Implementation of High Speed Multiplier on FPGA

Dilip J Udhani, Department of Instrumentation and Control Engineering, L.D. College of Engineering , Ahmedabad, India.

Associate Prof. R. C. Patel, Department of Instrumentation and Control Engineering, L. D. College of Engineering, Ahmedabad, India.

Abstract— **The purpose of this paper is to design a high speed multiplier as Faster additions and multiplications are of extreme importance in DSP (Digital Signal Processing) for convolution, discrete Fourier transform, digital filters, etc. Therefore, DSP engineers are constantly looking for new algorithms and hardware to implement them. The multiplier is based on vedic mathematics. The algorithm is designed using "Urdhva Tiryagbhyam Sutra" which means vertically and cross wise. Among the various methods of multiplication in Vedic mathematics, Urdhva tiryagbhyam, being a general multiplication formula, is equally applicable to all cases of multiplication. This is more efficient in the multiplication of large numbers with respect to speed and area. Different schemes of multiplier for 4x4 bit and 8x8 bit multiplier are discussed.**

Index Terms— **Terms—FPGA, Vedic Mathematics, Urdhva Tiryagbhyam**

I. Introduction

The requirement for high speed processing has been increased because of newer computer applications and to achieve the desired performance in many real time signal and image processing applications, higher throughput arithmetic operations are important. Instead of having more consuming processing time system, we have proposed Urdhva Triyagbhyam Vedic method for arithmetic operations which perform a large no of mathematical calculations in a very less time. It increases the overall speed of the different electronics devices. Digital multipliers are the center components of all the digital signal processors (DSPs) and the speed of the DSP is mostly determined by the speed of its multipliers. They are necessary in the implementation of computation systems like Fast Fourier transforms (FFTs) and multiply accumulate (MAC).

In this paper, we have proposed a new multiplication algorithm which avoids the need of large multipliers by reducing the large number to the smaller number multiplications count which reduces the propagation delay linked with the conventional large multipliers significantly. The structure of the proposed algorithm is based on the Urdhava Tiryagbhyam Sutra (formula) of Vedic mathematics which is simply means: "vertical and crosswise multiplication". The procedure of multiplication using the Urdhava Tiryagbhyam involves minimum calculations, which in turn will lead to reduced number of steps in computation, reducing the space, saving more time for computation. Hence it optimizes to take full advantage of reduction in the number of bits in multiplication. Although Urdhava Tiryagbhyam is applicable to all cases of multiplication, it is more efficient when the numbers involved are large.

II. VARIOUS CONVENTIONAL MULTIPLIERS

The need for high speed processing has been increasing as a result of expanding signal processing and computer 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. 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 . 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 .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 . 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 full adder of the same no of bit, and the carry output signal thereof is supplied to the next stage of the full adder 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 . 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 . 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 architecture providing the solution of the aforesaid problems adopting the sutra (formula) of Vedic Mathematics called Urdhva Tiryagbhyam (Vertically and Cross wise). The designs of the multiplier is 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.

III. VEDIM MATHEMATICS

We appreciate the efforts put by Jagadguru Swami Sri Bharati Krishna Tirthaji Maharaja to introduceVedic Mathematics and also acknowledge the work of various people regarding Vedic Mathematics as the Vedic mathematics approach is totally different and considered very close to the way a human mind works.

The multiplication of numbers is utilized in almost all branches of engineering; therefore the demand for high efficiency multiplier architecture increases. Vedic mathematics is based on sixteen sutras which serve as somewhat cryptic instructions for dealing with Different mathematical problems. Below is a list of those sixteen sutras, translated from Sanskrit into English.

These Sutras along with their brief meanings are enlisted below alphabetically.

1) (Anurupye) Shunyamanyat – If one is in ratio, the other is zero.

2) Chalana-Kalanabyham – Differences and Similarities.

3) Ekadhikina Purvena –By one more than the previous one.

4) Ekanyunena Purvena – By one less than the previous one.

5) Gunakasamuchyah – The factors of the sum is equal to the sum of the factors.

6) Gunitasamuchyah – The product of the sum is equal to the sum of product.

7) Nikhilam Navatashcaramam Dashatah – All from 9 and the last from 10.

8) Paraavartya Yojayet – Transpose and adjust.

9) Puranapuranabyham – By the completion or Non-completion.

10) Sankalana-vyavakalanabhyam – By addition and by subtraction.

11) Shesanyankena Charamena – The remainders by the last digit.

12) Shunyam Saamyasamuccaye – When the sum is the same that sum is zero.

13) Sopaantyadvayamantyam – The ultimate and twice the penultimate.

14) Urdhva-Tiryagbyham – Vertically and crosswise.

15)Vyashtisamanstih – Part and Whole.

16) Yaavadunam – Whatever the extent of its deficiency.

IV. URDHVA TIRYAGBHYAM SUTRA

Urdhva tiryagbhyam Sutra is a general multiplication formula applicable to all cases of multiplication. It literally means "Verti-cally and crosswise". To illustrate this multiplication scheme, let us consider the multiplication of two decimal numbers (325 * 728). Line diagram for the multiplication is shown in Figure.1. The digits on the two ends of the line are multiplied and the result is added with the previous carry. When there are more lines in one step, all the results are added to the previous carry. The least significant digit of the number thus obtained acts as one of the result digits and the rest act as the carry for the next step. Initially the carry is taken to be zero.

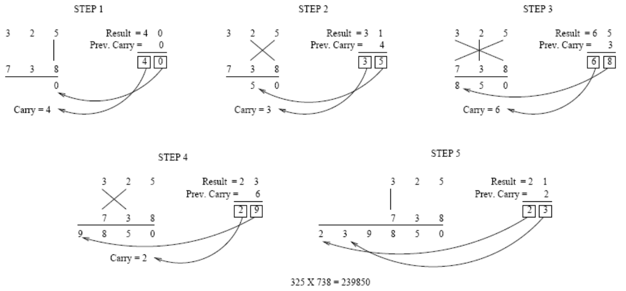

Fig. 1. Multiplication of two decimal numbers by Urdhva Triyagbhyam Sutra

V. VARIOUS MULTIPLIER TECHNIQUES

In this paper the multiplier architecture for different schemes is implemented in verilog and the FPGA synthesis is done using Xilinx ISE 13.2. The design is optimized for speed using Xilinx device family : Spartan 3E XC3S400, package pq208 speed grade -4. The different multiplier schemes for 4x4 bit and 8x8 bit vedic multiplier are shown in

ISSN: 2278 – 7798

the table. For the designing of different multiplier schemes we have used the architectures given below.

Block diagram of a new type of 4 bit adder is shown in fig.2 [1].A,B,C,D are four inputs; C0 and C1 are the LSB and the MSB of carry outputs respectively and Sum is the sum of four inputs. Boolean expressions for this adder are shown below.

$$Sum = \sum m(1,2,4,7,8,11,13,14) = A \oplus B \oplus C \oplus D$$

$$C_0 = \sum m(3,5,6,7,9,10,11,12,13,14) = \overline{B}D + C\overline{D} + B\overline{C}$$

$$C_1 = \sum m(15) = ABCD$$

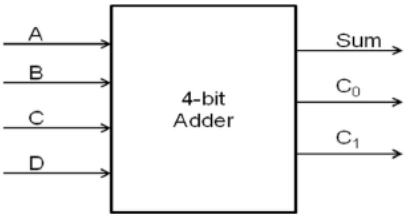

Fig. 2. 4 bit adder

Block diagram of 4x4 bit multipliers a new type of 4 bit adder, with RCA(Ripple Carry Adder) and CLA(Carry Look ahead Adder) are shown in fig. 3.[1], in fig. 4[4]. And in fig.5[4]. Respectively.

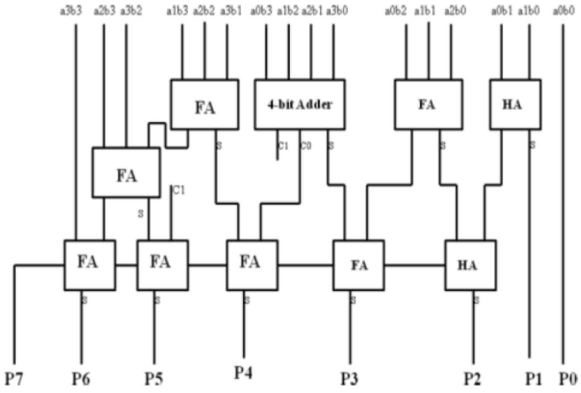
a3b3 a2b3 a3b2 a1b3 a2b2 a3b1 a0b3 a1b2 a2b1 a3b0 a0b2 a1b1 a2b0 a0b1 a1b0 a0b0
FA
4-bit Adder
FA
HA
FA
C1
C1 C0
FA
FA
FA
FA
HA
P7
P6
P5
P4
P3
P2
P1
P0

Fig. 3. Hardware architecture of Urdhva Triyaghbhyam 4x4 mulitplier using 4 bit adder

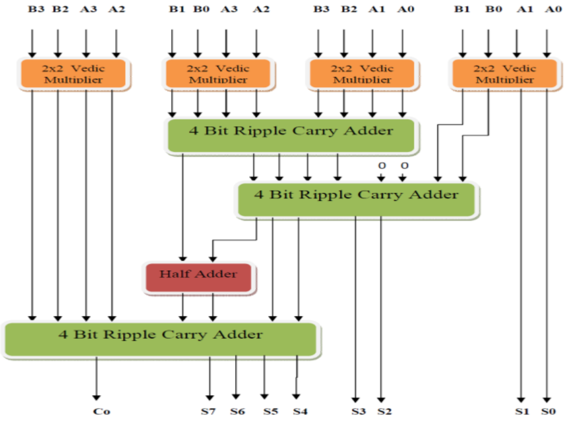

Fig. 4. Block diagram for 4x4 bit vedic multiplier using ripple carry adder

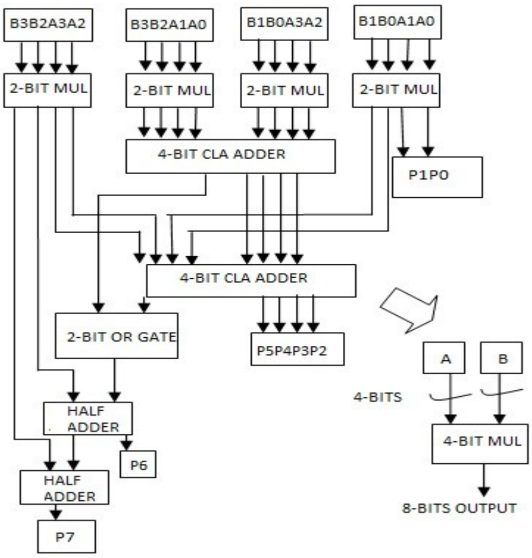

Fig. 5. Block diagram for 4x4 bit vedic multiplier

Block diagram of 8x8 bit multipliers with RCA(Ripple Carry Adder) and CLA(Carry Look ahead Adder) are shown in fig. 6[3] and fig. 7[3] respectively.

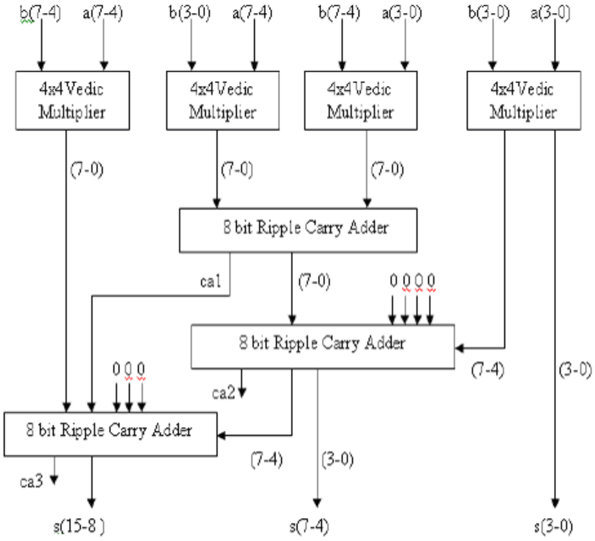

Fig. 6. Block diagram for 8x8 bit vedic multiplier

ISSN: 2278 – 7798

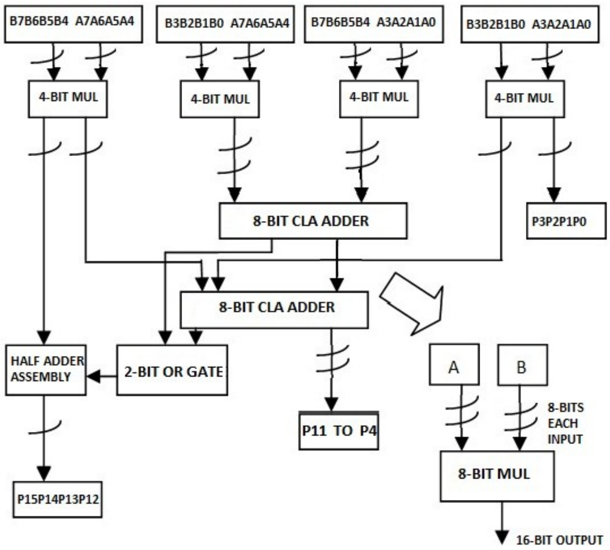

Fig. 7. Block diagram for 8x8 bit vedic multiplier

The combinational path delay for different multiplier scheme is given in the following table.

Sr No	Type of Multiplier	Scheme	Delay using vedic multiplier
1	4x4 bit Multiplier	Using a new 4 bit adder	17.711
2	4x4 bit Multiplier	Using 4 bit RCA	22.309
3	4x4 bit Multiplier	Using 4 bit CLA	20.511
4	4x4 bit Multiplier	4x4 -> 2x2 -> And logic and half adder	19.907
5	4x4 bit Multiplier	4x4 -> 2x2 -> operator (*)	18.232
6	8x8 bit Multiplier	8x8 -> 4x4 [A new 4 bit adder], 8 bit CLA	30.431
7	8x8 bit Multiplier	8x8 -> 4x4 -> 2x2 , 8 bit CLA	32.026
8	8x8 bit Multiplier	8x8 -> 4x4 [A new 4 bit adder] , 8 bit RCA	31.239
9	8x8 bit Multiplier	8x8 -> 4x4 -> 2x2 , 8 bit RCA	33.664
10	8x8 bit Multiplier	8x8 -> 4x4 -> 2x2 -> and logic and half adder	31.780

VI. CONCLUSION

For 4x4 bit multiplier the scheme 1(A new 4 bit adder) is the best one among all the different schemes as given in the table. For 4x4 bit multiplier it is also concluded that using multiplication operator (*) for 2x2 bit multiplier will give good results. For 4x4 bit multiplier it is also concluded that the multiplier with CLA is faster than the multiplier with RCA.

For 8x8 bit multiplier the scheme no.7(using * operator for 4x4 bit multiplier) is the best one among all the different schemes as given in the table. For 8x8 bit multiplier it is also concluded that the multiplier with CLA is faster than the multiplier with RCA. If we have to design fast multiplier then we have to find a fast adder or we have to find a scheme like scheme no. 7(using * operator for 4x4 bit multiplier) as given in the table for 8x8 bit multiplier.

REFERENCES

[1] Krishnaveni D., Umarani T.G. "Vlsi implementation of vedic multiplier with reduced delay", International Journal of Advanced Technology & Engineering Research (IJATER) National Conference on Emerging Trends in Technology (NCET-Tech) , ISSN No: 2250-3536 Volume 2, Issue 4, July 2012.

[2] Virendra Babanrao Magar, "Intelligent and superior vedic multiplier for fpga based arithmetic circuits " International Journal of Soft Computing and Engineering (IJSCE) ISSN: 22312307, Volume-3, Issue-3, July 2013 .

[3] R.K.Bathija, R.S.Meena, S. Sarkar, Rajesh Shah Tinjrit, "Low power high speed 16*16 bit multiplier using vedic mathematics",International journal of computer Application, volume 59, december 2012.

[4] Gaurav Sharma , Arjun Singh Chauhan ,Himanshu Joshi, Satish Kumar Alaria , "Delay Comparison of 4 by 4 Vedic Multiplier based on Different Adder Architectures using VHDL" International Journal of IT, Engineering and Applied Sciences Research (IJIEASR) ISSN: 2319-4413 Volume 2, No. 6, June 2013 .

[5] Asmita Haveliya "A Novel Design for High Speed Multiplier for Digital Signal Processing Applications(Ancient Indian Vedic mathematics approach)" International Journal of Technology And Engineering System(IJTES): Jan – March 2011- Vol2 .No1.

[6] Himanshu Thapliyal, Hamid R Arbania "A time area power Efficient multiplier and square architecture based on ancient indian vedic mathematics"

[7] Parth Mehta, Dhanashri Gawali. "Conventional versus vedic mathematical method for Hardware implementation of a multiplier" International conference on advances in computing, control and telecommunication technologies 2009.

[8] Prof. J M Rudagi, Vishwanath Ambli. "Design and implementation of efficient multiplier using vedic mathematics" International conference on advances in recent technologies in communication and computing 2011.

[9] G Vaithiyanathan, K Venkatesan "Simulation and implementaion of vedic multiplier using VHDL code" International journal of scientific and engineering research volume 4, issue 1, Jan 2013.

[10] Badal sharma,"Design and hardware implementation of 128 bit vedic multiplier" International journal for advance research in engineering and technology. Vol1 issue V Jun 2013.

[11] vinay Kumar, Mr. Arun Kumar Chatterjee, "Analysis, verification and fpga implementation of vedic multiplier with bist capability"Department of Electronics and Communication Engineering Thapar University June – 2009