RESEARCH ARTICLE OPEN ACCESS

FPGA Implementation of Complex Multiplier Using Urdhva Tiryakbham Sutra of Vedic Mathematics

Rupa A. Tomaskar*, Gopichand D. Khandale**
*(Department of Electronics Engineering, RTMNU University, Nagpur)
** (Department of Electronics Engineering, RTMNU University, Nagpur)

Abstract
In this work VHDL implementation of complex number multiplier using ancient Vedic mathematics is presented, also the FPGA implementation of 4-bit complex multiplier using Vedic sutra is done on SPARTAN 3 FPGA kit. The idea for designing the multiplier unit is adopted from ancient Indian mathematics "Vedas". The Urdhva Tiryakbhyam sutra (method) was selected for implementation since it is applicable to all cases of multiplication. The feature of this method is any multi-bit multiplication can be reduced down to single bit multiplication and addition. On account of these formulas, the partial products and sums are generated in one step which reduces the carry propagation from LSB to MSB. The implementation of the Vedic mathematics and their application to the complex multiplier ensure substantial reduction of propagation delay. The simulation results for 4-bit, 8-bit, 16-bit and 32 bit complex number multiplication using Vedic sutra are illustrated. The results show that Urdhva Tiryakbhyam sutra with less number of bits may be used to implement multiplier efficiently in signal processing algorithms.

Keywords: FPGA Implementation, Signal Processing Algorithms, Vedic Multiplier, VHDL Implementation, Vedas, Complex number multiplier.

I. Introduction

The fundamental and the core of all the Digital Signal Processors (DSPs) are its multipliers and the speed of the DSPs is mainly determined by the speed of its multiplier. Complex number operations are the backbone of many digital signal processing algorithms which mostly depend on extensive number of multiplication. A systems performance is generally determined by the performance of the multiplier, since the multiplier is generally the slowest element in the system [2]. Complex number multiplication involves four real number multiplication and two additions/ subtractions [3]. While doing real number multiplication, carry needs to be propagated from the least significant bit (LSB) to most significant bit (MSB) when binary partial products are added. The overall speed is drop down by the addition and subtraction after binary multiplication [4] [5].

Vedic Mathematics is an ancient mathematics which is based on 16-sutras and 16-sub sutras invented by Jagadguru Shankaracharya Bharati Krishna Teerthaji Maharaja (1884-1960) [1]. Mainly multiplication in Vedic mathematics in carried out using three sutras Nikhilam Navatascaraman Dasatah, Ekadhikena Purvena and Urdhva Tiryakbhyam [1]. Urdhva Tiryakbhyam sutra is the targeted Vedic sutra (algorithm) as it is suitable for all cases of multiplication. The most common multiplication algorithms used are Array multiplication and Booths algorithm. The computational time in case of Array multipliers are comparatively less since the partial results are calculated in parallel. Multiplication using Booths algorithms takes comparable computational time [11]. These algorithms are used for multi-bit and exponential operations that require large partial results and carry registers [3]. This paper presents multiplication algorithm that may be useful in the efficient implementation of signal processing algorithms. The framework of this multiplication algorithm is based on Urdhva Tiryakbhyam sutra of Vedic mathematics.

This paper is organized as follows. In section II the implementation of complex number multiplier using Gauss's multiplication equations, and Urdhva Tiryakbhyam sutra of Vedic mathematics is explained with example. In Section III the proposed methodology for the implementation of complex multiplier is discussed.

II. Implementation

While implementing complex number multiplication, the multiplication system can be divided into two main components giving the two separate results known as real part (R) and imaginary part (I).

$$R + j\,I = (A + j\,B)\,(C + j\,D) \quad \text{-------} \quad (1)$$

Gauss's algorithm for complex number multiplication gives two separate equations to calculate real and imaginary part of the final result. From equation (1) the real part of the output can be given by (AC - BD) and the imaginary part of the

result can be computed using (BC + AD). Thus four separate multiplications and are required to produce the real as well as imaginary part numbers [10] [11] [12].

2.1. Vedic multiplication method

Multiplication process is the critical part for any complex number multiplier design. There are three major steps involved for multiplication. Partial products are generated in first step. In second step partial product reduction to one row of final sums and carries is done. Third and final stage adds the final sums and carries to give the result.

The proposed complex number multiplier is based on the Vedic multiplication formulae (Sutras). These sutras have been traditionally used for the multiplication of two numbers in decimal number system. In this work the same ideas are applied to binary number system, to make the proposed algorithm compatible with the digital hardware.

2.2 Urdhva Tiryakbhyam sutra

Urdhva Tiryakbhyam sutra is suitable for all cases of multiplication. It literally means "Vertically and crosswise" [8]. Multiplication using this sutra is performed by vertically and crosswise, vertically means straight above multiplication and crosswise means diagonal multiplication and taking their sum. The feature of this method is any multi-bit multiplication can be reduced down to single bit multiplication and addition [1]. The multiplication is illustrated below using an example. The crosswise and vertical multiplication be implemented starting either from right hand side or left hand side [7] [8].

Example 1: Multiplication of 42 and 13

Step 1: Starting at the left multiply two left hands most significant digits vertically and set down results underneath as the left hand most significant part of the answer.

```
4       2
1       3
4 ((4 * 1) = 4)
```

Step 2: Next multiple crosswise and add these partial results. Set down the result of addition as illustrate below.

```
4       2
1       3
4       4 (((4 * 3) + (1* 2)) = 14)
1
```

Step 3: multiply two rights hand least significant digits vertically and set down results underneath as the right hand least significant part of the answer.

```
4       2
1       3
4       4       6 ((2 * 3) = 6)
1
```

Step 4: Finally add the digits vertically as illustrated

```
4       2
1       3
4       4       6
1
5       4       6
```

Result of multiplication 42 * 13 = 546.

Thus the above method is equally applicable for binary multiplication. Fig. 3 shows the general multiplication procedure of the 4x4 multiplication. The multiplication of two 4-bit binary numbers a3a2a1a0 and b3b2b1b0 is performed starting from left hand side. Every step in fig. 3 has a corresponding expression as follows:

$$r0=a0b0 \quad (1)$$
$$c1r1=a1b0+a0b1 \quad (2)$$
$$c2r2=c1+a2b0+a1b1 + a0b2 \quad (3)$$
$$c3r3=c2+a3b0+a2b1 + a1b2 + a0b3 \quad (4)$$
$$c4r4=c3+a3b1+a2b2 + a1b3 \quad (5)$$
$$c5r5=c4+a3b2+a2b3 \quad (6)$$
$$c6r6=c5+a3b3 \quad (7)$$

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

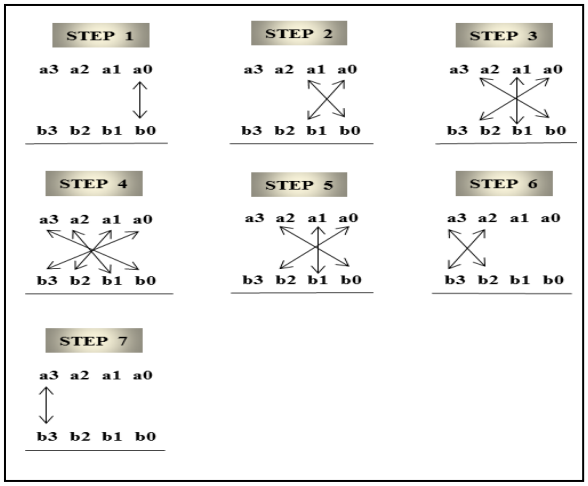

Fig.1. Vertically crosswise multiplication of four bit binary number [3]

III. Discussion

The complex number multiplier using Urdhva Tiryakbhyam sutra is implemented using VHDL. The methodology used to implement the complex multiplier using Gauss's multiplication equations are described with the help of block diagram is shown in fig. 2. As discussed earlier complex multiplication requires four multiplications and an addition and subtraction.

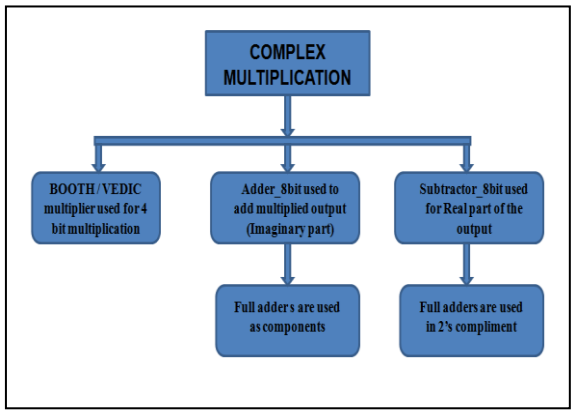

Fig.2. Proposed methodology for complex number multiplication

Using the proposed algorithm 4-bit, 8-bit, and 16-bit complex multiplication is achieved. Functional verification of the code through simulation is carried out using Xilinx ISE simulator. The complete code is synthesized using Xilinx synthesis tool (XST). Table 1 indicates the device utilization summary of the Vedic complex multiplier for 4-bit, 8-bit, 16-bit multiplication. Figure 3,4 and 5 shows the RTL schematic of 16-bit complex multiplier using Vedic algorithm.

TABLE 1. DEVICE UTILIZATION SUMMARY

Vedic complex Algorithm	NO. of Slices	NO. of 4 input LUT's	NO. Of IO's	No. of bonded IOB's	Delay in ns.
4-bit	84	147	33	33	18.419
8-bit	385	674	64	64	30.900
16-bit	1166	2038	128	128	40.250

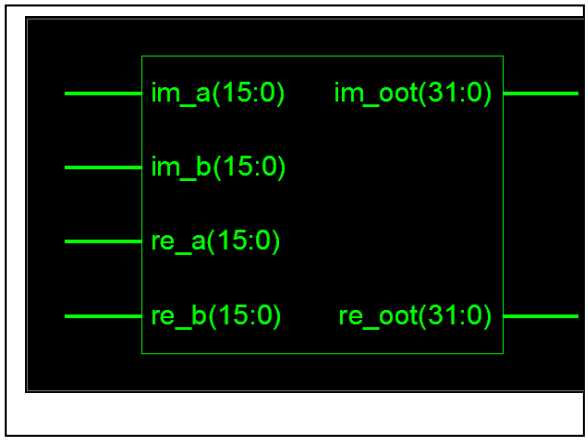

Fig. 3. RTL schematic [partial] of 4-bit complex multiplier (Vedic multiplier)

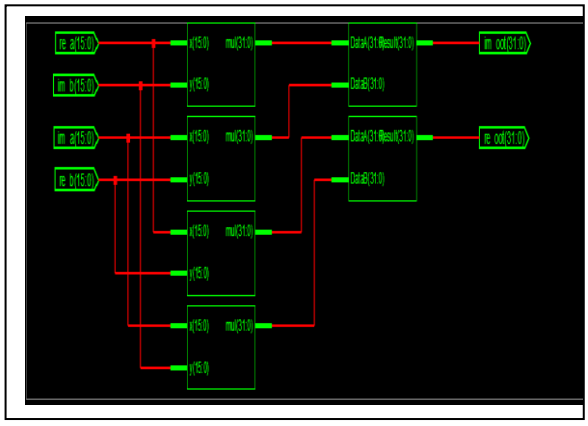

Fig. 4.. RTL schematic [partial] of 4-bit complex multiplier (Vedic multiplier)

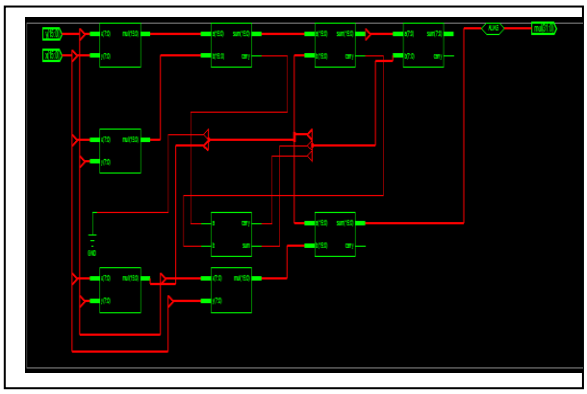

Fig. 5. RTL schematic [partial] of 4-bit complex multiplier (Booth's)

IV. 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 obtained results are presented in table 1, and waveforms for 4-bit, 8-bit and 16-bit unsigned complex multiplication using Urdhva Tiryakbhyam algorithm is shown in figure 6, 7 and 8 respectively.

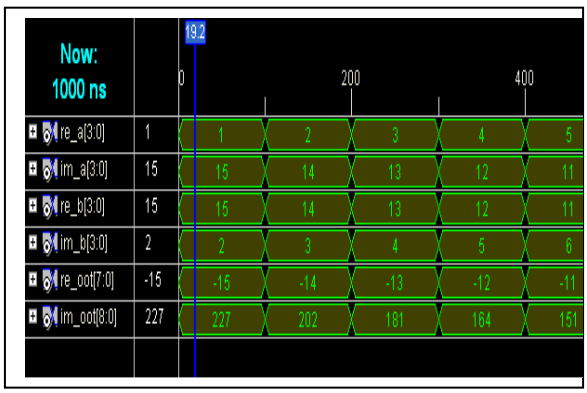

Fig. 6. Simulation waveforms for 4-bit complex multiplication

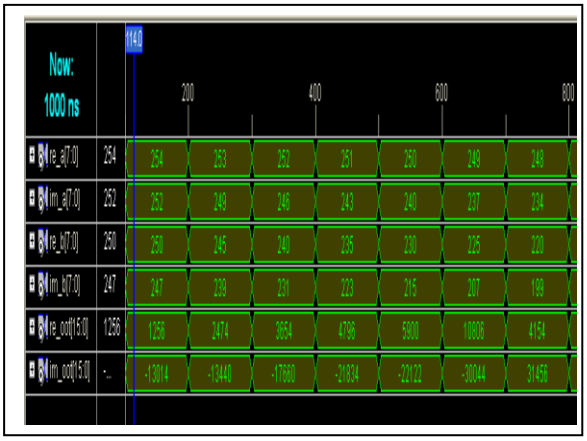

Fig. 7. Simulation waveforms for 8-bit complex multiplication

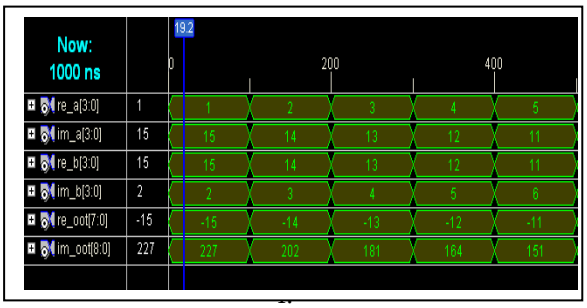

Fig. 8. Simulation waveforms for 16-bit complex multiplication

The FPGA implementation of 4 bit unsigned complex number using the proposed algorithm on SPARTAN 3FPGA kit is shown in figure 9.

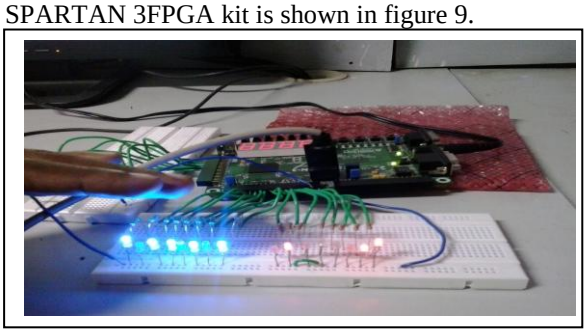

Fig. 9 . FPGA implementation of 4-bit complex Vedic multiplier on Spartan 3 kit.

The input is taken as:
Input A = (13 + 9i)
Input B = (10 + 7i)
13 = 1101, 9 = 1001
10 = 1010, 7 = 0111
After multiplication the result is:
A x B = (AC - BD) + (BC +AD) i
= (67) + (181) i
67 = 01000011, 181 = 10110101
Here red LED's are used to show the real part of the output, while blue LED's indicates the imaginary part of the output.

V. Conclusion

In many real-time DSP applications number of complex multiplications are involved, in which high performance is a prime target. However, achieving this may be done at the expense of area, power dissipation and accuracy. The performance in terms of throughput of the processor is limited by the multiplication. So efforts have to be made to decrease the number of multipliers and to increase their speed. A high speed complex number multiplier design using Vedic Mathematics (Urdhva Tiryakbhyam sutra) is implemented using VHDL. This sutra is applicable to all cases of multiplication. The results show that Urdhva Tiryakbhyam sutra with less number of bits may be used to implement high speed complex multiplier efficiently in digital signal processing algorithms.

References

[1] Jagadguru Swami Sri Bharati Krishna Teerthaji Maharaja, "Vedic Mathematics," Motilal Banarsidas Publishers Pvt. Ltd, 2001.

[2] L. Sriraman, T. N. Prabakar, "Design and Implementation of Two Variable Multiplier Using KCM and Vedic Mathematics," 1st International Conf. on Recent Advances in Information Technology, RAIT, 2012.

[3] Sandesh S. Saokar, R. M. Banakar, and Saroja Siddamal, "High Speed Signed Multiplier for Digital Signal Processing Applications," 2012 IEEE.

[4] S.S. Kerur, Prakash Narchi Jayashree C .N., Harish M. Kittur "Implementation of Vedic multiplier for digital signal processing," international journal of computer application, 2011, vol. 16, pp 1-5.

[5] 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.

[6] V Jayaprakasan, S Vjayakumar, and V S Kanchana Bhaaskaran, "Evaluation of the Conventional vs. Ancient Computation methodology for Energy Efficient Arithmetic Architecture," 2011 IEEE.

[7] Rudagi J. M.,et al, "Design And Implementation of Efficient Multiplier Using Vedic Mathematics ," International Conference on Advances in Recent Technologies in Communication and Computing, 2011,pp 162-166.

[8] Thakre L. P., et al,"Performance Evaluation and Synthesis of Multiplier used in FFT operation using Conventional and Vedic algorithms," Third International Conference

on Emerging Trends in Engineering and Technology, ICETET.2010, pp 614-619.

[9] 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.

[10] H. S. Dhillon, et al, "A Reduced Bit Multiplication Algorithm for Digital Arithmetic," International Journal of Computational and Mathematical Sciences, 2008, pp 64-69.

[11] 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.

[12] 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.