A Time-Area- Power Efficient Multiplier and Square Architecture Based On Ancient Indian Vedic Mathematics

Himanshu Thapliyal
Department of Computer Engineering
College of Technology
G. B. Pant University of Agril & Technology
Pantnagar, Uttaranchal -263145, India.
(thapliyalhimanshu@yahoo.com)

Hamid R. Arabnia
The University of Georgia
Department of Computer Science
415 Graduate Studies Research Center
Athens, Georgia 30602-7404, U.S.A.
(hra@csa.uga.edu)

Abstract

In this paper new multiplier and square architecture is proposed 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 design implementation is described in both at gate level and high level RTL code (behavioural level) using Verilog Hardware Description Language. The design code is tested using Veriwell Simulator. The code is synthesized in Synopys FPGA Express using: Xilinx, Family: Spartan Svq300, Speed Grade: -6. The present paper relates to the field of math coprocessors in computers and more specifically to improvement in speed and power over multiplication and square algorithm implemented in coprocessors. In FPGA implementation it has been found that the proposed Vedic multiplier and square are faster than array multiplier and Booth multiplier.

Keywords: *Vedic Mathematics*, *Multiplier*, *Array Multiplier, Square Architecture*.

1. Introduction

The need for high speed processing has been increasing as a result of expanding signal processing and computer applications. Higher throughput arithmetic operations are important to achieve the desired performance in many real time signal and image processing applications. One of the important arithmetic operations in such applications is to perform a large no of mathematical calculations in a very less time. Since in performing mathematical calculations especially multiplication a computer spends a considerable amount of its processing time, an improvement in the speed of a math coprocessor for performing multiplication will increase the overall speed of the computer. Multiplication can be implemented using several algorithms such as: array, Booth, carry save, modified Booth algorithms and Wallace tree. A number of interesting parallel and serial-parallel multiplier architectures have been proposed based on aforesaid algorithm which improve the cost-throughput efficiency [1-21].

In an array multiplier multiplication of two binary numbers can be obtained with one micro-operation by using a combinational circuit that forms the product bits all at once thus making it a fast way of multiplying two numbers since the only delay is the time for the signals to propagate through the gates that form the multiplication array. However, an array multiplier requires a large no gates and for this reason it is less economical [22].

The other aspect of improving the multiplier efficiency is through the arrangement of adders. As methods of arrangement of adders are concern, there are two methods: a carry save array (CSA) method and a Wallace tree method [23].In the CSA method, bits are processed one by one to supply a carry signal to an adder located at a one bit higher position. This is in fact much similar to a manual calculation method, where the layout thereof corresponds to the logic and is regular, and hence the design of layout is easy. The CSA method has its own limitation since an execution time depends upon the number of bits of the multiplier; there is some difficulty in achieving high speed operation [23].

In the Wallace tree method, three bit signals are passed to a one bit full adder ("3W")which is called a three input Wallace tree circuit , and the output signal (sum signal) is supplied to the next stage full adder of the same bit, and the carry output signal thereof is passed to the next stage fulladder of the same no of bit, and the carry output signal thereof is supplied to the next stage of the fulladder located at a one bit higher position. In the Wallace tree method, the circuit

layout is not easy although the speed of the operation is high since the circuit is quite irregular.

Another improvement in the multiplier is by reducing the numbers of partial products generated. The Booth recording multiplier is one such multiplier; it scans the three bits at a time to reduce the number of partial products [24]. These three bits are: the two bit from the present pair; and a third bit from the high order bit of an adjacent lower order pair. After examining each triplet of bits, the triplets are converted by Booth logic into a set of five control signals used by the adder cells in the array to control the operations performed by the adder cells.

The method of Booth recording reduces the numbers of adders and hence the delay required to produce the partial sums by examining three bits at a time. The high performance of Booth multiplier comes with the drawback of power consumption. The reason for this is the large number of adder cells (15 cells for 8 rows-120 core cells) that consume power [24]. The conclusion is that the current methodology of multiplication leads to more consumption of power and reduction in efficiency.

This paper proposes a novel multiplier and square architecture providing the solution of the aforesaid problems adopting the sutra (formula) of Vedic Mathematics called Urdhva Tiryakbhyam (Vertically and Cross wise)[25,26,27]. The designs of the multiplier and square are considerably faster than existing multipliers reported previously in the literature. It is demonstrated that this design is quite efficient in terms of silicon area/speed.

2. Multiplier and Square Architecture

2.1 The New Vedic Multiplier Architecture

The proposed multiplier is based on an algorithm Urdhva Tiryakbhyam (Vertical & Crosswise) of ancient Indian Vedic Mathematics. 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 Urdhava Triyakbhyam explained in Table-1 for 4 x 4 bit number. The algorithm can be generalized for n x n bit number. Since the partial products and their sums are calculated in parallel, the multiplier is independent of the clock frequency of the processor. Thus the multiplier will require the same amount of time to calculate the product and hence is independent of the clock frequency. The net advantage is that it reduces the need of microprocessors to operate at increasingly high clock frequencies. While a higher clock frequency generally results in increased processing power, its disadvantage is that it also increases power dissipation, resulting in higher device operating temperatures. By adopting the proposed multiplier, microprocessors designers can easily circumvent these problems to avoid catastrophic device failures [28]. The processing power of multiplier can easily be increased by increasing the input and output data bus widths since it has a quite a 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 its gate delay and area increases very slowly as compared to other multipliers. Therefore it is, time, space and power efficient. It is demonstrated that this architecture is quite efficient in terms of silicon area/speed.

2.2 Square Architecture:

In order to calculate the square of a number we have utilized “Duplex” D property of Urdhva Triyakbhyam. In the Duplex, we take twice the product of the outermost pair, and then add twice the product of the next outermost pair, and so on till no pairs are left. When there are odd number of bits in the original sequence there is one bit left by itself in the middle, and this enters as its square. Thus for 987654321,

D= 2 * (9 * 1)+2 * (8 * 2)+2 * (7 * 3)+
2 * (6 * 4)+ 5 * 5=165. Further, the Duplex can be explained as follows

For a 1 bit number D is its square.

For a 2 bit number D is twice their product

For a 3 bit number D is twice the product of the outer pair + the square of the middle bit.

For a 4 bit number D is twice the product of the outer pair + twice the product of the inner pair.

Thus D (1)= 1 * 1;
D(11)=2 * 1 * 1;
D(101)=2 * 1 * 1+0 * 0;
D(1011)=2 * 1 * 1+2 * 1 * 0;

In Table-2 the algorithm is explained for 4 x 4 bit number. The Vedic square has all the advantages of the Vedic multiplier. Further, it is quite faster and smaller than the array, Booth and proposed Vedic multiplier.

TABLE 1- 4 x 4 bit Vedic multiplier Using Urdhva Tiryakbhyam

CP- Cross Product (Vertically and Crosswise)

				X3	X2	X1	X0	Multiplicand
				Y3	Y2	Y1	Y0	Multiplier
H	G	F	E	D	C	B	A	
P7	P6	P5	P4	P3	P2	P1	P0	Product

PARALLEL COMPUTATION

1. CP X0 = X0 * Y0 = A
   Y0

2. CP X1 X0 = X1 * Y0+X0 * Y1= B
   Y1 Y0

3 CP X2 X1 X0 = X2 * Y0 +X0 * Y2 +X1 * Y1=C
   Y2 Y1 Y0

4 CP X3 X2 X1 X0 = X3 * Y0 +X0 * Y3+X2 * Y1 +X1 * Y2=D
   Y3 Y2 Y1 Y0

5 CP X3 X2 X1 = X3 * Y1+X1 * Y3+X2 * Y2=E
   Y3 Y2 Y1

6 CP X3 X2 = X3 * Y2+X2 * Y3=F
   Y3 Y2

7 CP X3 = X3 * Y3 =G
   Y3

METHODOLOGY

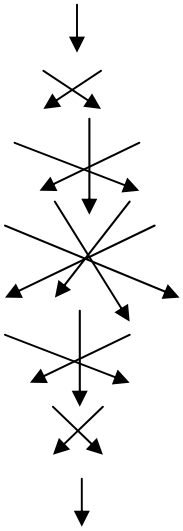


TABLE 2- 4 x 4 bit Square Using Duplex of Urdhva Tiryakbhyam

D- Duplex

				X3	X2	X1	X0	Multiplicand
				X3	X2	X1	X0	Multiplier
H	G	F	E	D	C	B	A	
P7	P6	P5	P4	P3	P2	P1	P0	Product

PARALLEL COMPUTATION

1. D = X 0 * X0 =A

2. D = 2 * X1 * X0=B

3. D = 2 * X2 * X0+ X1 * X1=C

4. D = 2 * X3 * X0+2 * X2 * X1=D

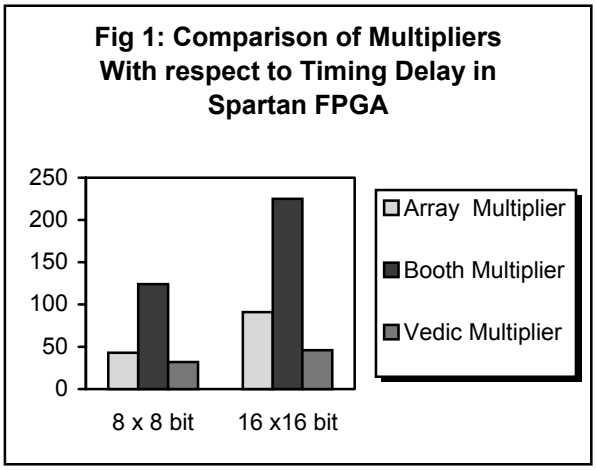
5 D = 2 * X3 * X1 + X 2 * X2 =E

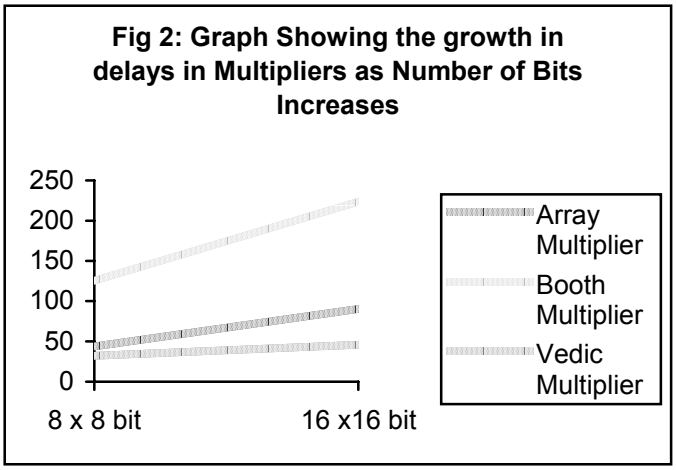
6. D = 2 * X3 * X2 =F

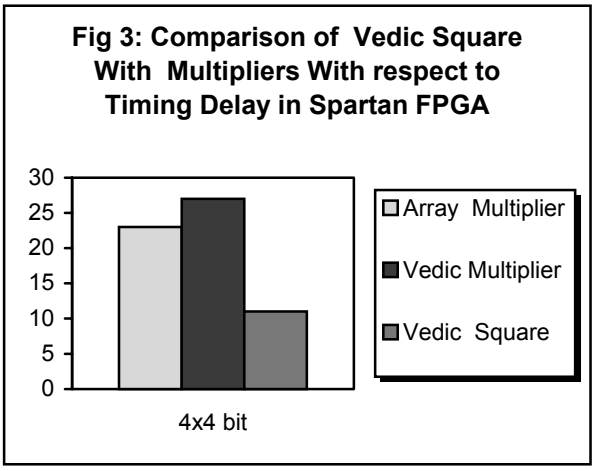
7. D = X3 * X3 =G

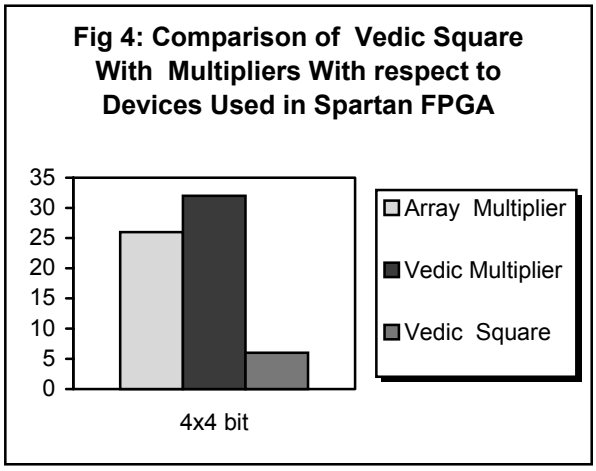
Table 3: Timing Simulations (Comparisons of the Multipliers)

Name of Multiplier		Array Multiplier		Booth Multiplier		Vedic Multiplier	
		8 X8 Bit	16 x16 Bit	8 x8 Bit	16 x16 Bit	8x 8 Bit	16 x16 bit
XILINX: Spartan S30VQ100 :-4	FMAP	150	491	283	1097	150	621
	HMAP	10	36	49	203	0	0
	DELAY	43	91	124	225	32	46
XILINX: Spartan-XL: S05XLPC8 :4	FMAP	150	491	283	1097	32	621
	HMAP	10	36	49	203	0	0
	DELAY	44	94	125	227	32	46
XILINX : VIRTEX: V300PQ240 :-6	LUT	132	511	283	1021	119	511
	DELAY	48	96	162	162	52	80









3. Verification and Implementation

In this study, the algorithms are implemented in Verilog HDL and logic simulation is done in Veriwell Simulator; the synthesis and FPGA implementation is done using Synopsys FPGA Express. After gate-level synthesis from high level behavioral and/or structural RTL HDL codes, basic schematics are optimized as our designed algorithmic approaches. The designs are optimized for speed and area using Xilinx families. The families used for testing are as follows

1. XILINX:SPARTAN:S30VQ100:-4
2. XILINX:SPARTAN:XL:S05XLPC8:4
3. XILINX : VIRTEX: V300PQ240:-6

4. Results and Discussions

The result obtained from proposed Vedic multiplier is faster than array multiplier and Booth multiplier. The result is grouped in Table-3 for 8x 8 and 16 x16 bit numbers. It has been found that as the number of bits increases from 8x8 bits to 16x16 bits the timing delay is greatly reduced for Vedic multiplier as compared to other multipliers. For the Xilinx, Spartan family it has been found that the gate delay in Vedic multiplier for 16x16 bit number is 46 ns while it is 91ns for array multiplier and 225 ns for Booth multiplier. The numbers of HMAP & FMAP are 527 for array multiplier, 621 for Vedic multiplier and 1300 for Booth multiplier. A comparison histogram and graph of gate delays of the multipliers for 8x8 bits, 16x16 bits is given in figure-1 and figure-2 respectively. The result shows that as the number of bits in the multiplier increases the Vedic multiplier takes the dominating role over array multiplier & Booth multiplier as shown by histogram and graph. If the bits in the multipliers are continuously increased to N X N (where N can be any number) bits the Vedic multiplier has the greatest advantage as compared to other multipliers over gate delays and regularity of structures. The results obtained by synthesizing the square algorithm of Vedic Mathematics are also quite encouraging. For Spartan family it has been found that the gate delay in Vedic square for 4 x 4 bit number is 11 ns while the gate delay in Vedic and array Multiplier are 27ns and 23 ns respectively. The area i.e the number of devices in Vedic square are 11 while in Vedic and array they are 33 and 26 respectively. Figure 3 and figure 4 shows the comparison of Vedic square with the multipliers. Thus the result shows that the proposed Vedic square is smallest and the fastest of the reviewed architectures.

5. Conclusion

It can be concluded that Vedic multiplier and square is faster than array multiplier and Booth multiplier. Due to factors of timing efficiency, speed and lesser area the proposed Vedic multiplier and square can be implemented in Arithmetic and Logical Units replacing the traditional multipliers and squares based on array and Booth multiplication. In summary, embodiments of the invention provide a design for a flexible multiplier and square which can be considerably faster than existing multipliers and squares reported previously in the literature. The speed improvements are gained by parallelizing the generation of partial products with their concurrent summations. It is demonstrated that this design is quite efficient in terms of silicon area/speed. Such a design should enable substantial savings of resources in the FPGA when used for image/video processing applications.

References

[1]. Ciminiera, L, and Valenzano, A. :" Low cost Serial Multiplier of High Speed Specialised Processors", IEE Proc. E. 1988,135,(5), pp-259-265.

[2]. Wey,C.L., and Chang, T.Y.: " Design and analysis of VlSI Based Parallel Multipliers",IEE Proc.E.1990,137,(4), pp. 328-336.

[3]. Chen, L-N., and Willoner, R.; “An O (n) Parallel Multiplier With Bit Sequential Input and Output", IEEE Trans, 1979, COM-28, pp, 721-727.

[4]. Gnanasekran, R. : " A Fast Serial- Parallel Binary Multiplier", IEEE Trans., 1985, COM-34,pp. 741-744.

[5]. Ait-Boudaoud, D., Ibrahim, M.K., and Hayes-Gill,B.R.:“Novel Pipelined Serial/Parallel Multiplier", Electron. Lett., 1990,26,pp, 582-583.

[6]. Santoro, M.R., and Horowitz., M.: "Spim : A Pipelined 64 x64 -Bit Iterative Multiplier", IEEE J. Solid-State Circuits,1989,24,pp.487-493.

[7]. Smith,S.G., and Denyer, P,B.: " Radix-4 Modules For High Performance Bit Serial Computations", IEE Proc . E, 1987,134,pp-271-276.

[8].Primlani, K.K., and Meador, J.L.: " A Nonredundant-Radix-4 Serial Multiplier", IEEE J. Solid- State Circuits,1989,24,pp,1729-1735.

[9]. Booth, A.D.:" A Signed Binary Multiplication Technique". Q. J. Mechan. Appl. Math., 1951,4,(2),pp.236-240.

[10].Sam. H., and Gubta, A.: " A generalised Multibit Recoding of 2's Complement Binary Numbers And Its Proof With Application In Multiplier Implementations", IEEE Trans., 1990,c-139,(80,pp,1006-1015.

[11].Ercegovac, M.D: " On-Line Arithmetic: An Overview ',SPIE. 1984, 495,pp,86-93.

[12]. Ercegovac, M.D, and Lang,T.: " Fast Multipication Without Carry Propagate Addition", IEEE Trans., 1990,c-139,(11),pp,1385-1390.
[13].Irwin, M.J., and Owens, R.M,:" Digit Pipelined Arithmetic As Ilustrated By the Paste-Up System: A Tutorial",IEEE Computer J., 1987,pp,61-73.
[14].Irwin, M.J., and Owens, R.M,:" Design Issues In Digit Serial Signal Processors", Proceedings of the IEEE International Symposium On Circuits and Sysems,1989,pp,441-444.
[15]. Hartley,R., and Corbett,P.: "Digit-Serial Processing Techniques",IEEE Trans,1990,C8-37,(60,pp,707-719.
[16].Parhi, K.K.:"A Systematic Approach For Design of Digit Serial Signal Processing Architectures", IEEE Trans., 1991, CS-38,(4),pp,358-375.
[17]. Ciminiera,L., and Serra,A.: " Arithmetic Array For Fast Inner Product Evaluation", Proceedings of Sixth Symposium On Computer Arithmetic,1983,pp,61-66.
[18].. De Mori,R., and Cardin, R.:" Iterative Parallel Multipliers Based On Multiplexers", Signal Processing,1984,6,pp,213-223.
[19]. De Mori,R., and Cardin, R.:" A Recursive Algorithm For Binary Multiplication And Its Implementation", ACM Trans. Comput. Syst.,1985,3,(4),pp,294-314..
[20] Vishal Verma and Himanshu Thapliyal , " High Speed Efficient N X N Bit Multiplier Based On Ancient Indian Vedic Mathematics", Proceedings of the 2003 International Conference On VLSI(VLSI03), Las Vegas Nevada , June 2003.
[21] Himanshu Thapliyal and Vishal Verma , " High Speed Efficient Signed/Unsigned N X N Bit Multiplier Based On Ancient Indian Vedic Mathematics" proceedings of the 7th IEEE VLSI Design & Test Workshop, Bangalore, August 2003.
[22] Morris Mano, "Computer System Architecture", PP-346-347, 3rd edition, PHI.1993.
[23] Gensuke Goto,"High Speed Digital Parallel Multiplier", United States Patent-5,465,226, November 7 1995.
[24] Tam Anh Chu, "Booth Multiplier with Low Power High Performance Input Circuitry", US Patent,, 6,393,454 B1, May 21 2002.
[25] Jagadguru Swami Sri Bharath, Krsna Tirathji, "Vedic Mathematics or Sixteen Simple Sutras From The Vedas", Motilal Banarsidas , Varanasi(India),1986.
[26] A.P. Nicholas, K.R Williams, J. Pickles , "Application of Urdhava Sutra", Spiritual Study Group, Roorkee (India),1984
[27] A.P. Nicholas, K.R Williams, J. Pickles, " Lectures on Vedic Mathematics", Spiritual Study Group, Roorkee (India),1982.
[28]. Beiu, "Microprocessor and a digital signal processor including adder and multiplier circuits employing logic gates having discrete and weighted inputs", United States Patent, 6,516,331, February 4, 2003.