International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:10, No:2, 2016

An Efficient Implementation of High Speed Vedic Multiplier Using Compressors for Image Processing Applications

Shobha Sharma, Amita Dev, Akanksha Kant

***Abstract*—**Digital signal processor, image signal processor and FIR filters have multipliers as an important part of their design. On the basis of Vedic mathematics, Vedic multipliers have come out to be very fast multipliers. One of the image processing applications is edge detection. This research presents a small area and high speed 8 bit Vedic multiplier system comprising of compressor based adders. This results in faster edge detection. This architecture is tested on Xilinx vertex 4 FPGA board and simulations were carried out using the Xilinx synthesis tool. Comparisons are made and this system is found to be smaller in area with high speed (the lesser propagation delay). This compressor based Vedic multiplier is 1.1 times speedier than a typical Vedic multiplier. Also, this Vedic Multiplier is 2 times speedier than a 'simple' multiplier.

***Keywords*—**Detection of edges, Vedic multiplier, image processing, Urdhva Tiryakbhyam sutra.

I. Introduction

SIGNAL processing technique processes an image. For this, the input is a picture or an image of a photo and the result of processing can be a photo, an image or a set of parameters of an image. Edge detection is one of the primary operations in image processing. Detection of edges is used for extraction of a feature. Edges are the boundaries between different regions in an image. Detection of edges helps with a recognition of an object and segmentation.

For the design of digital signal processing operations, multiplication performs a significant role in signal processing operations such as convolution and correlation, etc. In today's time, Vedic Mathematics is becoming popular and it is a speedier method to compute and analyze.

In this research work, Vedic multiplier is designed using adder which are compressor based. Full adder adds only three bits at one time, whereas compressor based adders are capable of adding more than three bits in one time. Compressor based adders count the number of ones only. The delay is reduced as XOR Gates are reduced [1]. So these adders have more speed and less area comparatively. Thus, the idea behind this research work is to improve the speed of the convolution operation through the proposed Vedic Multiplier and further implement it in an image processing application. In addition, a comparative analysis has been done between the presented multiplier and existing architecture on the basis of propagation delay as well as hardware use of a FPGA in total.

The presented high speed Vedic Multiplier is used for edge detection, thereby improving the computational speed involved in multiplication operations of image processing.

The research is divided as follows: Section II is the literature review. It gives the details on Vedic Multiplier, their different architectures and their applications in digital signal processing and image processing. Section III deals with the Vedic Multiplier's sutra (algorithm). Section IV describes the architecture of compressors and their use over conventional adders. Section V presents the architecture of compressor based Vedic Multiplier implemented. Its functionality is explained through dot diagram. Section VI deals with an image processing mechanism. It includes convolution, edge detection algorithm, and block diagram of the process. A thorough explanation of all components used in an image display and interfacing is depicted. Section VII shows the experimental results along with comparative analysis and Section VIII is the conclusion.

II. Background

Vijayan et al. [2] designed high speed Vedic Multiplier using high speed Ripple Carry Adder for image processing. Because of this, the speed is increased. Kerur et al. [4] proposed Vedic Multiplier's application in compression of an image by improving the computational speed of matrix multiplication in Discrete Cosine Transform (DCT). It is because of DCT; computational time has been reduced significantly in comparison to other conventional multipliers. In [5], a modified Vedic Multiplier emerged as an efficient architecture to implement multilevel 2D Discrete Wavelet Transform to increase image resolution. The modified Vedic Multiplier uses 8 half adders and 1 full adder compared to 32 half adders in regular Vedic Multiplier. Therefore, it offers a reduction in the area and power by 20% and 10% respectively. Huddar et al. [6] researched a novel high speed architecture, utilizing compressor based adders and ancient Vedic mathematics. The compressor based Vedic multiplier is 1.12 times efficient than the existing 'Vedic Mathematics' based multiplier. Deepthi et al. [7] implemented Vedic Multiplier and Divider. This implementation is in the asynchronous low power DSP processor core. The speed is 4 times faster than a conventional array multiplier with a power advantage of 40%. Naaz et al. [8] presented application of the high speed Vedic

Shobha Sharma is with IGDTUW, Delhi, Kashmere Gate, India (e-mail: shobhaa_sharma16@ yahoo.co.in).
Amita Dev is principal with BPIBS, Shakarpur Delhi India (e-mail: amita_dev@hotmail.com).
Akanksha Kant was a M.Tech VLSI student at IGDTUW, Delhi, India (e-mail: akansha214@gmail.com)

Multiplier. It is designed using high speed carry select adder, due to which its delay and area are reduced. In [9], it is shown that Vedic mathematics is used in the design of a novel complex multiplier. This multiplier is used for high speed complex arithmetic circuits with wide applications, in the fields of image and signal processing. The propagation delay is only 4 ns and consumes 6.5 MW power. In [10], 7:3, 6:3, 5:3 and 4:3 compressors are used for partial products summation, in ancient mathematics' based multipliers. With the use of Vedic algorithm-'Urdhva Triyakbhyam' and efficient compressor, an efficient multiplier system is built. Use of higher order compressors, reduces the number of computational stages; thereby requiring lesser hardware. Hence, the proposed design achieves significant reduction in area, leakage power, dynamic power and total power dissipation. In [11], a new technique is presented to further reduce the delay of a Vedic multiplier. This research work uses compressor based adders instead of common full adders and half adders. Modified compressor based multiplier architecture is introduced to catch the forever expanding demand for the lower area design and high speed processing. This changed circuit utilizes the 7:2 compressor and 4:2 compressor circuits. Chaithra et al. [12] presented canny edge detection algorithm developed on Spartan FPGA 3E. They have shown 'evolved' interfacing of VGA to show images on the monitor. In this research work 128×128 image is taken and with the help of FPGA, it is shown on the monitor. In [13], a Sobel edge detection algorithm's FPGA implementation is done. A Sobel Edge detection algorithm is used for edge detection of images. The Sobel Edge detection algorithm is efficient in getting smooth edges and is less sensitive to noise. It was appropriate to choose Sobel edge detection because this operator incorporates not only edge detection but also smoothing operation. This can detect edges more efficiently even in the high noisy atmosphere.

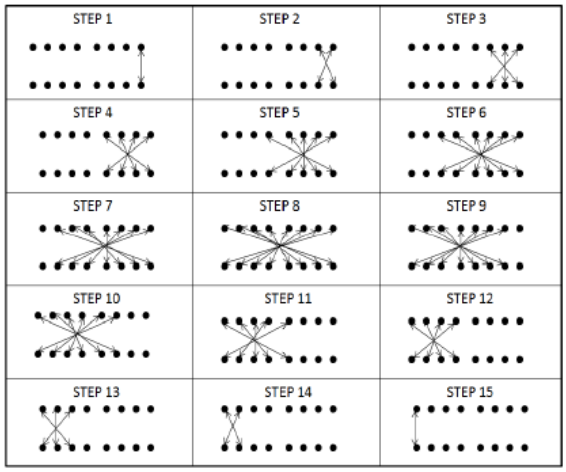

Fig. 1 Multiplication of 2 eight bit numbers using Urdhva Tiryakbhyam Sutra

III. Multiplication Using Vedic Mathematics

Vedic mathematics has 16 algorithms. These are known as sutras. The most efficient algorithm for speed is "Urdhva Tiryakbhyam (UT)". This sutra actually means vertically and crosswise. It makes almost all the numeric computations faster by generating partial product and sum in a single iteration. The advantage of the multiplier based on this sutra [14] is less increase of the area and delay, with the increase in the number of bits. Fig. 1 graphically explains the detailed method of multiplying two '8 bit' numbers using UT algorithm. The black circles indicate the bits of the multiplier and multiplicand, and the two-directional arrows show the bits to be multiplied, to achieve the individual bits of the final sum of product.

IV. Compressor Adders

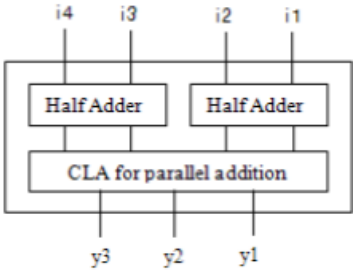

Fig. 2 Block diagram of 4-3 compressor

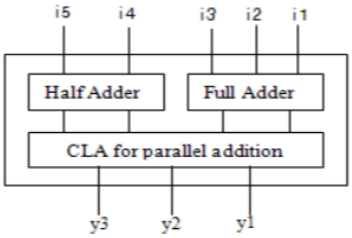

Fig. 3 5-3 compressor Block diagram

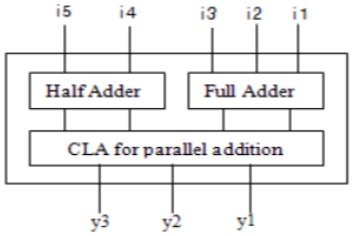

Fig. 4 6-3 compressor Block diagram

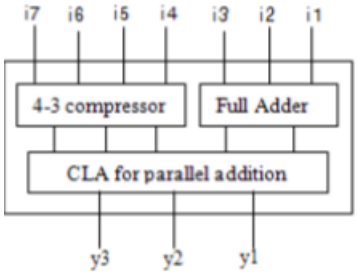

Fig. 5 7-3 compressor Block diagram

In the conventional design, traditional half adder and full adders are used to get the sum of intermediate products. The full adder can add only 3 bits in one go, so an extra hardware and intermediate stages are required. Here the compressor comes into the picture. Compressor architectures can add more than 3 bits in one go, with smaller circuit and increased speed. The block diagram of 4-3, 5-3, 6-3 and 7-3 compressors are shown in Figs. 2, 3, 4 and 5 respectively. The block diagrams of 4-3 compressor contain three sub-units. It consists of two adder units and a Carry Look Ahead adder (CLA) unit, for parallel addition of the outputs. The 5-3 compressor can add 5 single bit input, which intern produces a 3 bit result. The 6-3 compressor can add 4 single bit input digits, which intern produces a 3 bit result. The 7-3 compressor is implemented using 4-3 compressor, a full adder and a CLA unit [10].

V. Compressor Based Vedic Multiplier

In the presented architecture, higher order compressors have been used. The summations of partial products are done in two stages to obtain the final result and hence giving an area efficient design.

The dot diagram of the proposed multiplier is shown in Fig. 6. Adders and Compressors are employed such that the minimum number of output bits is generated. For example, in column 5, there are 7 partial products to be added [10]. These could be added using a full adder and 4-3 compressor, thereby generating five output bits. But instead of this, a 7-3 compressor has been used which will generate only 3 output bits. The same approach has been utilized for other columns as well.

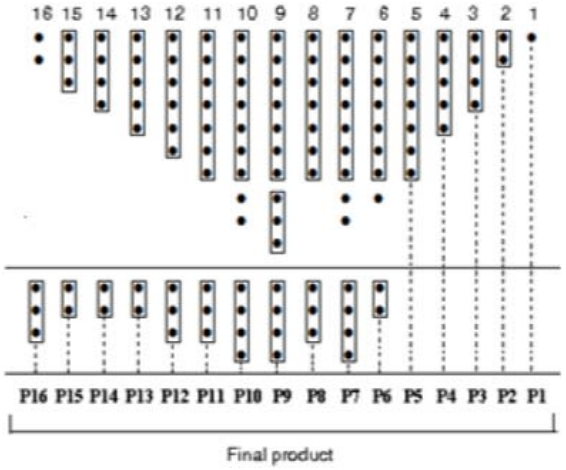

Fig. 6 Dot diagram of proposed 8*8 multiplier

VI. Image Processing

Image Processing is nothing but processing of an image using 'some' operators in order to have good information [2]. An intensity value is represented by each pixel value of a 2 dimensional matrix of a gray type picture. In 8 bit FPGA, only three bits are in red and green colour. Two bits are in blue colour. So, we chose a white and black image as an input image to be processed in this research work. In a black and white image, 4 bits are for black color and four are for white.

A. Convolution

Convolution is an initial and basic idea in an analysis of signal processing technique. Convolution is used in many techniques of image processing. Two dimensional input signals are operated upon by convolution. These give the output signals and are for the input dimensions. Convolution may be considered as modifying the kernel, which is an inverted on input. Wherever overlapping is there, multiplication is performed. At a position wherever kernel is in the middle, the output value is found. All such outputs together give the output in final form. In this multiplication process, Vedic multiplier is used [2].

Images are shown in 8bit binary form, when a convolution operation of two dimensions is put on pictures. Multiplication of convolution kernel coefficient and picture intensity values happens in convolution. Here multiplication is done with 8 bit Vedic multiplier.

We take two dimensional numbers of f and g. They should be in sequence. Convolution operation is acted on single dimension sequence using (1) [3]. Equation (2) is similar in nature:

$$(f * g)[x,y] = \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} \mathrm{f[m,n]g[x-m,y-n]} \quad (1)$$

$$(f * g)[x,y] = \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} f[x-m,y-n]g[m,n] \quad (2)$$

B. Edge Detection

To implement edge detection, convolution is used as a basic technique. So, if edge detection on images/pictures is made quickly, the very heavy tasks can be done quickly [2].

Sobel operator is incorporated in edge detection algorithms of image processing. It outputs an image which highlights transitions and edges. The Sobel operator uses kernels which are two 3×3. These are convolved with the original image. This gives approximations on the derivatives. The source image is an A. Two images are G_x and G_y. These images at each point, contain the horizontal and vertical derivatives. The processing is as follows:

$$Gy = \begin{matrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{matrix} * A$$

$$Gx = \begin{matrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{matrix} * A$$

where * denotes the 2-dimensional convolution operation.

C. Method

VGA screen shows the convolution operation, which is done using multiplier and processed picture/image. Block RAM known as BRAM1 is storing the picture/image in the form of column vectors. Sobel operator is used for carrying out convolution using 3*3 kernel. Image stored in BRAM1 is accessed. This image is used as input to MAC block which is

'multiply and accumulate' block. Weights are attached with each pixel during convolution. Normalization of resultant value is done and this value goes to next BRAM called BRAM2. This value is again reset to zero in multiply and accumulate unit. This keeps repeating till the image' last value is reached. This way complete picture is convolved with the help of convolution kernel. Then a signal is given to block RAM2, so that the data can be sent to VGA screen. Fig. 7 is the block diagram of an image processing method used in this research work.

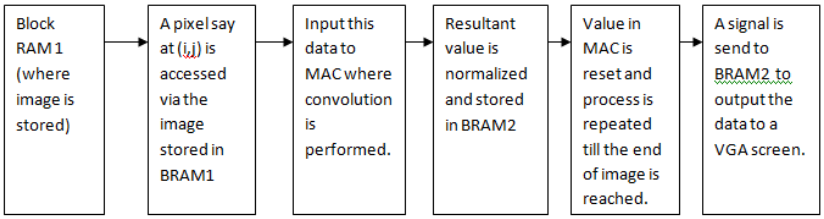

Fig. 7 Block diagram of Image Processing [12]

D. Components

1. *Block RAM:* Gray scale picture is of size 90X90, which is an input image. Pixel size is about 8 bits. MATLAB is used to produce the coefficient (coe) file of the input picture. Block RAM stores this COE file [12]. Virtex-4 FPGA's block RAM has the capacity to store up to 18Kb data [2].
2. *Multiplier and Accumulator (MAC):* MAC unit analyzes and multiplies the two numbers. This number is added to the previously stored number. Everything in MAC unit is the same except that the normal multiplier is substituted with a Vedic multiplier for performing the convolution operation [2].
3. *VGA Interfacing:* Video Graphics Array is the typical interface for the display of video. This interfacing is done between FPGA to the monitor to show the image of edge detection. Along with the developments of speedy image processing techniques, VGA Interfacings are getting demands to display real time pictures [12]. Intensity values of the output pictures are sent to be shown on a VGA screen. These values are put in block RAM2. With the help of on board VGA controller, these output image intensity values are sent in a serial way to the VGA monitor. The size of a typical VGA screen is 640X480. As shown in Fig. 8, each pixel is made to turn off and on by scanning the column sequentially from column 0 to column 639 and by scanning the row, in a sequence from row 0 to row 479.

During the scan, a very high rate horizontal shifting of the output values on the monitor is done. Scanning starts from the left corner of top area and reaches the end of the right corner of bottom. This process is repeated again and again. Refresh rate is termed as the rate at which scanning is done. This scanning rate should be more than 30 otherwise flickers will be shown. In retrace action, all the pixels are turned off on video graphics array monitor.

A VGA monitor is controlled by 4 signals. These are horizontal sync, vertical sync, horizontal blank and vertical blank signals. Scan rate timings are dependent on synchronization signals. To control the horizontal deflection circuitry, horizontal synchronization signal is responsible. This is done for the proper display of pixels in the display area of the monitor. For vertical direction, vertical synchronization signals are used to carry out the same function as a horizontal synchronization signal. Depending on the state of horizontal blank signal, the input of color signals to VGA is controlled.

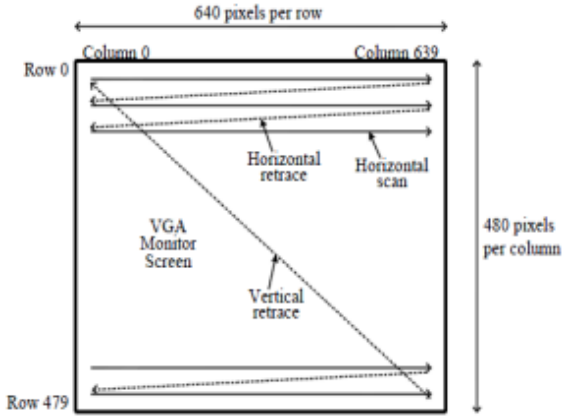

Fig. 8 VGA Screen

Fig. 9 VGA screen with Xilinx Vertex 4 development board

For displaying the edge detected image, Fig. 9 shows the Xilinx Vertex 4 FPGA's interfacing with VGA screen.

VII. Experimental Results

TABLE I
Comparison Results of Proposed Multiplier with Existing Multipliers

Type	Delay (ns)	% by which proposed multiplier is faster	Percentage of area utilization
Proposed Vedic multiplier using a compressor	16.935	-----	1
Conventional multiplier without compressor	18.374	13	2
Conventional multiplier	31.8	87.7	7

Fig. 10 shows the delay comparison of the three multipliers and Fig. 11 shows the percentage of area utilization of the three types of multiplier.

In this research, various higher order compressors have been utilized to design an 8x8 multiplier. The comparison of performance of Urdhwa Tiryakbhyam based conventional Vedic multiplier without compressors, proposed compressor based Vedic multiplier and conventional multiplier have been done. This is shown in Table I.

As compared to the traditional multiplier circuit, this multiplier circuit has a major improvement in performance. This 8 bit Vedic multiplier, implemented on Xilinx vertex 4 FPGA' has a delay of 16.935 ns. It can be seen that in comparison with conventional Vedic multiplier without compressor, this Vedic multiplier is 1.1 times faster in terms of speed and is 2 times speedier than 'just a conventional ' multiplier.

The use of higher order compressors reduces the number of computational stages, thereby requiring lesser hardware. In comparison to full adder and half adder based compressors, 4:2 compressor has less area due to reduced number of gates. In comparison to Vedic multiplier without compressor, this Vedic multiplier with compressor has 1% less area and 6% less area compare to conventional multiplier. Thus the presented design achieves significant reduction in area. Hence proved that this proposed compressor based Vedic multiplier is speedier and is an area efficient compared to other Vedic multiplier. Fig. 12 shows this multiplier's waveform.

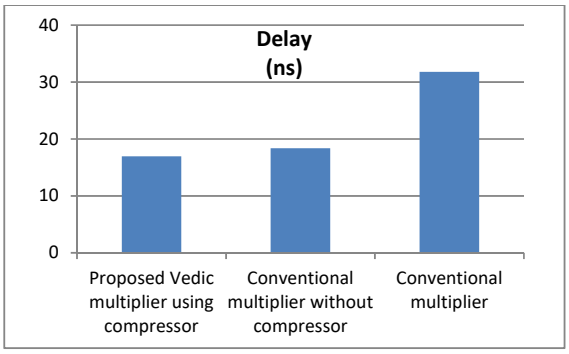

Fig. 10 Delay comparisons of different multipliers

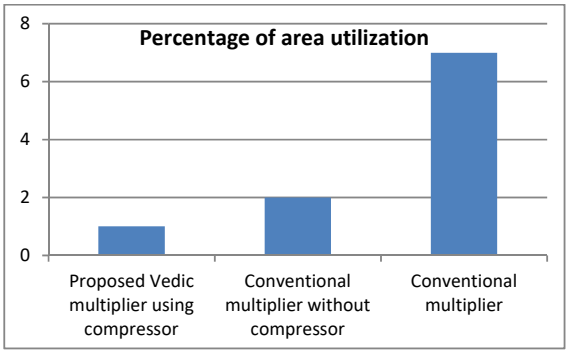

Fig. 11 Comparison of area utilization of different multiplier

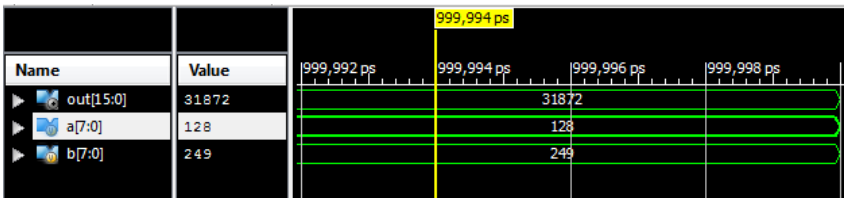

Fig. 12 Compressor based Urdhwa Tiryakbhyam VM simulation waveform

In this research work, we implemented Image Processing for Edge Detection on Xilinx Vertex 4 FPGA. FPGA's block RAM stores an image of size 90X90. Then the edge detection takes place and VGA screen displays this image. At each point, gradient of two dimensional map is formed which is nothing but edge detection (Sobel operator). This result can be seen as an image with edges as white lines. Fig. 13 is the input image and Fig. 14 is the output picture obtained after input is acted upon by edge detector.

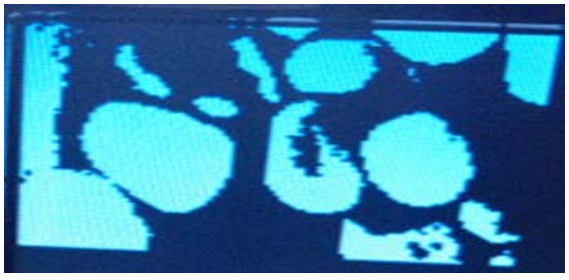

Fig. 13 Input Image

Fig. 14 Result of an output image after edge detection to an input image

VIII. Conclusion

This research work has presented the Vedic Multiplier based on UT sutra and is realized using compressors. This Vedic multiplier is more efficient with respect to speed and has less area utilization. Real time high speed edge detection is done efficiently using this Vedic multiplier. This Vedic multiplier has its usefulness in image processing where less area and high speed are the requirements.

References

[1] Yogita Bansal, Charu Madhu and Pardeep Kaur, "High Speed Vedic Multiplier Designs- A Review", Proceedings of 2014 RAECS UIET Punjab University Chandigarh, 06 – 08 March, 2014.

[2] Aravind E Vijayan, Arlene John and Deepak Sen, "Efficient Implementation of 8-bit Vedic Multipliers for Image Processing Application", International Conference on Contemporary Computing and Informatics (IC3I), 2014.

[3] Gonzalez and Woods, "Digital Image processing" 3rd Ed. Addison Weslet Pub.

[4] S. S. Kerur, Prakash Narchi, Harish M Kittur, Girish V. A, "Implementation of Vedic Multiplier in Image Compression using DCT Algorithm", 2nd International Conference on Devices, Circuits and Systems (ICDCS), 2014.

[5] J. Vinoth Kumar and Dr. C. Kumar Charlie Paul, "Design of Modified Vedic Multiplier and FPGA implementation in Multilevel 2d-DWT for Image Processing Applications", 2nd International Conference on Current Trends in Engineering and Technology, ICCTET'14.

[6] Sushma R. Huddar, S. Rao, Rupanagudi, Kalpana M., Surabhi Mohan, "Novel High Speed Vedic Mathematics Multiplier using Compressors", IEEE 2013.

[7] Deepthi P, V. Chakravarthi, "Design of Novel Vedic Asynchronous Digital Signal Processor Core", 2nd International Conference on Devices, Circuits and Systems (ICDCS), IEEE 2014

[8] A. Naaz.S, Pradeep N, S. Bhairannawar, Srinivas H., "FPGA Implementation of High Speed Vedic Multiplier Using CSLA for Parallel FIR Architecture", 2nd International Conference on Devices, Circuits and Systems (ICDCS), IEEE 2014.

[9] Prabir Saha, Arindam Banerjee, Partha Bhattacharyya, Anup Dandapat, "High Speed ASIC Design of Complex Multiplier Using Vedic Mathematics", Proceeding of the 2011 IEEE Students' Technology Symposium, 2011, lIT Kharagpur.

[10] Nidhi Pokhriyal, H. Kaur1, Dr. N. Prakash, "Compressor Based Area-Efficient Low-Power 8x8 Vedic Multiplier", Int. Journal of Engineering Research and Applications, Vol. 3, Issue 6, Nov-Dec 2013, pp.1469-1472.

[11] N Rajasekhar, T. Shanmuganantham, "A Modified Novel Compressor based Urdhwa Tiryakbhyam Multiplier", 2014 International Conference on Computer Communication and Informatics (ICCCI -2014), Jan. 03 – 05, 2014, Coimbatore, India.

[12] Chaithra. M., K.V. Reddy "Implementation of Canny Edge Detection Algorithm on FPGA and displaying Image through VGA Interface", International Journal of Engineering and Advanced Technology (IJEAT, Volume-2, Issue-6, August 2013

[13] Dhanabal R, Bharathi V, S. Kartika, "Digital Image Processing Using Sobel Edge Detection Algorithm in FPGA", Journal of Theoretical and Applied Information Technology, Volume 58, 2013.

[14] Hardik Sangani, Tanay M. Modi and V.S. Kanchana Bhaaskaran," Low Power Vedic Multiplier Using Energy Recovery Logic", International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2014.