Efficient FPGA based Matrix Multiplication using MUX and Vedic Multiplier

Article *in* INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY · January 2014

DOI: 10.24297/ijct.v12i5.2915

Raja K B
University Visvesvaraya College of Engineering
188 PUBLICATIONS **1,341** CITATIONS
SEE PROFILE

Venugopal K R
University Visvesvaraya College of Engineering
925 PUBLICATIONS **3,702** CITATIONS
SEE PROFILE

Lalit M Patnaik
Indian Institute of Science
837 PUBLICATIONS **8,673** CITATIONS
SEE PROFILE

ISSN 2277-3061

EFFICIENT FPGA BASED MATRIX MULTIPLICATION USING MUX AND VEDIC MULTIPLIER

Satish S Bhairannawar[1], Raja K B[2], Venugopal K R[3], L M Patnaik[4]

[1]Department of Electronics and Communication Engineering, DayanandSagar College of Engineering, Bangalore, India
[1]satishbhairannawar@gmail.com
[2]Department of Electronics and Communication Engineering, UVCE, Bangalore, India
[3]Principal, University Visvesvaraya College of Engineering, Bangalore University, Bangalore, India
[4]Honorary Professor, Indian Institute of Science, Bangalore, India

ABSTRACT

Most of the algorithms which are used in DSP, image and video processing, computer graphics, vision and high performance supercomputing applications require multiplication and matrix operation as the kernel operation.In this paper, we propose Efficient FPGA based matrix multiplication using MUX and Vedic multiplier. The 2x2, 3x2 and 3x3 MUX based multipliers are designed. The basic lower order MUX based multipliers are used to design higher order MxN multipliers with a concept of UrdhvaTiryakbyham Vedic approach. The proposed multiplier is used for image processing applications. It is observed that the device utilization and combinational delay are less in the proposed architecture compared to existing architectures.

Indexing terms/Keywords

Multiplier, MUX, Matrix, FPGA, Image processing, Vedic multiplier.

SUBJECT CLASSIFICATION

VLSI for Signal Processing.

Council for Innovative Research

Peer Review Research Publishing System

Journal: INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY

Vol 12, No. 5

editor@cirworld.com

www.cirworld.com, www.ijctonline.com

INTRODUCTION

Matrix multiplication plays an important role in image processing applications. The computation algorithms such as image processing, video processing, numerical analysis and computer graphics involve matrix operation as the kernel operation. The performance of matrix multiplication is evaluated with respect to speed and area of hardware system. Matrix-vector multiplication generally requires several Multiply Accumulate Units(MAC). The traditional DSP processors are limited by parallel processing of multiplications and additions which in turn take several clock cycles to perform all necessary MAC operations. Field programmable Gate Arrays (FPGA's) which involve parallel processing are equipped with large embedded resources. Modern FPGA's can provide higher and more efficient processing rates with efficient resource utilization. Multipliers which form basic computational unit in FPGA's plays a major role in increasing the speed of intensive matrix computations for image processing applications. Vedic algorithms[1] used in multiplication reduces the delay in generating partial products as the architecture is based on vertical and crosswise structure of ancient Vedic mathematics which generates the partial products and their sum using minimum number of clock cycles. This also has an advantage of modular design where low order multipliers are used to design higher order multipliers of any size which reduces the design complexity while dealing with larger number of bits.

*Contribution:*In this paper Efficient FPGA based matrix multiplication using MUX and Vedic multiplier is proposed. The competent MxN multiplier is designed using Low order MUX based multipliers and Vedic multiplier.

*Organization:*The paper is organized as follows: Section II proposes the Related Work. Section III gives details of Methodology. Section IV provides the performance analysis and results. Section V provides conclusion followed by future work.

RELATED WORK

ZdenkaPurushottam D. Chidgupkar and Mangesh T. Karad [2] proposed multiplication process based on Vedic mathematics and its implementation on 8085 and 8086 microprocessors. A comparative study of processing time of conventional multipliers for 8085 and 8086 were analysed and shown that there is an appreciable saving in the processing time of the Vedic multiplier as when compared to that of a conventional multiplier. Vedic algorithms implementations on specially designed BCD architecture will also help to enhance processor throughput.Honey DurgaTiwari et al., [3] proposed multiplier and square architecture based on algorithm of ancient Indian Vedic Mathematics for low power and high speed applications. It is based on generating all partial products and their sums in one step.The multiplier using UrdhvaTiryakbyham sutra and Nikhilam sutra was compared with array and booth multiplier and reached to the conclusion that result obtained from Vedic multiplier is faster than array multiplier and Booth multiplier. SumitVaidya and Deepak Dandekar [4]proposed expanded UrdhvaTiryakbyham sutra for 16 bit multiplied output. Comparative study of different multipliers was done for low power requirement and high speed. The expanded UrdhvaTiryakbyham sutra is not an efficient algorithm for the multiplication of large numbers as a lot of propagation delay is involved in such cases and to overcome this problem, Nikhilam Sutra was suggested.

Paramasivam and Sabeenian [5] proposed a method for decomposing a perfect binary multiplication into smaller size using Nikhilam sutra and hence reducing the computation time and power consumption. The algorithm was broadly divided in three parts namely the initialization, pre-processing and processing. The algorithm evidently reduces a given 4 bit multiplication to a 2-bit multiplication by making use of basic shifting and addition operations, as a result of which the carry propagation in any standard 4 x 4 - bit multiplier is reduced to a great extent. Jayaprakasan et al., [6] discussed the use of an ancient (or *Vedic*) mathematical approach for building an ALU. Validation for the low power operation of the circuit were made by designing a conventional CMOS counterpart whose power is compared with ancient arithmetic design.Binary 4x4 Array Multiplier and UrdhvaTiryakbhyamas VedicMultiplier were taken for the comparison. Finally it was concluded UrdhvaTiryakbhyamis best for multiplication with respect to number of adders and power consumption.ShamimAkhter [7] proposed VHDL implementation of a NXN multiplier based on the Vedic mathematics.This gives less computation time for calculating the multiplication result for NxN bit and a way to implement the design of the Urdhva sutra based multiplier used bottom up design methodology. The design complexity gets reduced for inputs of large number of bits with increase in modularity.

HimanshuThapliyalet al., [8] proposed a design for square and cube architectures. It was very clearly evident that, the Vedic square and cube architecture were faster than the conventional square and cube calculations. The parallel architectures for computing square and cube of a given number based on ancient Indian Vedic mathematics were discussed. Vaithiyanathan et al.,[9] proposed the comparative study of different multipliers for low power requirement and high speed, also gives information ofUrdhvaTiryakbhyam algorithm of Ancient Indian Vedic Mathematics which is utilized for multiplication to improve the speed, area parameters ofmultipliers.Sriraman andPrabakar [10] proposed multiplier architecture based on ROM approach using Vedic Mathematics. Theproposed architecture is similar to that of a Constant Coefficient Multiplier (KCM). However, for KCM one input is to be fixed, while their proposed multiplier can multiply two variables.Harpreet Singh Dhillon and AbhijitMitra [11] proposed NikhilamSutra algorithm which was further optimized by use of some general arithmetic operations such as expansion and bit-shifting to take advantage of bit-reduction in multiplication. Thealgorithm was implemented by reducing a general 4x4-bit multiplication to a single 2x2-bit multiplication operation.

NareshNaik et al.,[12]proposed a method of multiplication technique by using Vedic mathematicsformula UrdhavaTiryakbhyam method which meansvertically and cross wire. All the operations in Vedicmultiplier were executed concurrently.Further the speed comparisons of thismultiplier with Normal Booth multiplier were presented.The results showed thatUrdhavaTiryakbhyam multiplier has great amount of impact on the DSPapplications to improve the execution speed of the DSPprocessors when compared to other multipliers.Jang et al., [13] proposed matrix multiplication as the benchmark to compare the performance of FPGAs, DSPs and embedded processors.The results show that the FPGAs can multiply two matrices with both lower latency and lower energy consumption than the other two types of devices making FPGA ideal choice for matrix multiplication in signal processing applications.Belkacemi et al., [14] presented the design and implementation of a high performance, fully parallel matrix multiplication core. The core was parameterized and scalable in terms of the matrix dimensions (i.e., number of rows and columns) and the input data word length. Fully floor planned FPGA configurations were generated automatically, from high-level descriptions of the matrix multiplication operation, in the form of Electronic Design Interchange Format (EDIF) netlist in less than one second.

Jianwen and Chuen [15] proposed Partiallyre-configurability feature which was exploited for the first time to compute matrix multiplication. Partially reconfigurable devices offer the possibility of changing the design implementation without stopping the whole execution process. The design was evaluated in terms of latency and area.MahendraVucha and ArvindRajawat[16] presented an effective design for the Matrix Multiplication using Systolic Architecture on Reconfigurable Systems like FPGAs. Here, the systolic architecture increases the computing speed by combining the concept of parallel processing and pipelining into a single concept.Syed M. Qasim et al., [17] presented a preliminary design and FPGA implementation of dense matrix–vector multiplication for use in an image processing application. The architecture was designed to multiply large matrix and a vector.Nivedita A. Pandeet al., [18] proposeda designmethodology for high-speed multiplications, where two integers of n-bit size each are multiplied to produce a 2n-bit product.

METHODOLOGY

In this section, we have introduced new concept of MxN bit multiplication based on multiplexer and UrdhvaTiryakbyham sutra (vertically and crosswise)Vedic concept. The disadvantage of direct multiplication s using UrdhvaTiryakbyham Vedic concept for higher order bits require more number of carry propagation results in more delay. The multiplexer based multiplier with UrdhvaTiryakbhyam Sutra concept eliminated delay and minimizes IC package count.

i)Design of MUX based 2x2 and 3x2 multiplier:

The 2x2 and 3x2 MUX based multiplier using 4:1 multiplexer as shown in Fig. 1 with A as multiplicand and B as multiplier. The four input lines for multiplier B with two bits for 2x2 multiplier and three bits for 3x2 multiplier are considered with A multiplicand having two controls S0 and S1.The first line input of multiplier B is always s 00 or 000. The second, third and fourth line input values are any combinations of two or three bits based on either 2x2 or 3x2 multiplier . The third line input is connected to shift left by one shifter and fourth input line is connected to shift left by one alone with adder to get proper multiplication results in the output

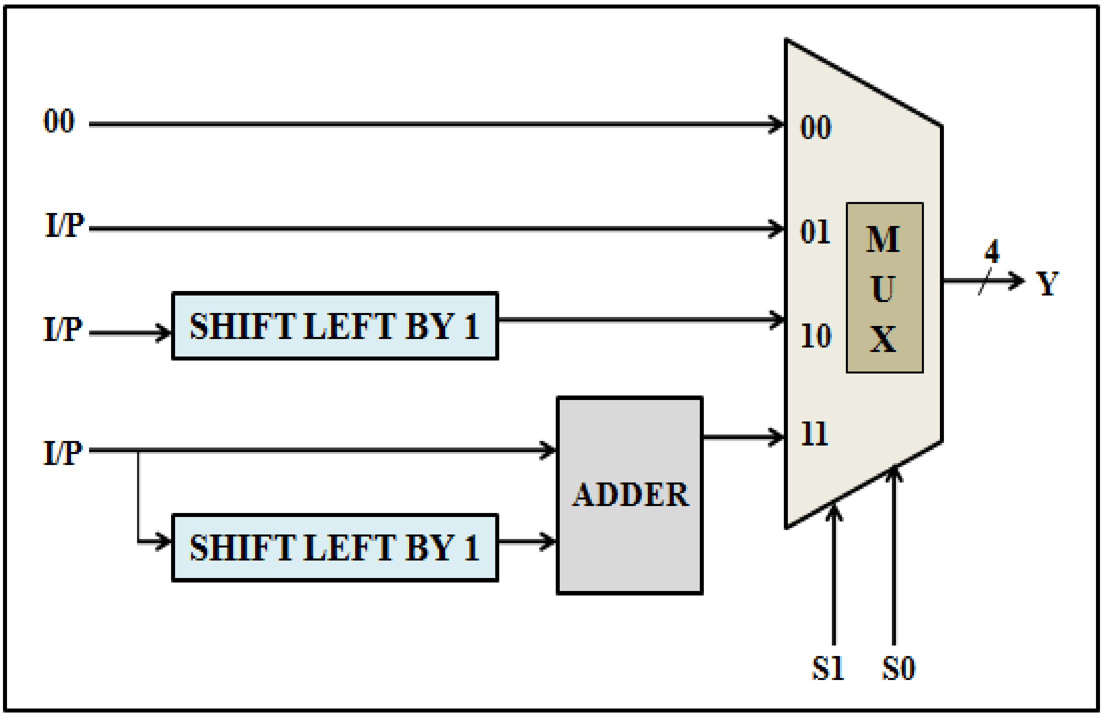

Figure1: Hardware architecture of 2x2 and 3x2 multipliers

The truth table of 2x2 mux based multiplier is given in Table 1. The input values the first line is always 00 or 000. For the control bits 01, the input of second line is passed to the output i.e. the multiplication of A and B. The third line input is shifted left by one and is passed to the output i.e. multiplicationof A and B. The fourth line input shifted left by one and added is passed to the output which is multiplication of A and B.

Table1: Truth table of 2x2 Mux based multiplier.

A Multiplicand		B Multiplier		Output Y			
S1	S0	B1	B0	Y3	Y2	Y1	Y0
0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

DESIGN OF MUX BASED 3X3 MULTIPLIER:

The 3x3 multiplier using multiplexer using 8:1 multiplexer is shown in Fig. 2. The multiplicand A has three bits S0, S1, S2. The multiplier B has eight input lines with first line input is always 000. The second line input values are any combinations of three bits and values are passed to output for control value 001. The third input line and forth input line are connected to shift left by one and shift left by one and shift left by two respectively and correspond input are passed to output for control values 010 and 100 respectively. The fourth and sixth line input are connected to shift left by one with adder and shift left by two with adder respectively and the corresponding modified inputs are transferred to output for control values 011 and101.The seventh line input is connected to shift left by one and two with adder and modified input is transferred to output for control value 110. The eight line input is connected to shift left by one and two. The modified input of eight lines is an addition of shift left by one, shift left by two and direct eight line input. The modified eight line input is transferred to output for control value 111.

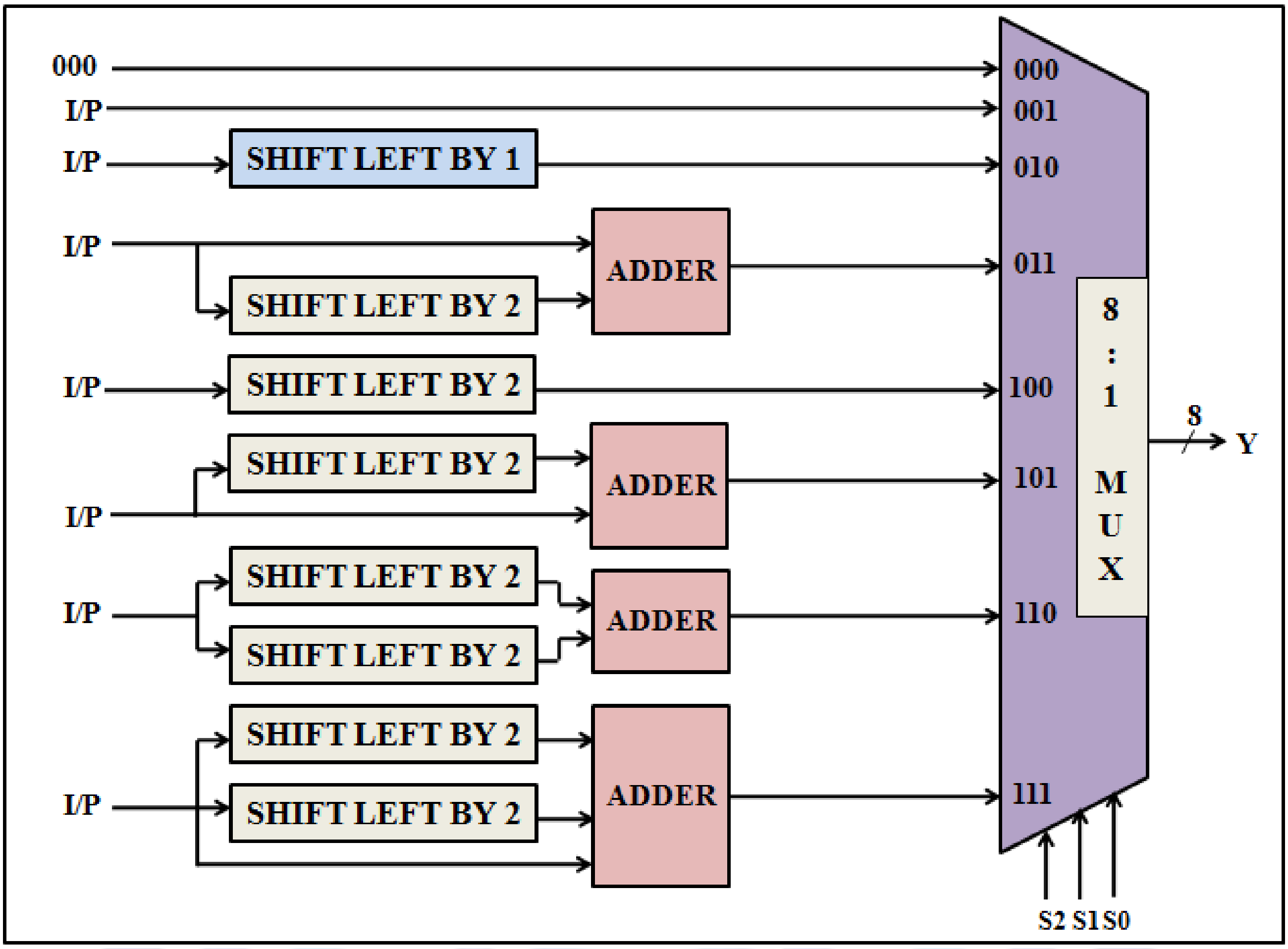

Figure 2: Hardware architecture of 3x3 multiplication.

PROPOSED 8X8 MULTIPLIER BASED ON MUX AND VEDIC CONCEPT

Multiplier using 3x3, 3x2, 2x2 mux based multiplier is shown in Fig3. The two eight bit number(a7 a6 a5 a4 a3 a2 a1 a0) and (b7 b6 b5 b4 b3 b2 b1 b0) are considered for multiplication. The bits (a2 a1 a0) and (a5 a4 a3) and (a7 a6) are group as C,B and A respectively. The bits (b2b1 b0), (b5b4b3) and (b7 b6) are grouped as F, E and D respectively.

The following steps are used for multiplication,

step1: The group C and F are multiplied using 3x3 mux based multiplier.

step2: (B*F) + (E*C) + carry of step1.

step3: (F*A) + (D*C) + (B*E) + carry of step2.

step4: (E*A) + (B*D) + Carry of step 3.

step5: (A*D) + carry of step4.

The one block of 2x2, four block of 3x2 and four block of 3x3 mux based multiplier along with four adders are used in 8x8 multiplier using Vedic concept as shown in Fig 4.

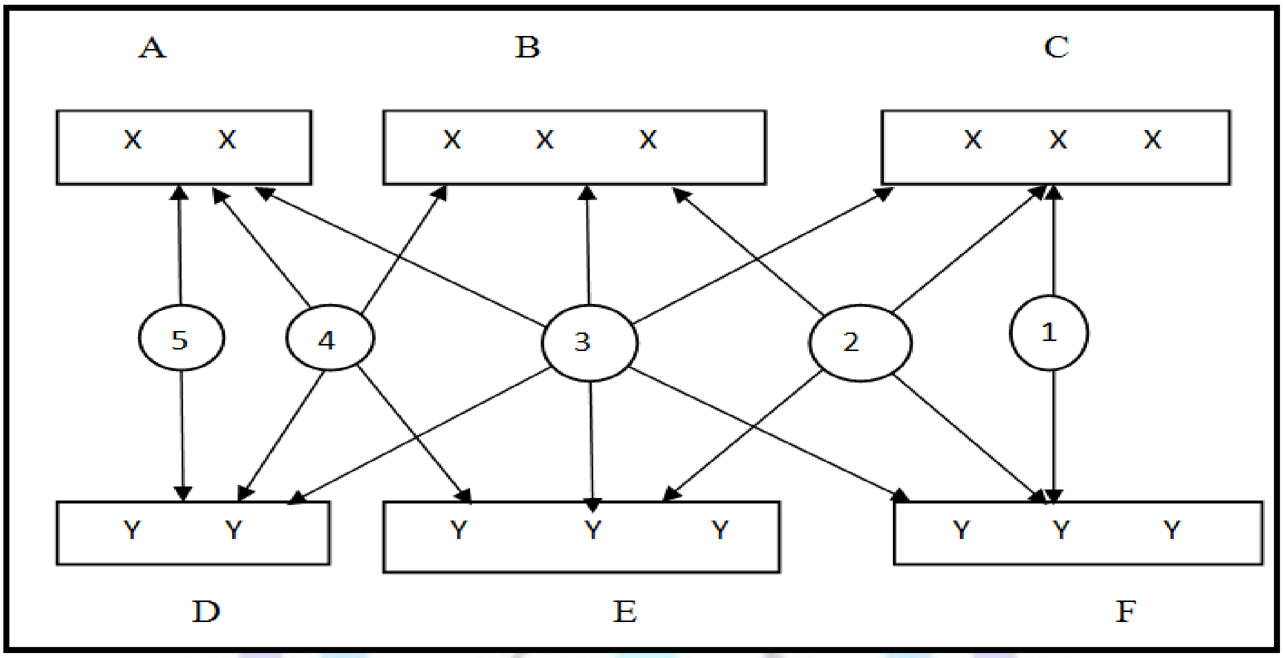

Figure 3.Line diagram of the proposed Vedic multiplier for 8x 8 bit multiplications

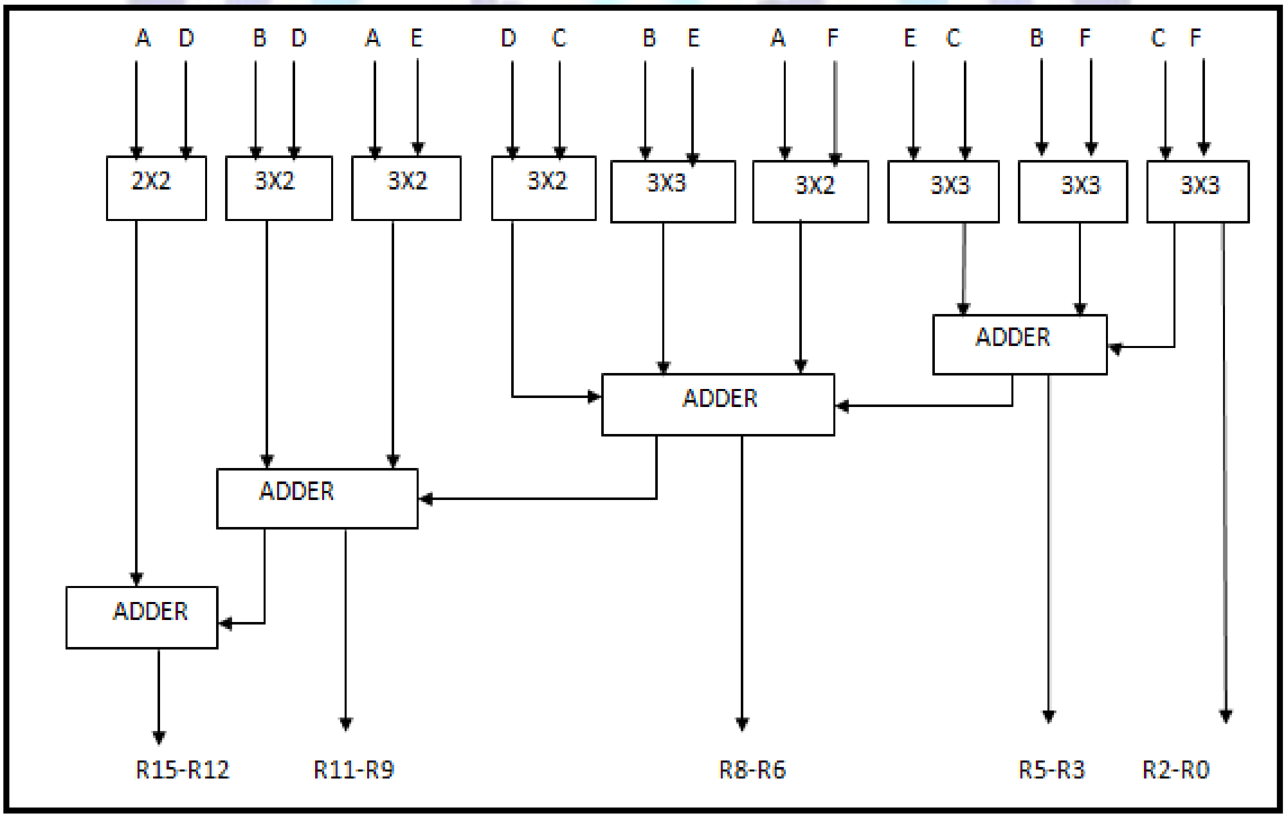

Figure 4. Architecture of 8X8 bit multiplier using 3X3, 3X2 and 2X2 bit multiplier blocks.

PROPOSED 16X16 MULTIPLIER USING 8X8 MULTIPLIER

The block diagram of 16 x16 multiplier using 8x8 multiplier is shown in Fig. 5. The four 8x8 multiplier along with two adders are used to implement 16x16 multiplier. The two numbers of 16 bits are a0 to a15 and b0 to b15 are considered. The proposed multiplier can be extended to any value of MxN.

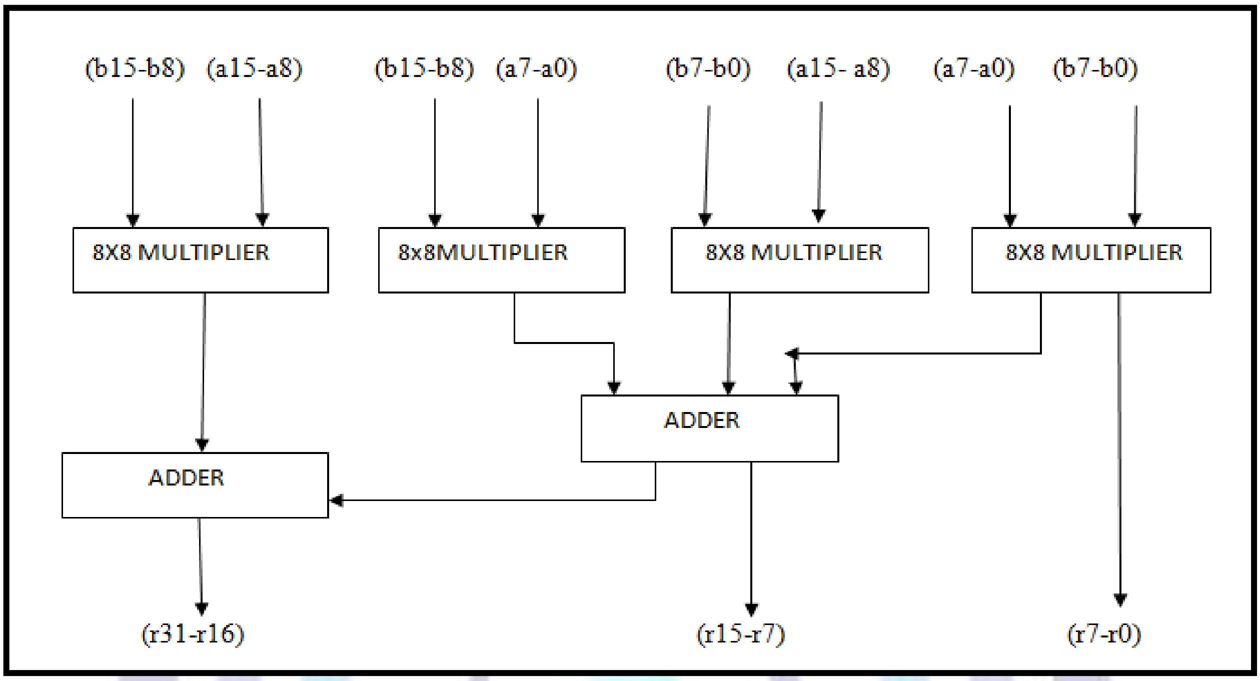

Figure 5. Architecture of 16x16 bit multiplier using 8x8 bit multiplier block

PROPOSED MATRIX ARCHITECTURE FOR IMAGE PROCESSING APPLICATIONS

The proposed multiplier of any size used in multiplication of two matrices of any size for image processing applications is shown in Fig. 6

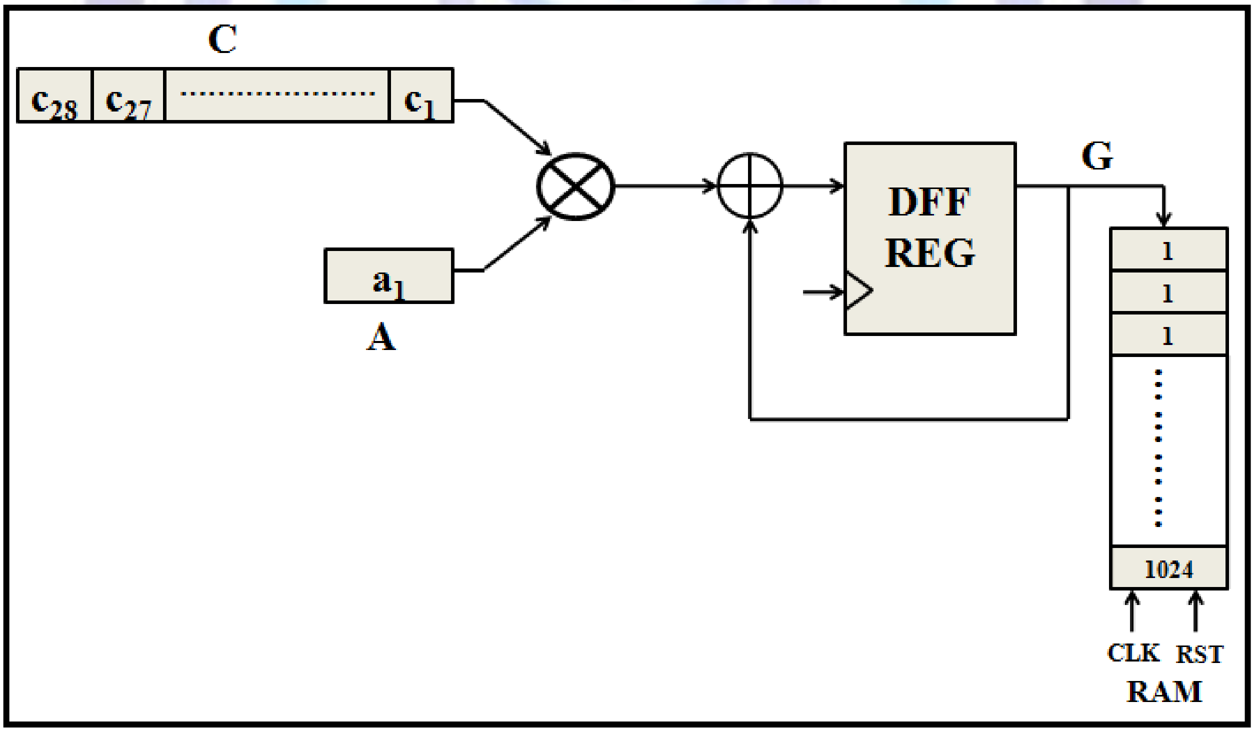

Figure 6. Architecture for matrix vector multiplication.

The matrices A and C are considered for multiplication. The corresponding column vector elements of C and the corresponding row elements of A shifted serially are multiplied by proposed mux based multiplier. The result of a multiplier is stored in RAM. For image processing applications, the image is considered as matrix A and the filter coefficients of any size can be considered as matrix C. The multiplication of any image with filter coefficients is performed using proposed mux based multiplier.

RESULTS AND DISCUSSIONS

Synthesis Results

The performance parameters such as number of slices, number of 4 input LUT's and maximum combinational path delay are considered to test the proposed multiplier using Xilnx Spartan 3 FPGA family [19]. The number of performance parameters is measured for 8x8 and 16x16 proposed multiplier and is tabulated in Table 1. It is observed that as the order of multiplication increases, the delay is not varied significantly.

TABLE I:PERFORMANCECOMPARISON OF 8x8 AND 16x16 MUX BASED MULTIPLIER.

Device : 3s50atq144-5	8x8 MULIPLICATION	16x16 MULTIPLICATION
Number of slices	72 out of 704	302 out of 704
Number of 4 input LUT	130 out of 1408	546 out of 1408
Number of I0s	32	64
Number of bondedIOB	32 out of 108	64 out of 108
Max combinational path delay	20.008ns	26.198ns

The performance parameters of proposed multiplier is compared with existing 8x8 multiplier presented by Pushpalathaand Mehta [20] and is tabulated in Table II. It is observed that the number of slices and the combinational delay is reduced in the proposed method compared to existing method.

TABLE II PERFORMANCE COMPARISION OF PROPOSED 8x8 MULTIPLIER WITH EXISTING MULTIPLIER.

Device : 3s50atq144-5	Existing method [20]	Proposedmethod
Number of slices	95 out of 704	72 out of 704
Number of 4 input LUT	166 out of 1408	130 out of 1408
Number of I0s	32	32
Number of bonded IOB	32 out of 108	32 out of 108
Max combinational path delay	21.679ns	20.008ns

The performance parameters of proposed multiplier is compared with existing 16x16 multiplier presented by Gurumurthy and Prahalad [21] and is tabulated in Table III. It is observed that the number of slices and the combinational path delay is reduced in the proposed method compared to existing method.

TABLE IIIPERFORMACE COMPARISION OF16X16 PROPOSED MULTIPLICATION AND EXISTING MULTIPLICATION.

Device: xc3s100e-5vq100.	Existing Method [21]	Proposed method
Number of slices	404 out of 960	302 out of 960
Number of 4 input LUT	716 out of 1920	546 out of 1920
Number of I0s	64	64
Number of bonded IOB	64 out of 66	64 out of 66
Max combinational path delay	37.668ns	25.609ns

TheMatrix multiplicationof 1028x28 and 28x1 is synthesized using Xilinx virtex4 200ff1513 board [22]. The performance comparison of proposed method with existing method presented by Syed MQasim[23] is given in table IV. It is observed that the number of slices and the combinational delay are reduced in the proposed method compared to existing method.

TABLE IV: PERFORMANCE COMAPRISON MATRIX MULTIPLICATION OF 1028x28 AND 28x1

Device : 4vlx200ff1513-11	Syed M. Qasim [23]	Proposed multiplier
Number of Slices	13010 out of 89088	368 out of 89088
Number of 4 input LUTs:	9612 out of 178176	580 out of 178176
Number of DSP48s	55 out of 96	--
Numberof FIFO16/RAMB16s:	--	2 out of 336
Maximum Frequency:	17.376 MHz	107.759MHz

Simulation results:

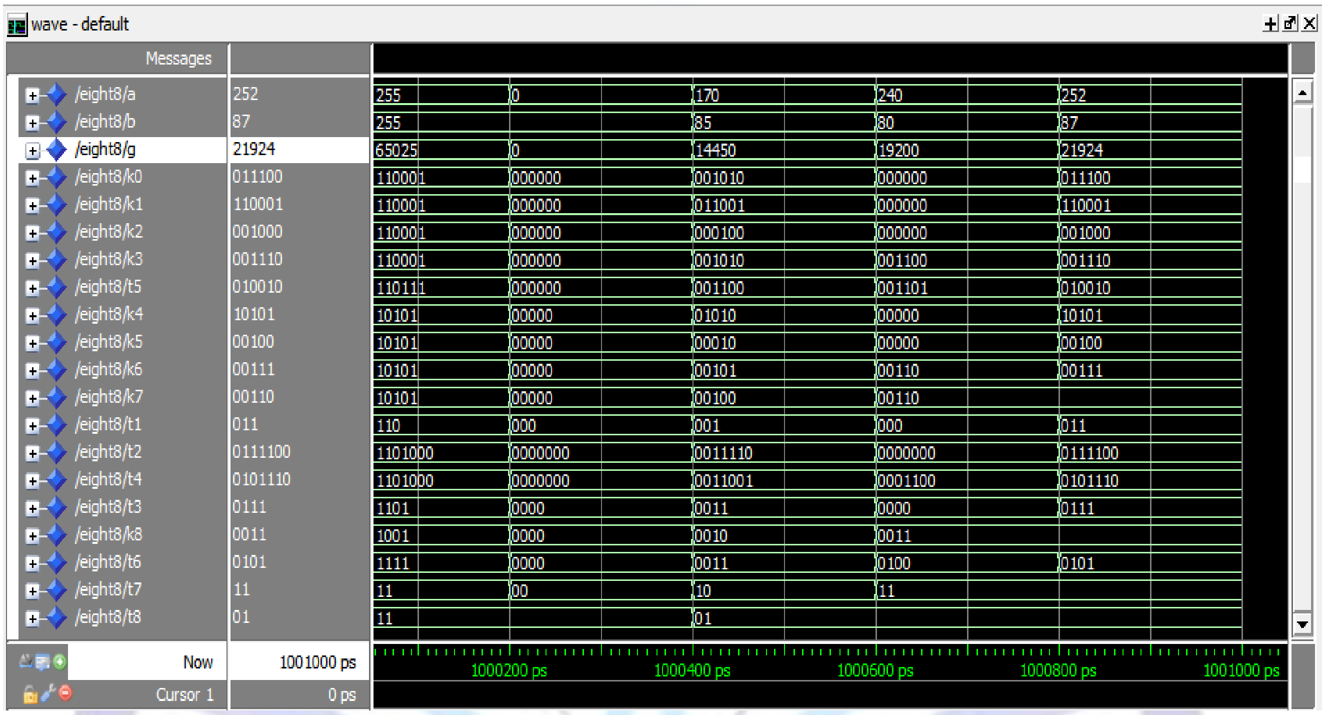
wave - default
Messages
/eight8/a 252 | 255 | 0 | 170 | 240 | 252
/eight8/b 87 | 255 | 85 | 80 | 87
/eight8/g 21924 | 65025 | 0 | 14450 | 19200 | 21924
/eight8/k0 011100 | 110001 | 000000 | 001010 | 000000 | 011100
/eight8/k1 110001 | 110001 | 000000 | 011001 | 000000 | 110001
/eight8/k2 001000 | 110001 | 000000 | 000100 | 000000 | 001000
/eight8/k3 001110 | 110001 | 000000 | 001010 | 001100 | 001110
/eight8/t5 010010 | 110111 | 000000 | 001100 | 001101 | 010010
/eight8/k4 10101 | 10101 | 00000 | 01010 | 00000 | 10101
/eight8/k5 00100 | 10101 | 00000 | 00010 | 00000 | 00100
/eight8/k6 00111 | 10101 | 00000 | 00101 | 00110 | 00111
/eight8/k7 00110 | 10101 | 00000 | 00100 | 00110
/eight8/t1 011 | 110 | 000 | 001 | 000 | 011
/eight8/t2 0111100 | 1101000 | 0000000 | 0011110 | 0000000 | 0111100
/eight8/t4 0101110 | 1101000 | 0000000 | 0011001 | 0001100 | 0101110
/eight8/t3 0111 | 1101 | 0000 | 0011 | 0000 | 0111
/eight8/k8 0011 | 1001 | 0000 | 0010 | 0011
/eight8/t6 0101 | 1111 | 0000 | 0011 | 0100 | 0101
/eight8/t7 11 | 11 | 00 | 10 | 11
/eight8/t8 01 | 11 | 01
Now 1001000 ps
Cursor 1 0 ps
1000200 ps
1000400 ps
1000600 ps
1000800 ps
1001000 ps

Figure 7. Simulation wave window for 8x8 multiplication.

Fig. 7 shows the simulation results of multiplier using two 8 bit numbers based on combination of multiplexer and Vedic multiplier for 5 different combinations of inputs with corresponding outputs.

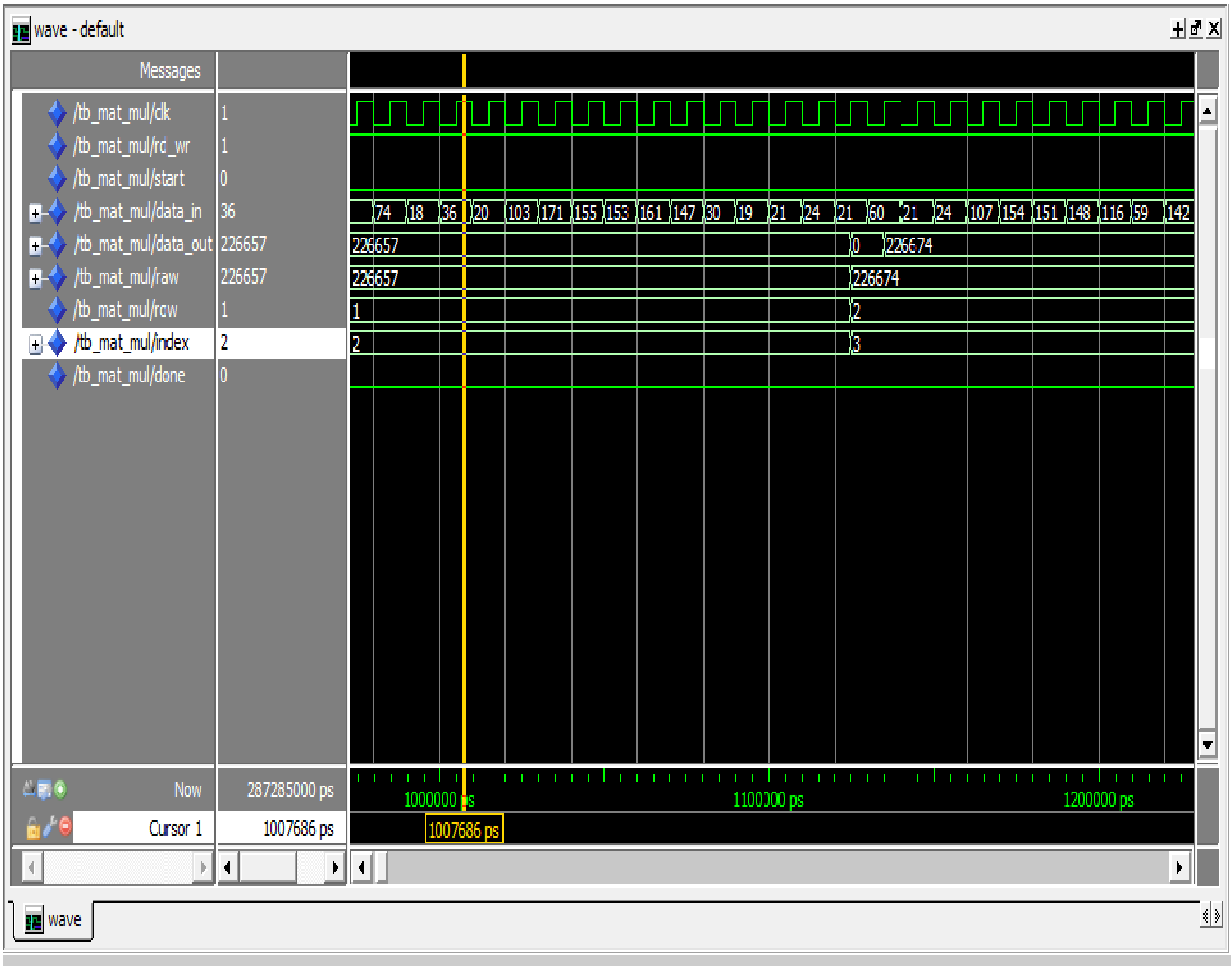

Fig 8. Matrix multiplication simulation wave window.

Fig. 8 shows simulator window of matrix multiplication of 8 bit numbers using Mux based Vedic multiplier for matrix multiplication.

CONCLUSION

Higher order multipliers are required in image processing applications. In this paper Efficient FPGA based Matrix Multiplication using Mux and Vedic multiplier is proposed. The lower order MUX based multipliers are used withVedic multipliers to design a novel higher order multiplier of any dimensions. The proposed multiplier is used in image processing applications. It is observed that the performance parameters such as area and delay are reduced compared to existing algorithms. In future, multiplier can be designed using higher order MUX based multipliers.

REFERENCES

[1] Jagadguru Swami Sri Bharati Krishna Tirthji Maharaja,“Vedic Mathematics”, Varanasi, India, 1986.

[2] Purushottam D. Chidgupkar and Mangesh T. Karad, “The Implementation of Vedic Algorithms in Digital Signal Processing”, *UICEE Annual conference on Engineering Education,* Vol.8, No. 2, pp. 153-158, February 2004.

[3] Honey DurgaTiwari, GanzorigGankhuyag, Chan Mo Kim and Yong Beom Cho, “Multiplier Design based on Ancient Indian Vedic Mathematics”, *International SoC Design Conference*,pp. 65-68,2008.

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

[5] M.E.Paramasivam and R.S.Sabeenian, “An Efficient Bit Reduction Binary Multiplication Algorithm using Vedic Methods”, *IEEE International Conference on Advance Computing*, pp. 25-28,2010.

[6] V Jayaprakasan, S Vijayakumar and V S KanchanaBhaaskaran, “Evaluation of the Conventional versus Ancient Computation Methodology for Energy Efficient Arithmetic Architecture”,*International Conference onProcess Automation, Control and Computing (PACC),* pp. 1-4,2011.

[7] ShamimAkhter, “VHDL Implementation of Fast NxN Multiplier Based on Vedic Mathematics”, *IEEE European Conference onCircuit Theory and Design*, pp.472 - 475,2007.

[8]HimanshuThapliyal, SaurabhKotiyal and M. B Srinivas, “Design and Analysis of a Novel Parallel Square and Cube Architecture Based On Ancient Indian Vedic Mathematics”,*IEEE Midwest Symposium on Circuits and Systems*, Vol. 2, pp. 1462 – 1465, 2005.

[9] G. Vaithiyanathan, K.Venkateshan,S. Shivaramakrishnan,S. siva and S. Jayakumar, “simulation and implementation of vedic multiplier using VHDL code”, *International Journal of scientific and Engineering Research*, Vol. 4, Issue 1, pp. 1-5, January 2013..

[10] L. Sriraman and T. N. Prabhakar, “Design and Implementation of Two Variable Multiplier using KCMand Vedic Mathematics”, *IEEEConference on Recent Advances in Information Technology,*pp. 782 – 787, 2012.

[11]Harpreet Singh Dhillon and AbhijitMitra “A Reduced-Bit Multiplication Algorithm for Digital Arithmetic”,*World Academy of Science, Engineering and Technology,*pp. 756-761,2008.

[12] R. NareshNaik, P. Siva Nagendra Reddy and Madan Mohan, “*Design of Vedic multiplier for DSP applications*, Vol. 4, Issue 7, pp.3025-3030, July 2013.

[13] J. Jang, S. Choi, and V. Prasanna, “Energy-Efficient Matrix Multiplication on FPGAs”, *International Conference on FieldProgrammableLogic and Applications,*pp. 534-544, 2002.

[14] S. Belkacemi, K. Benkrid, D. Crookes, and A. Benkrid, “Design and Implementation of a High Performance Matrix Multiplier Core for Xilinx Virtex FPGA”,*IEEE International Workshop on Computer Architectures for Machine Perception,* pp. 156-159, 2003.

[15] L. Jianwen and J. C. Chuen, “Partially Reconfigurable Matrix Multiplication for Area and Time Efficiency on FPGAs”, *Euromicro Symposium on Digital System Design*, pp. 244-248, 2004.

[16]MahendraVucha and ArvindRajawat, “Design and FPGA Implementation of Systolic ArrayArchitecture for Matrix Multiplication”,*International Journal of Computer Applications*, Vol. 26, No.3,pp. 18-22, July 2011.

[17]Syed M. Qasim, Ahmed A. Telba and Abdulhameed Y. AlMazroo, “FPGA Design and Implementation of Matrix Multiplier Architectures for Image and Signal Processing Applications”,*International Journal of Computer Science and Network Security*, Vol.10, No.2,pp. 168-176, February 2010.

[18]Nivedita A. Pande, VaishaliNiranjane and Anagha V. Choudhari”, Vedic Mathematics for Fast Multiplication in DSP”, *International Journal of Engineering and Innovative Technology*, Vol. 2, Issue 8, Febraury 2013.

[19]Xilinx, Spartan-3 FPGA family datasheet, June 2008.

[20]PushpalataVerma and K. K. Mehta, “Implementation of an Efficient Multiplier Based on Vedic Mathematics Using EDA Tool”, *International Journal of Engineering and Advanced Technology*, Vol. 1, Issue 5, pp. 75-79, June 2012.

[21] K.S. Gurumurthy and M.S Prahalad, “Fast and Power Efficient 16×16 Array of Array Multiplier using Vedic Multiplication”, *International Conference on Microsystems Packaging Assembly and Circuits Technology*, pp. 1 - 4, October 2010.

[22] Xilinx, Virtex-4 FPGA family overview, September, 2007.

[23] Syed M. Qasim, Ahmed A. Telba and Abdul hameed Y. AlMazroo, “FPGA Design and Implementation of Dense Matrix-Vector Multiplication for Image Processing Application”, *International Conference world congress on Engineering and Computer science*, Vol. 1, October 2010.

AUTHOR BIOGRAPHY

Satish S Bhairannawar is an Associate Professor, Dept of Electronics & Communication, DayanandaSagar College of Engineering, Bangalore. He obtained his B.E degree in Electronics & Communication from Bangalore University and M.E. Degree in Electronics and communication from University Visvesvaraya College of Engineering, Bangalore. He is pursuing his Ph.D. in Computer Science and Engineering at Bangalore University, Karnataka. He has over 9 research publications in refereed International Journals and Conference ProceedingsHis research interests include VLSI Architectures for Image Processing, Biometrics, VLSI for signal processing applications, Video Processing, VLSI Design and Circuit.

K B Raja is an Assistant Professor, Dept. of Electronics and Communication Engineering, University Visvesvaraya college of Engineering, Bangalore University, Bangalore. He obtained his B.E and M.E in Electronics and Communication Engineering from University Visvesvaraya College of Engineering, Bangalore. He was awarded Ph.D. in Computer Science and Engineering from Bangalore University. He has over 116 research publications in refereed International Journals and Conference Proceedings. His research interests include Image Processing, Biometrics, VLSI Signal Processing, computer networks.

K R Venugopal is currently the Principal and Dean, Faculty of Engineering, University Visvesvaraya College of Engineering, Bangalore University, Bangalore. He obtained his Bachelor of Engineering from University Visvesvaraya College of Engineering. He received his Masters degree in Computer Science and Automation from Indian Institute of Science, Bangalore. He was awarded Ph.D. in Economics from Bangalore University and Ph.D. in Computer Science from Indian Institute of Technology, Madras. He has a distinguished academic career and has degrees in Electronics, Economics, Law, Business Finance, Public Relations, Communications, Industrial Relations, Computer Science and Journalism. He has authored 27 books on Computer Science and Economics, which include Petrodollar and the World Economy, C Aptitude, Mastering C, Microprocessor Programming, Mastering C++ etc. He has been serving as the Professor and Chairman, Department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore University, Bangalore. During his three decades of service at UVCE he has over 370 research papers to his credit. His research interests include computer networks, parallel and distributed systems, digital signal processing and data mining.

L M Patnaik is the Honorary Professor, Indian Institute of Science, Bangalore, India. During the past 35 years of his service at the Indian Institute of Science, Bangalore, He has over 621 research publications in refereed International Journals and Conference Proceedings. He is a Fellow of all the four leading Science and Engineering Academies in India; Fellow of the IEEE and the Academy of Science for the Developing World. He has received twenty national and international awards; notable among them is the IEEE Technical Achievement Award for his significant contributions to high performance computing and soft computing. His areas of research interest have been parallel and distributed computing, mobile computing, CAD for VLSI circuits, soft computing, and computational neuroscience.