International Journal of Innovative Research in Advanced Engineering (IJIRAE)
Volume 1 Issue 1 (April 2014)

Performance Evaluation of High Speed Complex Multiplier Using Vedic Mathematics

Rajashri Bhongade
P.C.E,Nagpur
R.T.M.Nagpur University
rajshree.bhongade@gmail.com

S.G.Mungale
P.C.E,Nagpur
R.T.M.Nagpur University
sharda_27@rediffmail.com

Karuna Bogavar
P.C.E,Nagpur
R.T.M.Nagpur University
karuved@rediffmail.com

Abstract*— Speed of multiplier determines the speed of the DSP. In this paper VHDL implementation of complex number multiplier using ancient Vedic mathematic. By using "Vedic Mathematic "concept can skip carry propagation delay. The "Urdhva Tiryakbhyam" sutra was selected for implementation since it is applicable to all cases of multiplication. Meaning of "Urdhva Tiryakbhyam" is vertically and crosswise. The partial product and sum are generated in single step which reduce carry propagation from LSB to MSB. The important feature of this project is to improve the speed of complex multiplier using Vedic mathematics. The implementation of Vedic mathematics and their application to complex multiplier ensure reduction of propagation delay. The coding is done for complex multiplier using VHDL and synthesized using Xilinx ISE version 9.1i. Vedic complex multiplier can bring great improvement in the DSP performance. The result clearly indicates that Urdhava Tiryakabham can have great impact on improving speed of Digital Signal Processors.

Keywords*— Vedic multiplier, Urdhava tiryakbhyam, VHDL, DSP.

I. INTRODUCTION

Vedic system of computation covers all form of mathematics; it may be geometry, trigonometry or algebra. This makes it the easiest and fastest way to perform any mathematical calculation. Our focus in this work is to develop complex multiplication operation which is most frequently used operation in signal processing application like many high performance systems such as FIR filters, microprocessor, digital signal processor, etc. Complex number multiplication is important in Digital single processing(DSP),specially in DIT-FFT twiddle factor multiplied with input is complex number, Image processing(IP).To implement Discrete Fourier Transform(DFT),Discrete Cosine Transform(DCT),Fast Fourier Transform(FFT) and wireless communication imaging, complex multiplier are required. Complex numbers mostly depend on extensive number of multiplication. Four real number multiplication and two additions or subtractions are involve in complex number multiplication .Multiplication is done by AND operation & addition is done by OR operation. Carry needs to be propagated from the least significant bit (LSB) to most significant bit (MSB) when binary partial products are added in real number multiplication.

Vedic mathematic is extracted from four Vedas, which is an upya-veda of Atharva-veda.Vedic Mathematics is based on 16-sutras and 16-sub sutras invented in (1884-1960).In Vedic mathematics there are sutras Nikhilam Navatascaraman Dasatah and Urdhva Tiryakbhyam used for multiplication[1]. Our targeted Vedic sutra (algorithm) which is suitable for all cases of multiplication is Urdhva Tiryakbhyam. In this paper high speed complex multiplier is design using Vedic mathematics. In this work, we try to present complex multiplication operations and the implementation of these using Vedic mathematical methods in VHDL language [1].

II. VEDIC MULTIPLICATION METHOD

The Vedic Mathematics is an ancient mathematic invented by Jagadguru Shankaracharya Bharati Krishna Teerthaji Maharaja who was Sanskrit scholar, mathematician and philosopher. He was organized and classified the whole of Vedic Mathematics into 16 Formulae called sutra and 16 sub-sutra. The proposed complex number multiplier is based on the Urdhva Tiryakbhyam sutra. In this work the same ideas are applied to binary number system as well as decimal number system, to make the proposed algorithm compatible with the digital hardware.

Vedic multiplication based on Urdhava Tiryakbhyam sutra is discussed below.

A. *Urdhva Tiryakbhyam sutra*

The basic meaning of Sanskrit word is "Vertically and crosswise". It is applicable to all multiplication. The vertical and crosswise multiplication can be implemented starting either from left hand side or from right hand side [7][8]. Significance of vertically is straight above multiplication and significance of crosswise is diagonal multiplication and add them. In "Urdhva Tiryakbhyam sutra" multiplication is done in single line manner, implies the increase in speed by reducing propagation delay [9][12]. The literally means "Vertically and crosswise". It is based on a novel concept through which the all partial products are generated concurrently. Since the partial products and their sums are calculated in parallel. "Urdhva Tiryakbhyam sutra" to binary number system knowledge that multiplication of two bit is done by AND operation and it is implemented by using AND operation. General multiplication procedure using "Urdhva Tiryakbhyam sutra" is illustrated below.

The below figure shows multiplication of (4x4) bit number, performance is starting from right hand side. Corresponding expression as follows:

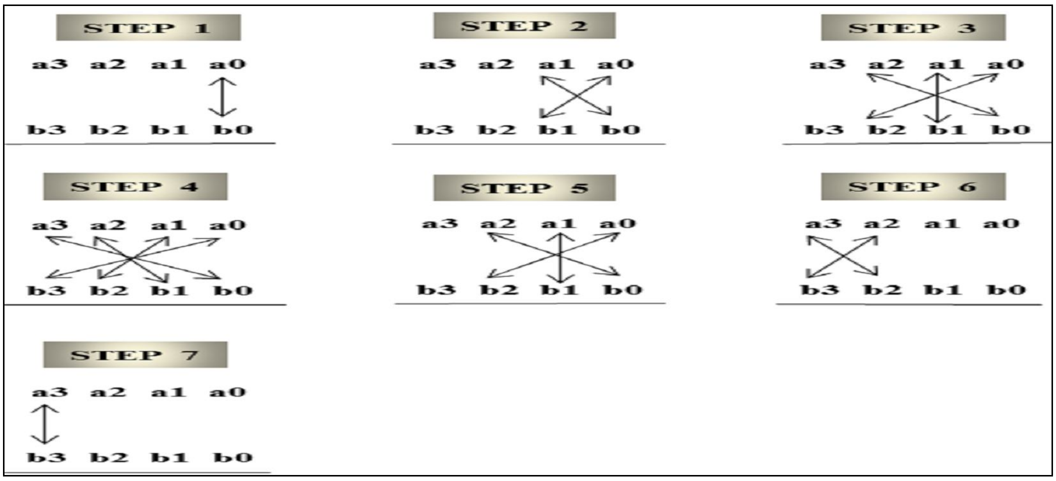

Figure 2.1: Line diagram for multiplication of two 4 - bit numbers.

Firstly, least significant bit(LSB) of the multiplier is multi lied with the least significant bit of the multiplicand (vertical). Then, the LSB of the multiplicand is multiplied with the next higher bit of the multiplier and added with the product of LSB of multiplier and next higher bit of the multiplicand (crosswise). The sum gives second bit of the product and the carry is added in the output of next stage sum obtained by the crosswise and vertical multiplication and addition of three bits of the two numbers from least significant position. Next, all the four bits are processed with crosswise multiplication and addition to give the sum and carry. The sum is the corresponding bit of the product and the carry is again added to the next stage multiplication and addition of three bits except the LSB. The same operation continues until the multiplication of the two MSBs to give the MSB of the product.

Thus we get the following expressions for every step in fig.2.1:

$$r0=a0b0 \quad (1)$$

$$c1r1=a1b0+a0b1 \quad (2)$$

$$c2r2=c1+a2b0+a1b1 + a0b2 \quad (3)$$

$$c3r3=c2+a3b0+a2b1 + a1b2 + a0b3 \quad (4)$$

$$c4r4=c3+a3b1+a2b2 + a1b3 \quad (5)$$

$$c5r5=c4+a3b2+a2b3 \quad (6)$$

$$c6r6=c5+a3b3 \quad (7)$$

With c6r6r5r4r3r2r1r0 being the final product [3] [4] [8].

The Hardware architecture realization of multiplier of Urdhava Tiryakbham shown in figure 2.All the partial product are calculated parallel and delay is associated to time taken by carry to propagate through adder. In order to multiply two 4bit numbers using 2bit multiplier we proceed as follows. Consider two 4bit numbers denoted as AHAL and BHBL where AH and BH corresponds to the most significant 2bits, AL and BL are the least significant 2bits of 4bit number. When the numbers are multiplied according to Urdhava Tiryakbham method, we get,

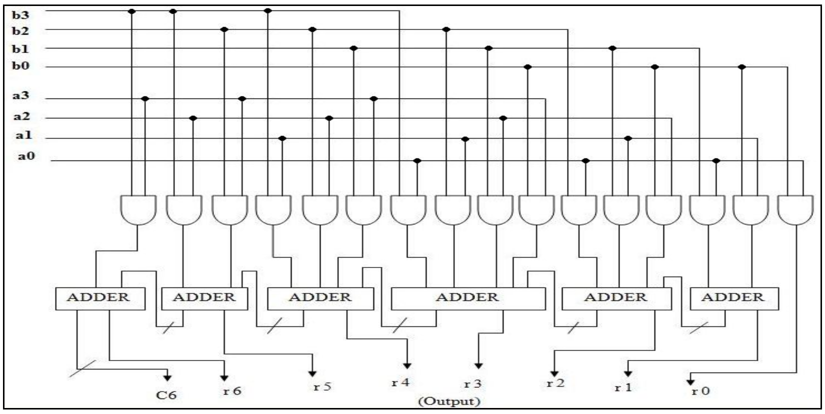

Figure 2.2: Hardware architecture of the Urdhava tiryakbhyam multiplier. [6]

When the numbers are multiplied according to Urdhava Tiryakbham method, we get,

AH AL
BH BL

(AH x BH) + (AH x BL + BH x AL) + (AL x BL).

Thus we need four 2bit multipliers and two adders to add the partial products and 2bit intermediate carry generated. Since product of 2x2 bit multiplier is 4bits long and two 4bit adders with 2bit carry. Similarly, 8bit multiplier has 4x4 multiplier and so on.

Example 1: Multiplication of binary number “1011 and 1101”, is using Urdhva Tiryakbhyam sutra.

Step 1: Divide four bit in two bit, as shown below.

AH	AL
1 0	1 1
1 1	0 1
BH	BL

Step 2: Starting at right hand most significant bit, multiply vertically i.e. AL x BL.

AL = 1 1
BL = 0 1

Again apply Urdhva Tiryakbhyam sutra

I) Vertical multiplication of right hand significant bit. (1*1=1)

II) Crosswise multiplication of bits and taking their sum. (1*1 + 0*1 = 1)

III) Vertical multiplication of left hand significant bit. (1*0=0) **AL* BL = 011**

Step 3: Next apply crosswise multiplication and add partial product i.e. (AH x BL + AL x BH)

I) repeat the step occurred in step 2.

II) AH x BL = 010.

III) AL x BH = 1001. **AH x BL + AL x BH = 1011**

Step 4: Finally add the right bit vertically as illustrated i.e. AH x BH. **AH x BH = 110**

Step 5**:** Finally if the number is (4x4) bit, for final result shift the bits by two. For (8x8) bit, shift the bit by four and so on

AL x BL = 0 1 1
AH x BL + AL x BH = 1 0 1 1
AH x BH = 1 1 0

A x B 1 0 0 0 1 1 1 1

Result of multiplication (1011 x 1101=10001111).

III. COMPLEX MULTIPLIER

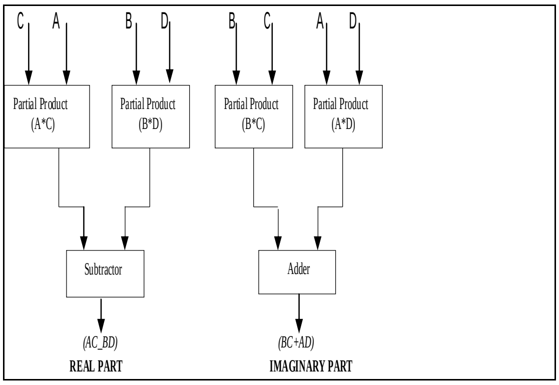

Fig 3.1 Block diagram of complex multiplier.

Complex number consist of two component knows as REAL PART(R) and IMAGINARY PART(I).

R + j I = (a + j b) (c + j d) (8)

(a + j b) is first complex number,

(c + j d) is second complex number,

From equation(8) gives two separate final result to calculate real and imaginary part .The real part of the output can be computed using (ac-bd), and the imaginary part of the result can be computed using (bc+ad). Thus four separate multiplications and addition/subtraction are required to produce the real as well as imaginary part numbers [9] [10] Multiplication is performed by using "Urdhva Tiryakbhyam sutra".

Complex Multiplication Algorithm :(a+jb) (c+jd) = R+jI.

<Input>:

A and B:

A = (a+jb) and B = (c+jd) both are complex number. (a+jb) and (c+jd) both are inputs.

Real part of A = a;
Imaginary part of A = b;
Real part of B = c;
Imaginary part of B = d;

<Output>:

Result: R and I are the real and imaginary part of complex number.

IV. DESIGN AND IMPLEMENTATION

The Vedic Complex Multiplier is implemented using VHDL implemented. The entire code is completely synthesizable. The synthesis is done using Xilinx Synthesis Tool (XST) available with Xilinx ISE 9.1i simulator. The design is optimized for speed and area using Xilinx Table 1 indicate the area consumed and speed taken to do the implementation for (4 x 4)bit,(8 x 8)bit,(16 x 16)bit Vedic complex. Figure 4.1, 4.2, 4.3 indicates the simulation waveform for (4 x 4) bit,(8 x 8)bit,(16 x 16)bit Vedic complex multiplier.

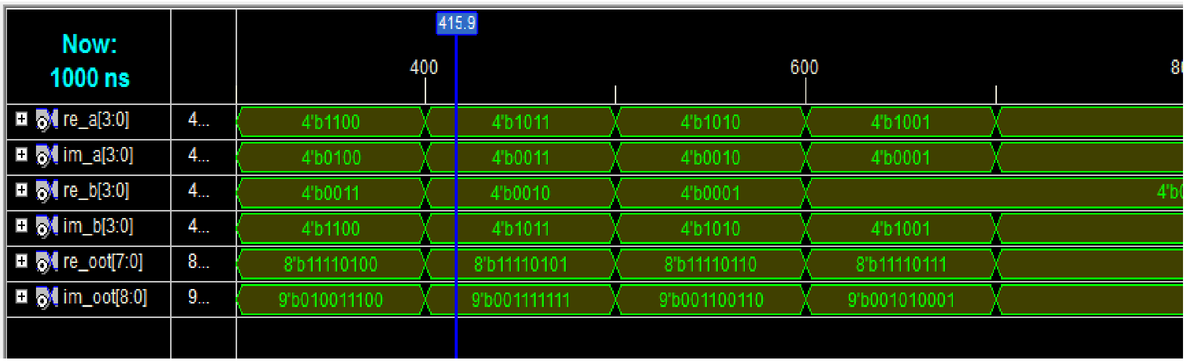

Figure 4.1: Simulation waveforms for 4-bit complex multiplication

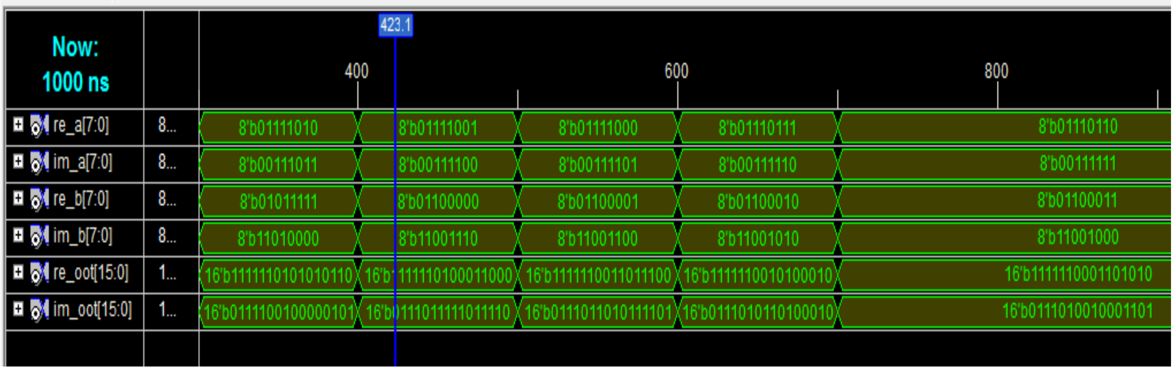

Figure 4.2: Simulation waveforms for 8-bit complex multiplication

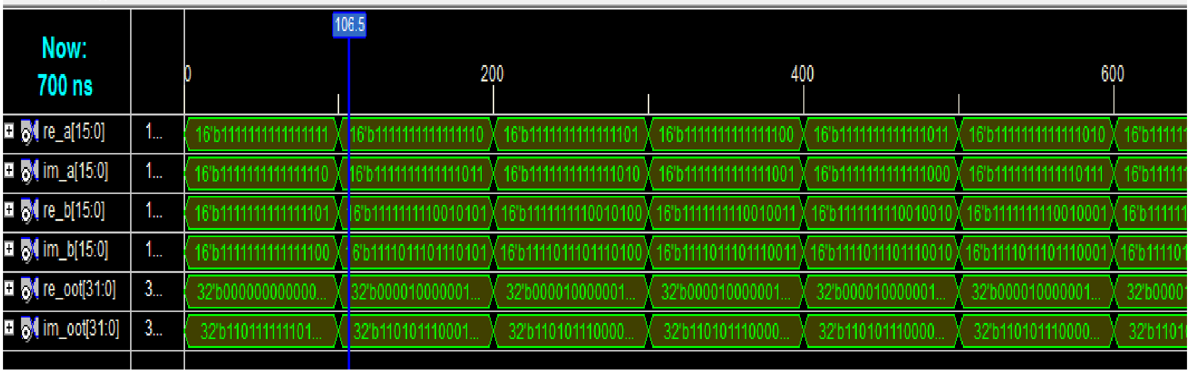

Figure 4.3: Simulation waveforms for 16-bit complex multiplication

V. EXPERIMENTAL RESULT

The work presented in this paper was implemented using VHDL and logic simulation was done using Xilinx ISE simulator and synthesis was done using Xilinx project navigator. The design was synthesized for Spartan3 (xc3s200-5-ft256) device. The device utilization in case of (4 x 4)bit Vedic complex is (No. of Slices: 84 out of 1920 - 4%, Number of 4 input LUTs: 147 out of 3840 - 3%, Number of bonded IOBs: 33 out of 173 - 19%),and delay for (4 x 4)bit Vedic complex is 18.41ns. The device utilization in case (8 x 8) of Vedic complex is (No. of Slices: 385 out of 1920 - 20%, Number of 4 input LUTs: 674 out of 3840 - 17%, Number of bonded IOBs: 66 out of 173 - 36%), and delay for (8 x 8) bit Vedic complex is 30.90ns. The device utilization in case (16 x 16) of Vedic complex is (No. of Slices: 1166 out of 1920 - 60%, Number of 4 input LUTs: 2038 out of 3840 - 53%, Number of bonded IOBs: 128 out of 173 - 73%), and delay for (16 x 16) bit Vedic complex is 40.25ns. Vedic complex multiplier has the greatest advantage as compared to other complex multipliers over gate delays and regularity of structures. The results also suggest that Vedic complex multiplier is faster than other complex multipliers and thus this is extremely advantageous.

VI. CONCLUSION

Vedic Mathematics gives us a clue of symmetric computation. Vedic mathematics deals with various top ics of mathematics such as basic arithmetic, geometry, trigonometry, calculus etc. All these methods are very efficient as far as manual calculations are concerned. The proposed Vedic complex multiplier p roves to be highly efficient in terms of the speed. The main advantage is delay increases slowly as the input bits increases. Most of the important DSP algorithms, such as convolution, discrete Fourier transforms, fast Fourier transforms, digital filters. Since the multiplication time is generally far greater than the addition time, the total processing time for any DSP algorithm primarily depends up on the number of multiplications. Hence, this multiplier can be used to implement the above DSP algorithms.

ACKNOWLEDGMENT

I **Ms.Rajashri K. Bhongade** seize this opportunity to thank all the people who directly or indirectly helped me in completion of my project. I take immense pride in paying gratitude to my project guide **Prof. Ms. Sharada Mungale; Prof. Mrs. Karuna Bogawar** , who always appreciated me for innovative ideas and given me a chance to implement them.

REFERENCES

[1] Charles. Roth Jr. "*Digital Systems Design using VHDL*,"Thomson Brooks/Cole, 7th reprint, 2005.

[2] Jagadguru Swami Sri Bharati Krisna Tirthaji Maharaja, *Vedic Mathematics: Sixteen Simple Mathematical Formulae from the Veda*, Delhi (1965).

[3] H. Thapliyal and M. B. Shrinivas and H. Arbania, "Design and Analysis of a VLSI Based High Performance Low Power Parallel Square Architecture", Int. Conf. Algo. Math.Comp. Sc., Las Vegas, June 2005, pp. 72-76.

[4] P. D. Chidgupkar and M. T. Karad, "The Implementation of Vedic Algorithms in Digital Signal Processing", Global J. of Engg. Edu, vol.8, no.2, 2004.

[5] Shamim Akhter,"VHDL Implementation Of Fast NXN Multiplier Based On Vedic Mathematics", Jaypee Institute of Information Technology University, Noida, 201307 UP, INDIA, 2007 IEEE.

[6] Harpreet Singh Dhillon and Abhijit Mitra, "A Reduced- Bit Multiplication Algorithm for Digital Arithmetics", International Journal of Computational and Mathematical Sciences 2;2 © www.waset

[7] Himanshu Thapliyal and M.B Srinivas, "An Efficient Method of Elliptic Curve Encryption Using Ancient Indian Vedic Mathematics", IEEE, 2005.

[8] H. S. Dillon, etal, "A Reduced Bit Multiplication Algorithm for Digital Arithmetic," International Journal of Computational and Mathematical Sciences, 2008, pp 64-69.

[9] Man Yan Kong, J.M. Pierre, and Dhamin Al-Khalili, "Efficient FPGA Implimentation of Complex Multipliers Using the Logarithmic Number System," 2008 IEEE. Pp 3154-3157.

[10]Langlois Rizalafande Che Ismail and Razaidi Hussin, " High Performance Complex Number Multiplier Using Booth-Wallace Algorithm," *ICSE2006 Proc. 2006, Kuala Lumpur, Malaysia, pp* 786-790.

[11] Deena Dayalan, S.Deborah Priya , "High Speed Energy Efficient ALU Design using Vedic Multiplication Techniques," ACTEA IEEE July 15-17, 2009 Zouk Mosbeh, Lebanon, PP 600-603.

[12] Devika Jaina, Kabiraj Sethi, and Rutuparna Pamda, "Vedic Mathematics Based Multiply Accumulate Unit," International conference on computational intelligence and communication system, 2011, pp 754-757.