Ancient Vedic Method for Complex Number Multiplication to Minimize Time Delay and Hardware Complexity.

[1] Shubhashish S. Wayzade, [2] Jyoti M. Varavadekar

[1] Post-Graduate student, EXTC Dept, K. J. Somaiya College of Engineering, Mumbai. India, Contact no. +918975757877, (e-mail: shubhashish.w@somaiya.edu)

[2] Associate Professor, EXTC Dept, K. J. Somaiya College of Engineering, Mumbai. India, Contact no. +919833115068 (e-mail: jyotivaravadekar@somaiya.edu)

Abstract— The core of all the digital signal processors (DSPs) are multipliers. Basically, the operational speed of any digital signal processor is strictly dependent upon the speed of the multipliers used [7]. Complex numbers have various applications in the area of digital signal processors [2]. Here, in this paper, various methodologies that are used for the complex number multiplications are discussed and their results on the basis of their performance are compared. Here, we have used Urdhva Tiryakbhyam method of ancient Vedic mathematics which is derived from the ancient Vedic sutras and booth algorithm method, which is another method for complex number multiplication are used. The meaning of "Urdhva Tiryakbhyam" is vertically and crosswise. Vertically means straight above multiplication and crosswise means diagonal multiplication and taking their sum [1]. The exceptional feature of this Urdhva Tiryakbhyam method is that it reduces any multi bit multiplication into single bit multiplication and addition. Therefore, all the partial product terms gets generate in one step which further reduces carry propagation that occurs from least significant bit to the most significant bit during the process of addition. The comparison between the two methods is done on the basis of performance parameters such as time delay and hardware complexity in terms of gate count. The results show that complex number multiplication using Urdhva Tiryakbhyam method of Vedic mathematics gives better results as compared to booth algorithm method. Hence, Urdhva Tiryakbhyam method of ancient Vedic mathematics with less number of bits can be used to implement multiplier efficiently in signal processing algorithms.

Keywords— Multipliers, Partial Product terms, Booths Algorithm, Ancient Vedic Mathematics, Nikhilam Sutra, Urdhva Tiryakbhyam Sutra, Complex Number Multiplication, etc.

1. INTRODUCTION

In many high performance systems, multipliers are the key components. The operational performance of any system is totally depends upon the operating speed of the multipliers. In other words, higher the operational speed of the multipliers, higher is the performance of that system [6]. High performance systems such as digital signal processors have wide area of application especially in multimedia applications such as 3D graphics which depends on large number of multiplications [3]. The complex number multiplication of any two complex numbers can be obtained separately as real part and an imaginary part. While obtaining the real part of the output, during binary multiplication, the carry needs to propagate from least significant bit (LSB) to the most significant bit (MSB). Hence, addition and subtraction after the binary number multiplication causes decrease in the overall speed of the process [5].

In 19th century, Jagadguru Shri Bharti Krishna Tirthaji maharaj has invented Vedic mathematics from the ancient Indian Vedas. In that, he discovered 16 sutras and their 16 sub sutras. The complex number multiplication can be done by using three sutras of Vedic mathematics. Those three sutras are Urdhva Tiryakbhyam sutra, Ekadhikena Purvena, and Nikhilam Navatascaraman Dasatah or simply Nikhilam [1]. The Nikhilam sutra of Vedic mathematics can only be applied to large number multiplication. While the Urdhva Tiryakbhyam method of Vedic mathematics can be efficiently applied to all cases of multiplication. Urdhva Tiryakbhyam sutra is very simple and it is very easy to implement. Another commonly used methodology is the booth algorithm method of multiplication [11]. This paper gives brief description of an effective method for the complex number multiplication. This method is based on the Urdhva Tiryakbhyam sutra of ancient Vedic mathematics.

This paper is organized as; section 1 gives the introduction. In section 2, multiplication of complex numbers is discussed. Section 3 gives different methods used for complex number multiplication and their performance using different parameters. Section 4 gives architectures for complex number multiplication using the urdhva tiryakbhyam sutra of vedic maths. Results and comparison are covered in section 5 followed by conclusion in section 6.

2. MULTIPLICATION OF COMPLEX NUMBERS

Let us consider two complex numbers (P + j Q) and (X + j Y). The output of the multiplication of these two complex numbers can be obtained separately as real part and an imaginary part. For example,

$$R + j\,I = (P + j\,Q)\,(X + j\,Y) \qquad \text{-----------} \ (I)$$

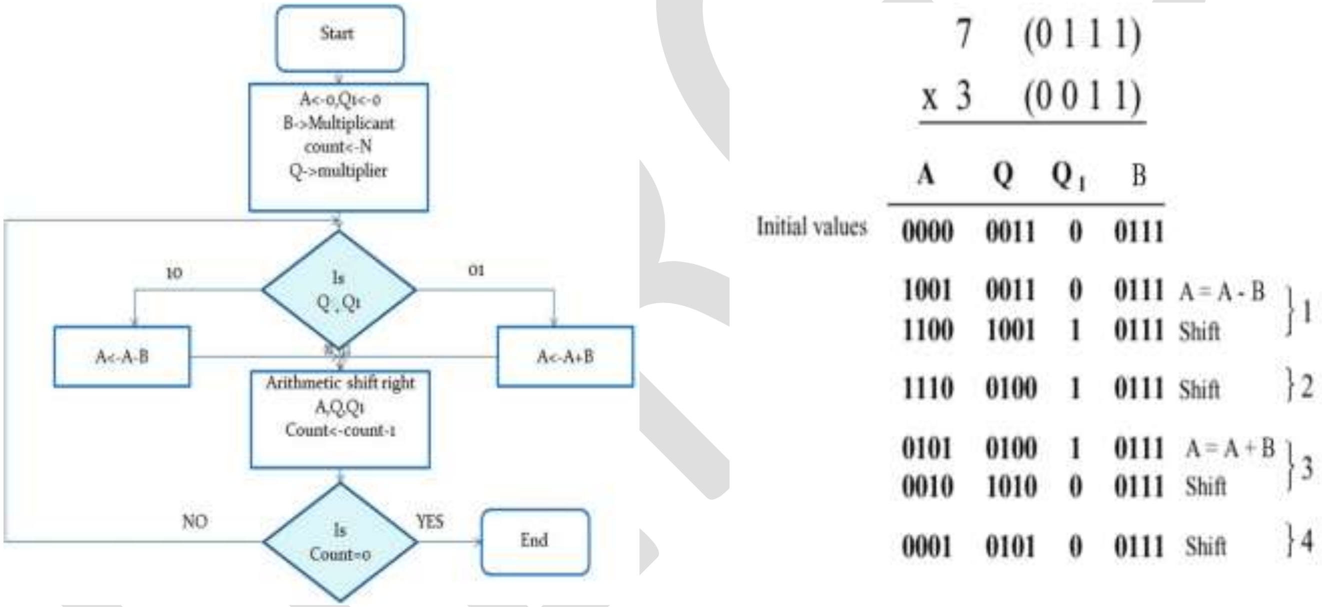
From above, multiplication of two complex numbers gives us two separate results. Here R represents the real part of the result and I represents an imaginary part of the result. The real part of the result of complex number multiplication can be obtained from (PX-QY) while the imaginary part of the result of complex number multiplication can be obtained from (PY+QX). Therefore, the result of the two complex number multiplication can be obtained by performing four separate multiplications, one addition and one subtraction [4].

3. DIFFERENT METHODS AND THEIR PERFORMANCES

For complex number multiplication different methods are used. Here, we have discussed booth algorithm method and the ancient Vedic methods for complex number multiplication. Ancient Vedic method includes Nikhilam sutra and Urdhva Tiryakbhyam sutra. The performance of these two methods can be observed by parameters such as operational time delay and hardware complexity in terms of gate count. The detailed description of these two methods is given below and the parameters are calculated using Urdhva Tiryakbhyam sutra and are compared with booths algorithm results

A) Booths Algorithm:

This is one of the good methods used for multiplication. Andrew Donald Booth the booths algorithm. Figure below shows the flow chart for booths algorithm and example explaining the implementation of booths algorithm.



(a)

	7	(0111)				
	x 3	(0011)				
	A	Q	Q_1	B		
Initial values	0000	0011	0	0111		
	1001	0011	0	0111	A = A - B	}1
	1100	1001	1	0111	Shift	
	1110	0100	1	0111	Shift	}2
	0101	0100	1	0111	A = A + B	}3
	0010	1010	0	0111	Shift	
	0001	0101	0	0111	Shift	}4

(b)

Figure 1. (a) Flow chart of the Booths Algorithm. (b) Multiplication of two numbers using booths algorithm.

Here, as shown in above figure, two registers A and Q1 should be considered and we have to initialise them to 0. Also, multiplicand value should be assigned to register B and multiplier value should be assigned to register Q. After the least significant bit of register Q, the Q1 register of one bit should be placed [9].

Now, the LSB of register Q i.e. Q(0) and the value of register Q1are checked for the three cases given below.

1) If value in Q(0) and Q1 are ‘0’ and ‘0’ or ‘1’ and ‘1’ then, we have to shift bits of register A, Q and Q1 to right by 1 bit position.

2) If value in Q(0) and Q1 are ‘0’ and ‘1’ then, add multiplicand with the bits in register A and then we have to shift bits of register A, Q and Q1 to right by 1 bit position.

3) If value in Q(0) and Q1 are ‘1’ and ‘0’ then, subtract multiplicand from the bits in register A and then we have to shift bits of register A, Q and Q1 to right by 1 bit position.

The final result will be the combination of bits in register A and Q respectively [11].

Although booths algorithm gives us proper results but it comes with certain drawbacks. Vedic mathematics method for multiplication can be used to overcome these drawbacks. This method is explained below.

B) Ancient Vedic Methods:

From all the 16 sutras in the ancient vedic mathematics, Nikhilam Navatascaraman Dasatah sutra and Urdhva Tiryakbhyam sutra can be used from complex number multiplication. These sutras are explained below.

i) Nikhilam Navatascaraman Dasatah or simply Nikhilam- The meaning of this Nikhilam Navatascaraman Dasatah is "all from 9 and last from 10". This Nikhilam sutra can also be applied to all cases but this method gives more effective results when it is applied to large numbers. Here, in this method, we have to select base, which has to be nearer and it must be greater than the numbers selected for multiplication [12]. The Nikhilam method takes the complement of the selected numbers for multiplication from its nearest base and performs multiplication over those two complement numbers. The necessary criteria to choose numbers for multiplication is that they should be greater than 10n/2. In other words, if we have to choose two numbers m and n, then the criteria to choose these numbers will be m > 10n/2 and n > 10n/2. The Nikhilam sutra gives more effective results when the numbers involved are large [10]. Therefore, larger the number more is the efficiency of the result. The illustration of Nikhilam sutra by taking example of two decimal numbers is given below.

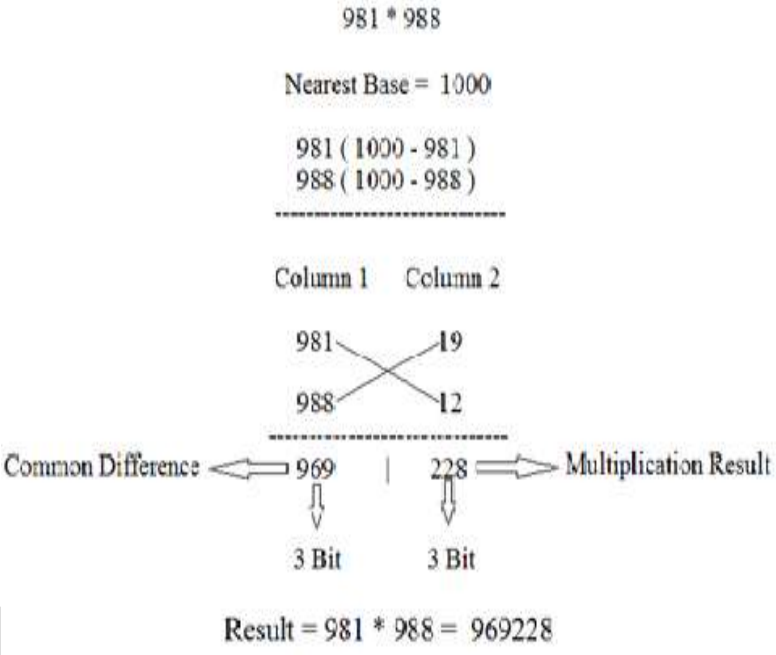


Figure 2: Multiplication of the two decimal numbers using Nikhilam sutra.

Here, as shown in above figure, we have considered two decimal numbers 981 and 988. The nearest base which has to be greater than these two numbers is 1000. As shown in above figure, the multiplicand and the multiplier should be written in column no.1. We have to write their respective complements in column no. 2. In this example, the complement of 981 can be obtained as 1000 - 981 = 19 and similarly, the complement of 988 can be obtained as 1000 – 988 = 12. The result of the multiplication of these two choosen numbers can be obtained in two different parts which are separated by a vertical line as shown in the above figure. We can obtain the left hand side part of the result by taking the common difference, that is, either we can have 981 – 12 = 969 or we can have 988 – 19 = 969. Also the right hand side part of the result can be obtained by simply multiplying the complements which are written in column no. 2. Hence, the total multiplication result can be obtained by combining the left hand side and the right hand side part of the result, that is, 981 x 988 = 969228.

There is a special case while applying Nikhilam sutra for multiplication. It is very important to note that, the result on the right hand side part must have n digits [8]. But, sometimes, the digits on the right hand side part of the result are less than n. This is called the special case in Nikhilam sutra. For that, let us consider an example of two decimal numbers 994 and 997 as shown in figure below. The nearest base which has to be greater than these two numbers is 1000. In this example, the complement of 994 can be obtained as 1000 - 994 = 6 and similarly, the complement of 997 can be obtained as 1000 – 997 = 3. The result of the multiplication of these two choosen numbers can be obtained in two different parts which are separated by a vertical line as shown in the above figure. We can obtain the left hand side part of the result by taking the common difference, that is, either we can have 994 – 3 = 991 or we can have 997 – 6 = 991. The right hand side part of the result can be obtained by simply multiplying the complements which are written in column no. 2, that is, 6 x 3 = 18. Here, we can observe that, the digits in the right hand side part of the results are less than n. So, for getting an appropriate result, we have to append a leading zero before all the digits in the right hand side part of the result. Thus, the right hand side part of the result becomes 018. Hence, the total multiplication result can be obtained by combining the left hand side and the right hand side part of the result, that is, 997 – 994 = 991018. On the other hand, if the digits on the right hand side part of the result are four, then the MSB will be taken as carry digit for left hand side part of the result.

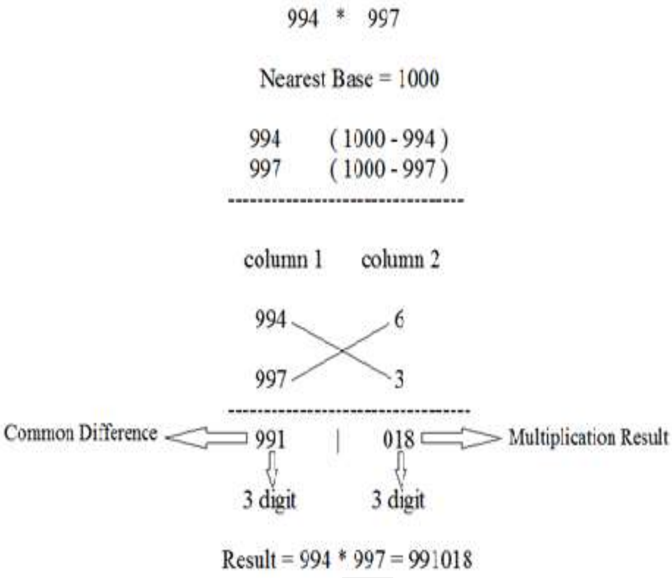


Figure 3: Multiplication of the two decimal numbers using Nikhilam sutra. (Special case).

ii) Urdhva Tiryakbhyam method- It is another method used for multiplication of complex number. Urdhva Tiryakbhyam method means “vertically and crosswise”. Vertically means straight above multiplication and crosswise means diagonal multiplication and taking their sum. It has advantage that it reduces the multi bit multiplication into single bit multiplication and addition. This results in generation of all the partial products in one step which further reduces carry propagation that occurs from LSB to MSB during the process of addition. We can either implement this sutra starting from right hand side or from left hand side [13].

The straight above multiplication and diagonal multiplication and taking their addition in Urdhva Tiryakbhyam method in case of two four bit numbers a and b can be better understand from step 1 to step 7 as shown in figure below.

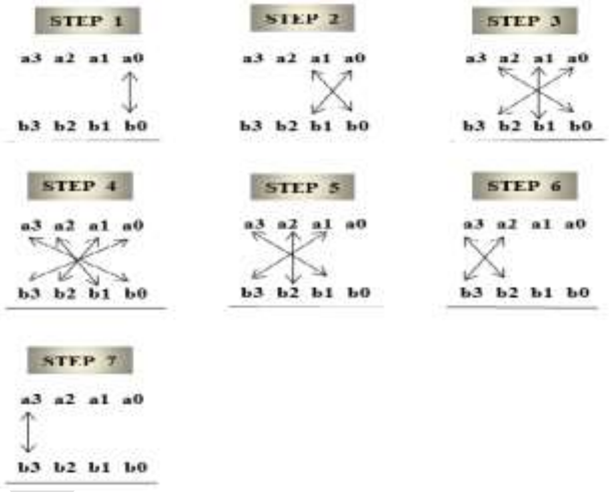


Figure 4: Line diagram of Urdhva Tiryakbhyam Sutra (method) for 4 x 4 binary number multiplication.

Let us take an example of two four bit binary numbers a and b by applying vertically and crosswise method to it, as shown in figure.

p0 = a0b0 -------------- (I)

s1p1 = a1b0 + a0b1 -------------- (II)

s2p2 = s1 + a2b0 + a1b1 + a0b2 ------------- (III)

s3p3 = s2 + a3b0 + a2b1 + a1b2 + a0b3 ------------- (IV) s4p4 = s3 + a3b1 + a2b2 + a1b3 ------------- (V)

s5p5 = s4 + a3b2 + a2b3 ------------- (VI)

s6p6 = s5 + a3b3 ------------- (VII)

From equation no. (I) to (VII), the final result can be obtained as s6p6p5p4p3p2p1p0.

4. ARCHITECTURES OF DIFFERENT MULTIPLIERS USING THE URDHVA TIRYAKBHYAM SUTRA

By using the Urdhva Tiryakbhyam sutra we can also implement N x N multiplier. The architectures for 2 x 2, 4 x 4, and 8 x 8 bit vedic multipliers using the Urdhva Tiryakbhyam sutra are explained below.

i) 2 x 2 bit Vedic multiplier architecture:

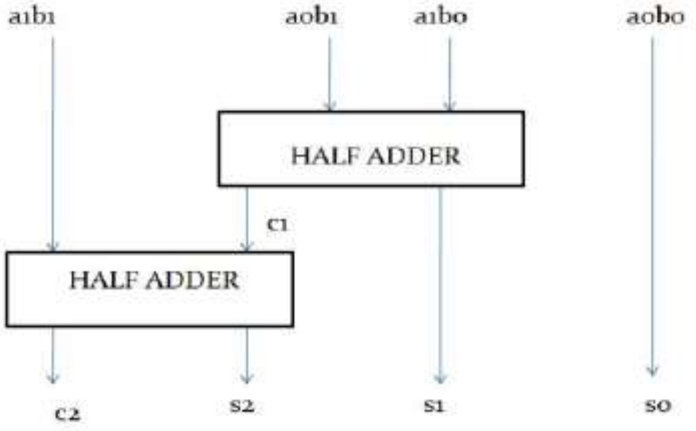


Figure 5. Architecture of 2 x 2 bit vedic multiplier.

The 2 x 2 bit Vedic multiplier using the Urdhva Tiryakbhyam sutra includes two half adders [8]. Here, we have two 2-bit numbers a and b as input. In this, we will require four AND gates. The overall multiplication process will be as shown in above fig. and the final output of multiplication obtained will be of 4-bit.

ii) 4 x 4 bit Vedic multiplier architecture:

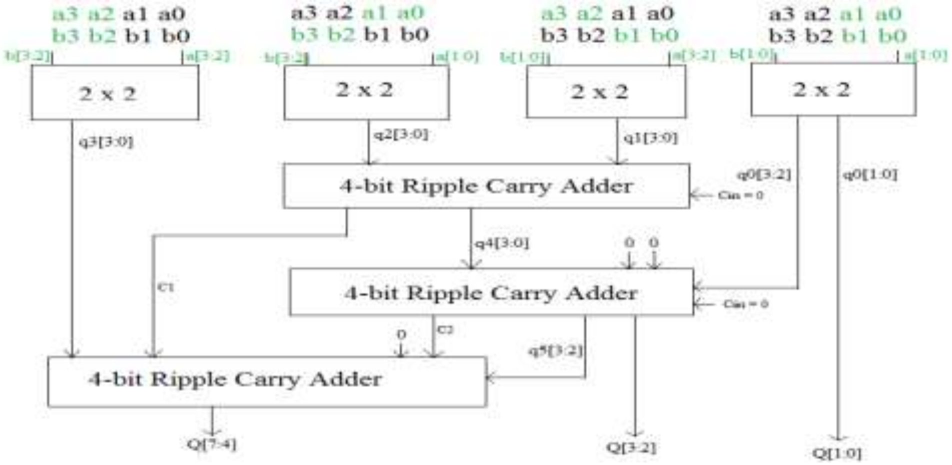


Figure 6. Architecture of 4 x 4 bit vedic multiplier

The 4 x 4 bit Vedic multiplier architecture includes four 2 x 2 bit Vedic multiplier module and three 4-bit ripple carry adders [13]. Each 4-bit ripple carry adder block contains four 1-bit full adders. It is called ripple carry adder because each time the carry ripples. Cin to the next block of full adder is Cout from the previous block of the full adder. Thus, the final result will be of 8-bit for 4 x 4 bit Vedic multiplier using the Urdhva Tiryakbhyam sutra.

iii) 8 x 8 bit Vedic multiplier architecture:

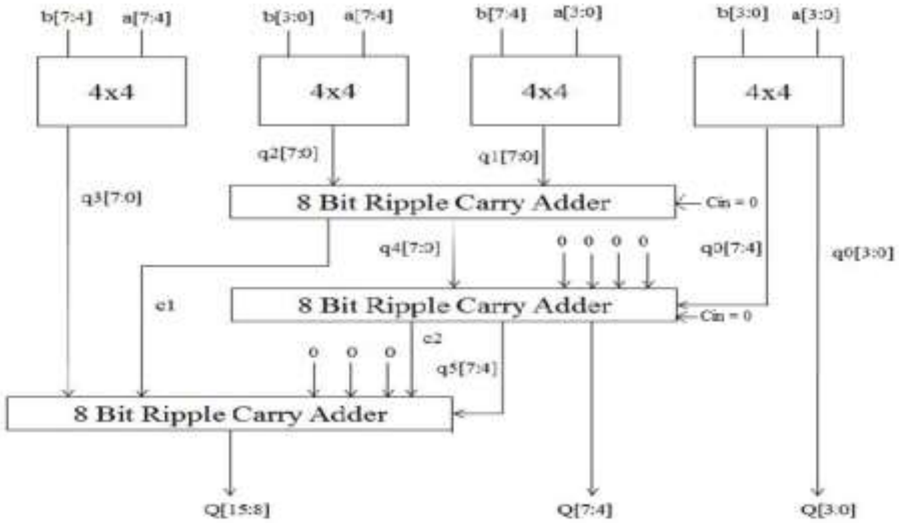


Figure 7. Architecture of 8 x 8 bit vedic multiplier

The 8 x 8 bit Vedic multiplier architecture uses four 4 x 4 bit Vedic multiplier module and three 8-bit ripple carry adders. Each 8-bit ripple carry adder block contains eight 1-bit full adders where Cout of the previous full adder is given as Cin to the next full adder block. So, we get output result of 16-bit in case of 8 x 8 bit Vedic multiplier using the Urdhva Tiryakbhyam sutra.

The proposed methodology using Urdhva Tiryakbhyam sutra of Vedic mathematics for complex number multiplication is as shown in figure below.

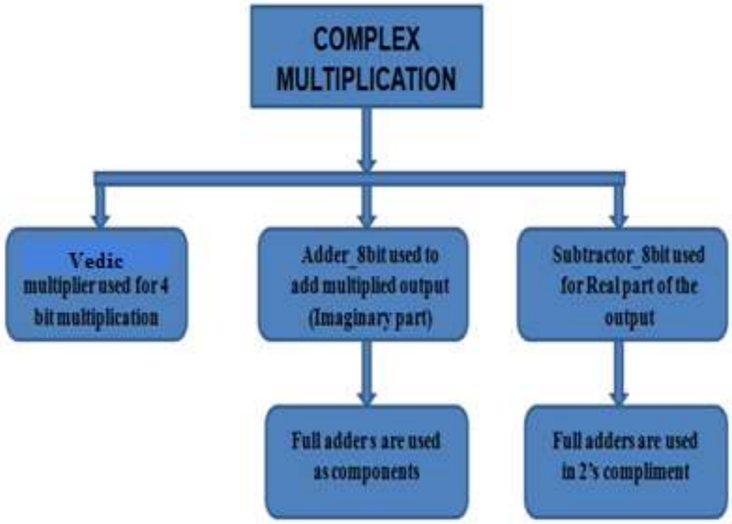
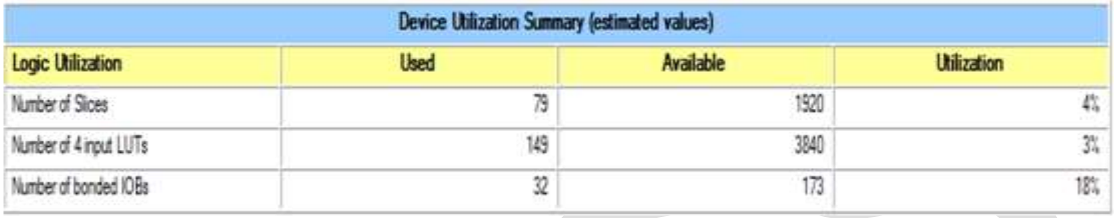

COMPLEX MULTIPLICATION
Vedic multiplier used for 4 bit multiplication
Adder_8bit used to add multiplied output (Imaginary part)
Subtractor_8bit used for Real part of the output
Full adders are used as components
Full adders are used in 2's compliment

Figure 8: Proposed Methodology using Urdhva Tiryakbhyam sutra for complex number multiplication.

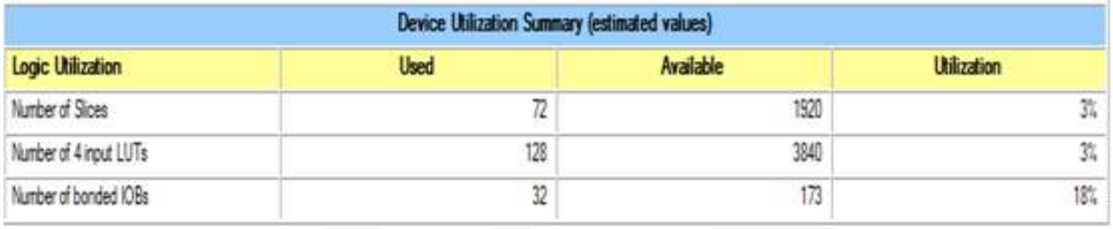
5. RESULTS AND COMPARISON

In this paper, we have seen two methods of complex number multiplication. The booth algorithm method is less effective as compared to Urdhva Tiryakbhyam sutra of Vedic mathematics. As all the partial products gets generate in one step which reduces carry propagation from LSB to MSB. Therefore Urdhva Tiryakbhyam method has the highest operational speed as compared to booth algorithm method. Also the hardware complexity in terms of gate count is less in case of Vedic mathematics method. Therefore, comparative result shows that it is an effective method for the complex number multiplication. Here, we have used Xilinx ISE 8.1i software and family used is SPARTAN 3. The device used is XC3S200 amd selected package is ft256 with the speed grade of -5.



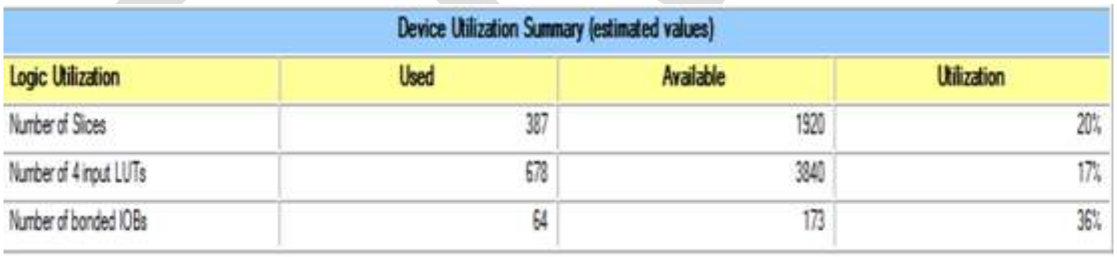
Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	79	1920	4%
Number of 4 input LUTs	149	3840	3%
Number of bonded IOBs	32	173	18%

Figure 9. Device utilization summary using the booths algorithm for 4 bit complex number multiplication



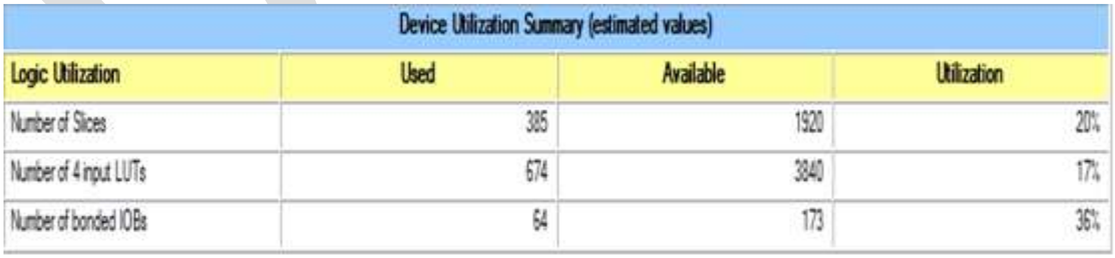
Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	72	1920	3%
Number of 4 input LUTs	128	3840	3%
Number of bonded IOBs	32	173	18%

Figure 10. Device utilization summary using vedic algorithm for 4 bit complex number multiplication



Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	387	1920	20%
Number of 4 input LUTs	678	3840	17%
Number of bonded IOBs	64	173	36%

Figure 11. Device utilization summary using the booths algorithm for 8 bit complex number multiplication

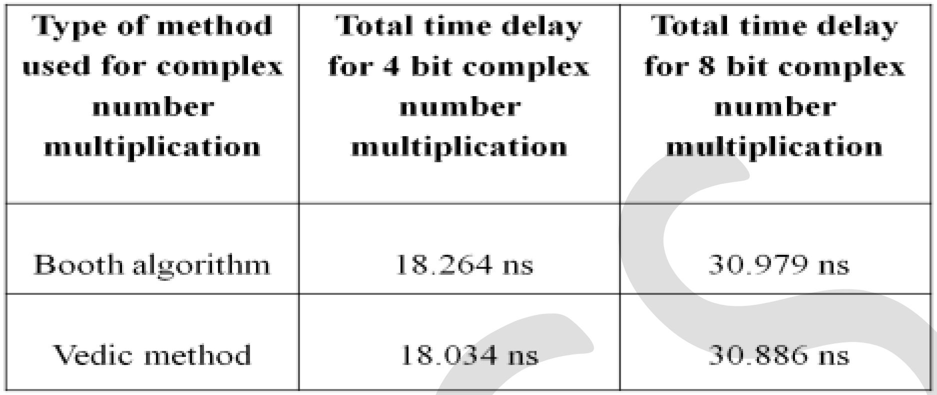


Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	385	1920	20%
Number of 4 input LUTs	674	3840	17%
Number of bonded IOBs	64	173	36%

Figure 12. Device utilization summary using Vedic algorithm for 8 bit complex number multiplication

Fig. 9, and Fig. 10 shows the device utilization summary for both booth algorithm and Vedic algorithm using Urdhva Tiryakbhyam sutra. Also, Fig. 11 and Fig. 12 shows the device utilization summary for booth algorithm and Vedic algorithm using Urdhva Tiryakbhyam sutra. The comparison results in terms of time delay and hardware complexity are shown in Table 1 and Table 2.

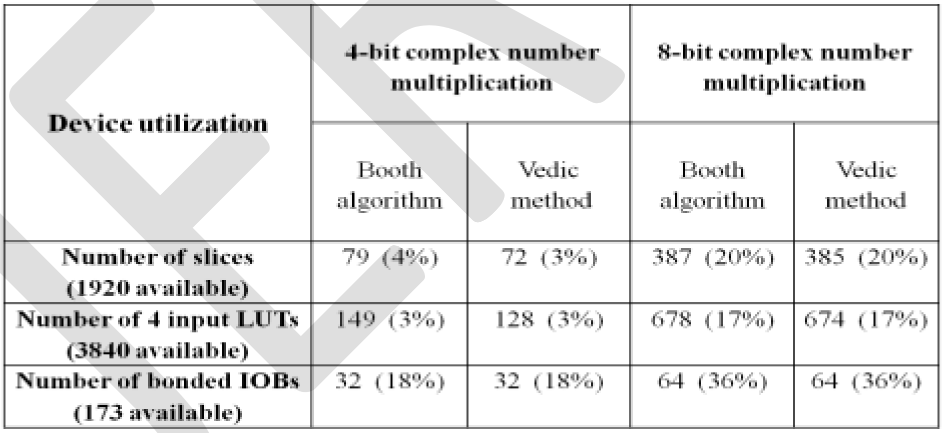
Table 1: Comparison of total time delay for 4 bit and 8 bit complex number multiplication



Type of method used for complex number multiplication	Total time delay for 4 bit complex number multiplication	Total time delay for 8 bit complex number multiplication
Booth algorithm	18.264 ns	30.979 ns
Vedic method	18.034 ns	30.886 ns

From table 1, it is clear that the total time delay for 4 bit complex number multiplication using booths algorithm is more as compared to Vedic method using Urdhva Tiryakbhyam sutra. In case of 4 bit multiplication using booths algorithm, the total time delay is 18.264 ns and that for Vedic method is 18.034 ns. In case of 8 bit complex number multiplication, the total time delay using booths algorithm is 30.979 ns and that of Vedic method is 30.886 ns.

Table 2: Comparison of hardware complexity for 4 bit and 8 bit complex number multiplication



Device utilization	4-bit complex number multiplication		8-bit complex number multiplication	
	Booth algorithm	Vedic method	Booth algorithm	Vedic method
Number of slices (1920 available)	79 (4%)	72 (3%)	387 (20%)	385 (20%)
Number of 4 input LUTs (3840 available)	149 (3%)	128 (3%)	678 (17%)	674 (17%)
Number of bonded IOBs (173 available)	32 (18%)	32 (18%)	64 (36%)	64 (36%)

From table 2, the number of slices used in 4 bit complex number multiplication are 79 (4% of the 1920 available) using the booths algorithm and that in case of Vedic method are 72 (4% of the 1920 available). The number of 4 input LUTs used in 4 bit complex number multiplication are 149 (3% of the 3840 available) using the booths algorithm and that in case of Vedic method are 128 (3% of the 3840 available). The number of bonded IOBs used in 4 bit complex number multiplication are 32 (18% of the 173 available) using the booths algorithm and that in case of Vedic method are 32 (18% of the 173 available). Also, the number of slices used in 8 bit complex number multiplication are 387 (20% of the 1920 available) using the booths algorithm and that in case of Vedic method are 385 (20% of the 1920 available). The number of 4 input LUTs used in 8 bit complex number multiplication are 678 (17% of the 3840 available) using the booths algorithm and that in case of Vedic method are 674 (17% of the 3840 available). The number of bonded IOBs used in 8 bit complex number multiplication are 64 (36% of the 173 available) using the booths algorithm and that in case of Vedic method are 64 (36% of the 173 available).

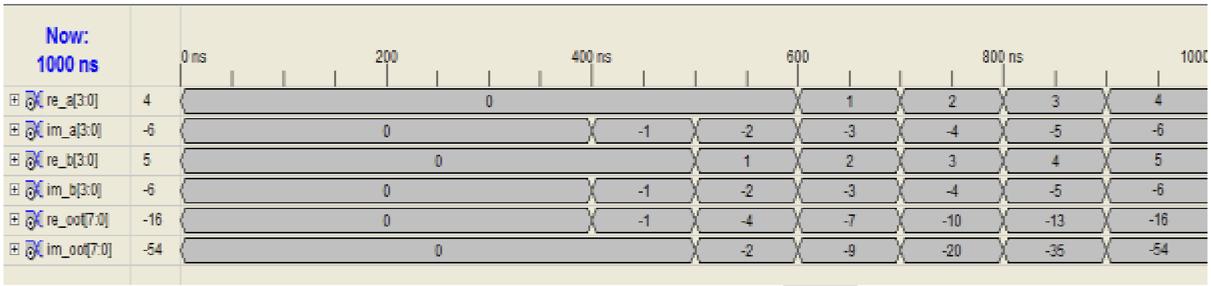


Figure 13. Result for 4 bit complex number multiplication using booths algorithm

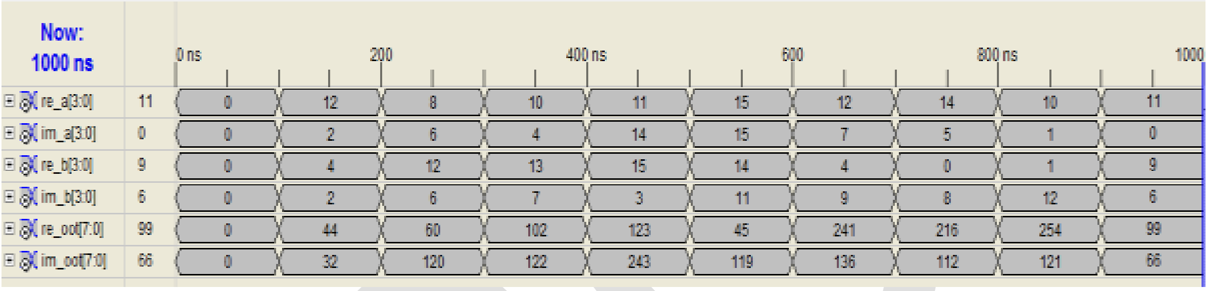


Figure 14. Result for 4 bit complex number multiplication using Urdhva Tiryakbhyam sutra of vedic method

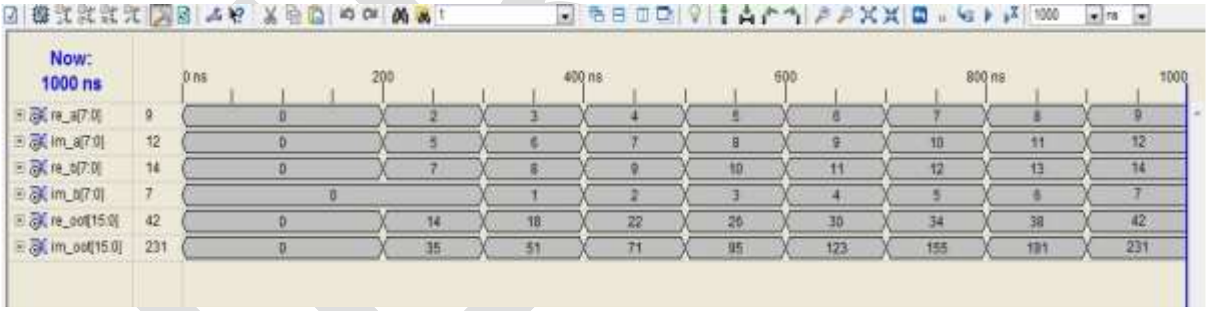


Figure 15. Result for 8 bit complex number multiplication using booths algorithm

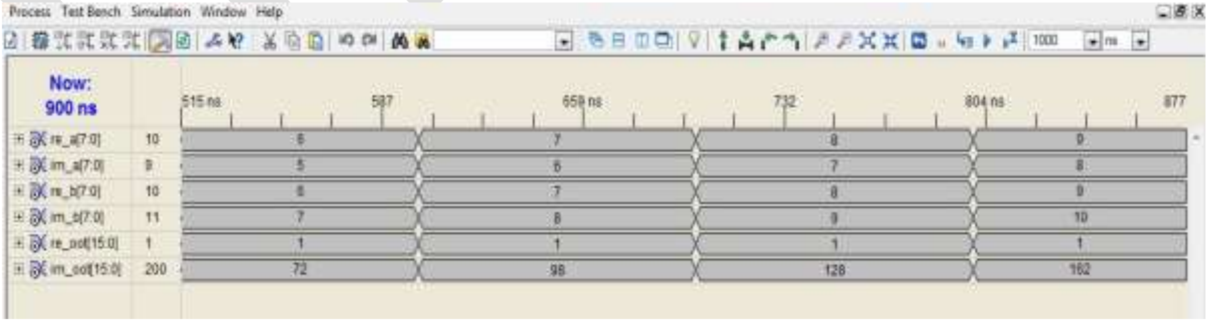


Figure 16. Result for 8 bit complex number multiplication using Urdhva Tiryakbhyam sutra of vedic method

The results for 4 bit and 8 bit complex number multiplication using booths algorithm and vedic method using Urdhva Tiryakbhyam sutra are shown in fig. 13, fig. 14, fig. 15, and fig. 16 above.

6. CONCLUSION

In this paper, for the multiplication of two complex numbers, we have seen two methods. Our main target is to reduce the partial product terms which generate during the multiplication process and hence to reduce the carry propagation which occurs from LSB to MSB during the process of addition. Our targeted method of Vedic mathematics fulfills both the above conditions by generating the partial product terms in one step only, thus reducing the propagation of carry from LSB to MSB. Therefore, the overall time required for the operation gets reduce. In addition to that, the hardware complexity in terms of gate count also reduces. Thus, the Urdhva Tiryakbhyam sutra of Vedic mathematics can be used to implement high speed multipliers in digital signal processing algorithms.

REFERENCES:

[1] Jagadguru Swami Sri Bharati Krisna Tirthaji Maharaja, “Vedic Mathematics: Sixteen Simple Mathematical Formulae from the Veda,” Motilal Banarasidas Publishers, Delhi, 2009, pp. 5-45.

[2] Ganzorig Gankhuyag, Chan Mo Kim, Yong Beom Cho, “Multiplier design based on ancient Indian Vedic Mathematics”, 2008 IEEE International SoC Design Conference.

[3] V Jayaprakasan, S Vijayakumar, V S Kanchana Bhaaskaran,“Evaluation of the Conventional vs. Ancient Computation methodology for Energy Efficient Arithmetic Architecture”, IEEE International Conference April 2011

[4] Sandesh S. Saokar, R. M. Banakar, Saroja Siddamal, “High Speed Signed Multiplier for Digital Signal Processing Applications”, 2012 IEEE

[5] Oscal T. C. Chen, Sandy Wang, and Yi-Wen Wu “Minimization of Switching Activities of Partial Products for Designing Low-Power Multipliers” IEEE Transaction on VLSI System.Vol. 11, No.3, pp. 418-433, June 2010.

[6] Harpreet Singh Dhillon, Abhijit Mitra, “A Reduced-Bit Multiplication Algorithm for Digital Arithmetics,” International Journal of Computational and Mathematical Sciences, Spring 2008, pp.64-69.

[7] Sarita Singh, Sachin Mittal “VHDL Design and implementation for optimum delay and area for Multiplier and Accumulator unit by 32 bit Sequential Multiplier” International Journal of engineering Trends and Technology Volume3 issue 5-2012.

[8] S. S. Kerur, Prakash Narchi, Jayashree C .N., Harish M.Kittur and GirishV.A.“Implementation of Vedic Multiplier for Digital Signal Processing,” International Journal of Computer Applications, 2011, vol. 16, pp. 1-5.

[9] B.K.V.Prasad, P.Satishkumar, B.Stephencharles and T.Prasad, “Low Power Design of Wallace Tree Multiplier”, International Journal of Electronics and Communication Engineering & Technology (IJECET), Volume 3, Issue 3, 2012, pp. 258 - 264, ISSN Print: 0976- 6464, ISSN Online: 0976 –6472.

[10] Virendra Babanrao Magar, “Intelligent and Superior Vedic Multiplier for FPGA Based Arithmetic Circuits”, International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-3, Issue-3, July 2013.

[11] A. D. Booth, “A signed binary multiplication technique,” Q. J. Mech.Appl. Math., vol. 4, pp. 236–240, 1951.

[12] H. Thapliyal and M. B. Srinivas, “High Speed Efficient N × N Bit Parallel Hierarchical Overlay Multiplier Architecture Based on Ancient Indian Vedic Mathematics”, Enformatika Trans., vol. 2, pp. 225-228, Dec. 2004.

[13] A.P. Nicholas, K.R Williams, J. Pickles, “Application of Urdhava Sutra”, Spiritual Study Group, Roorkee (India),1984

[14] VHDL Primer by J. Bhaskar