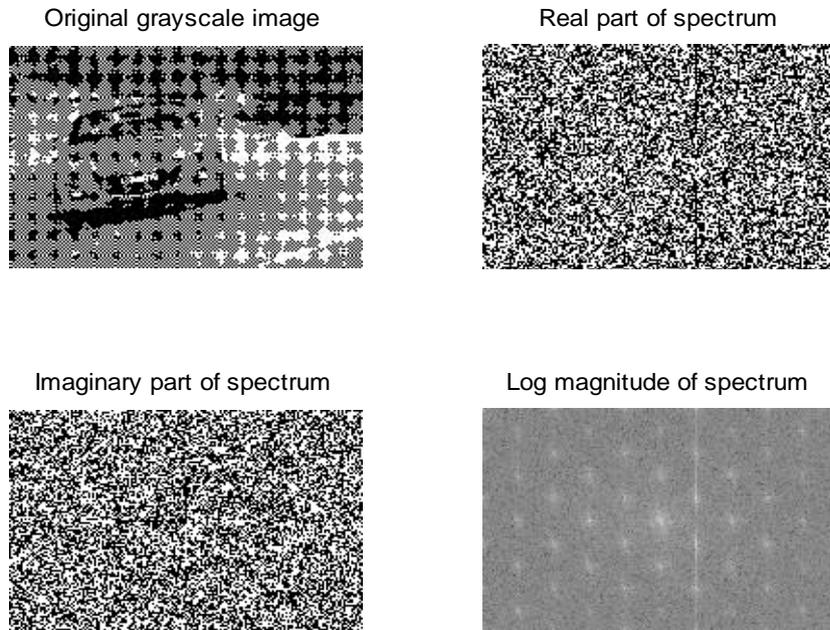


Figure 3 (a) Original gray scale image. (b) Real part of spectrum. (c) Imaginary part of spectrum. (d) Log magnitude of spectrum.

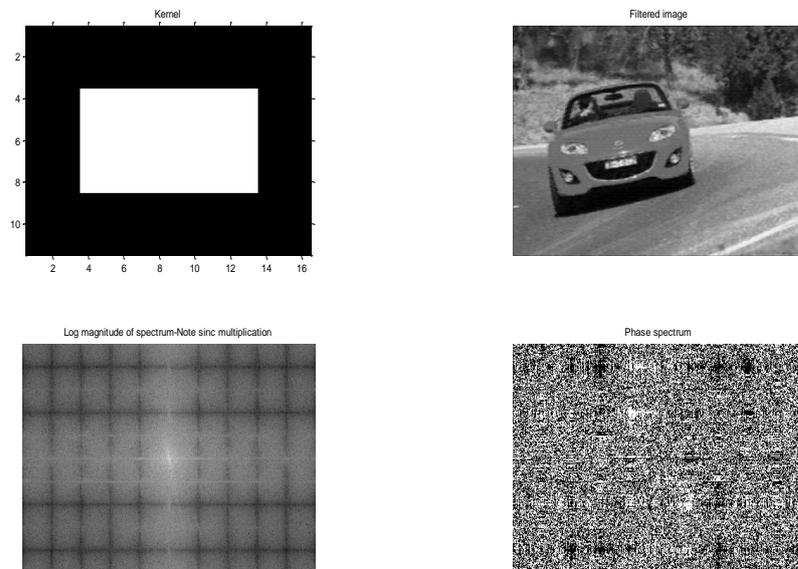


Figure 4 (a) Kernel of image. (b) Filtered Image. (c) Log magnitude of spectrum using sinc multiplication. (d) Phase spectrum of filtered image.

8. Conclusion

The frequency domain approach is an ideal tool for experimenting quickly and with full control over filter parameters. We have implemented two-dimensional FFT with Vedic algorithm using MATLAB program. We have taken an input original gray scale image of 'Newspaper print of a car' and obtained the enhanced or filtered image of Real part of the spectrum, Imaginary part of the spectrum, Log magnitude of spectrum, Kernel or Convolution matrix of an image, Phase spectrum of filtered image and an output of Filtered image. One can observe that the difference between the quality of input original gray scale image and the enhanced or filtered image visually as shown in Figure 3(a) and Figure 4(b). The proposed two-dimensional FFT using Vedic multiplication sutra called 'Urdhva Tiryakbyham' with MATLAB shows significant performance as compared to the conventional FFT.

References

- [1] Anvesh Kumar and Ashish Raman, "Low Power ALU Design by Ancient Mathematics", IEEE 2010, Vol.5, pp: 862-865.
<http://dx.doi.org/10.1109/iccae.2010.5451892>
- [2] Anvesh Kumar, Ashish Raman, Sarin R. K., and Arun Khosla., "Small Area Reconfigurable FFT Design by Vedic Mathematics", IEEE, Vol.5, 2010, pp: 836-838. <http://dx.doi.org/10.1109/iccae.2010.5451884>
- [3] Arunachalam S., Khairnar S. M., and Desale B. S., "The Fast Fourier Transform Algorithm and its Application in Digital Image Processing", The International Institute for Science, Technology and Education (IISTE) Journal of Mathematical Theory and Modeling (ISSN 2225-0522), Vol.3, No.6, June 2013, pp: 267-273.
- [4] Arunachalam S., Khairnar S. M., and Desale B. S., "Implementation of Fast Fourier Transform (FFT) using Vedic Algorithm", International Conference on Information and Mathematical Sciences Proceedings Published by Elsevier Science and Technology (ISBN 978-93-5107-162-4), October 2013, pp: 282-285.
- [5] Arunachalam S., Khairnar S. M., and Desale B. S., "Application of Fast Fourier Transform (FFT) Algorithm in Finite Impulse Response(FIR)

Linear Filtering using MATLAB”, International Journal of Engineering Research and Technology (ISSN 2278-0181) Vol.2, Issue 11, November 2013, pp: 3534-3542.

[6] Cooley J. W. and Tukey J. W., “An Algorithm for Machine Calculation of Complex Fourier Series”, Math, Computation, 19, April 1965, pp: 297-301.
<http://dx.doi.org/10.1090/s0025-5718-1965-0178586-1>

[7] Cooley J. W, Peter A. W. Lewis and Peter D. Welch., “The Fast Fourier Transform and its Applications”, IEEE Transactions on Education, Vol.12, No.1, March 1969, pp: 27-34. <http://dx.doi.org/10.1109/te.1969.4320436>

[8] Chi-Liyu., Jung-Sub Kim., et.al., “FPGA Architecture for 2D Discrete Fourier Transform Based on 2D Decomposition for Large Sized Data”, J. Sign Process Syst., 2010, pp: 21-34.

[9] Chidgupkar P. D., and Mangesh T. Karad., “The Implementation of Vedic Algorithm in Digital Signal Processing”, International Journal of Education, Australia, Vol.8, No.2, 2004, pp: 153-158.

[10] Gonzalez R. C., Woods R. E., and Eddins S. L., “Digital Image Processing using MATLAB”, Second Edition, Tata McGraw Hill, New Delhi, 2010.

[11] Gonzalez R. C., and Woods R. E., “Digital Image Processing”, Third Edition, Pearson Education, Inc., 2009.
<http://dx.doi.org/10.1117/1.3115362>

[12] Hanumantharaju M. C., Jayalaxmi H., Renuka R. K., and Ravishankar M., “A High Speed Block Convolution using Ancient Indian Vedic Mathematics”, IEEE International Conference on Computational Intelligence and Multimedia Application, 2007, pp:169-173.
<http://dx.doi.org/10.1109/iccima.2007.332>

[13] Himanshu Thapliyal, Srinivas M. B., “An Efficient Method of Elliptic Curve Encryption Using Ancient Indian Vedic Mathematics”, Proc. IEEE MIDWEST Symp. Circuits and Systems, Cincinnati, August 2005, pp. 826-829.
<http://dx.doi.org/10.1109/mwscas.2005.1594228>

[14] Himanshu Thapliyal and Srinivas M.B., “High Speed Efficient N by N Bit Parallel Hierarchical Overlay Multiplier Architecture Based on Ancient Indian Vedic Mathematics”, *Enformatica Trans*, Vol.2, December 2004, pp. 225 – 228.

[15] Jagadguru Swami Sri. Bharati Krishna Tirthaji Maharaja, “Vedic Mathematics”, Motilal Banarasidass Publishers Private Limited, Delhi, 1992.

[16] Kenneth R. Castleman., “Digital Image Processing”, Pearson Education, Inc., 1996.

[17] Kumaravel S., and Marimuthu R., “VLSI Implementation of High Performance RSA Algorithm using Vedic Mathematics” *IEEE, ICCIMA-2007*, pp: 126-128.
<http://dx.doi.org/10.1109/iccima.2007.238>

[18] Kunchigi V., Kulkarni L., and Subhash Kulkarni., “Simulation of Vedic Multiplier in DCT Applications”, *International Journal of Computer Applications*, Vol.63, No.16, February 2013, pp. 27-32.
<http://dx.doi.org/10.5120/10552-5744>

[19] Nicholas A. P., Williams K. R., and Pickles J., “Application of Urdhva Sutra”, *Spiritual Study Group, Roorkee (India)*, 1984.

[20] Paramasivam M. E., and Sabeenian R. S., “An Efficient Bit Reduction Binary Multiplication Algorithm using Vedic Methods”, *IEEE 2nd International Advance Computing Conference*, 2010, pp: 25-28.
<http://dx.doi.org/10.1109/iadcc.2010.5423043>

[21] Pushpangadan R., Sukumaran V., et.al., “High Speed Vedic Multiplier for Digital Signal Processors”, *IETE Journal of Research*, Vol.55, Issue 6, December 2009, pp: 282-286. <http://dx.doi.org/10.4103/0377-2063.59167>

[22] Ramalatha M., Deena Dayalan K., Dharani P., and Priya S. D., “High Speed Energy Efficient ALU using Vedic Multiplication Techniques”, *IEEE, ACTEA 2009*, pp: 600-603.
<http://dx.doi.org/10.1109/actea.2009.5227842>

[23] Tiwari H. D., Ganzorig Gankuyag., Chan Mo Kim and Young Beom Cho., “Multiplier Design Based on Ancient Indian Vedic Mathematics”, IEEE, International SoC Conference 2008, pp: 65-68.
<http://dx.doi.org/10.1109/socdc.2008.4815685>

Received: December 15, 2014; Published: March 21, 2015