International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 4 Issue 03, March-2015

Design and Implementation of 32bit Complex Multiplier using Vedic Algorithm

Ankush Nikam, Swati Salunke, Sweta Bhurse
Center for VLSI and Nanotechnology
Visvesvaraya National Institute of Technology
Nagpur, Maharashtra, India.

***Abstract*—This paper discusses the design of 32 bit Complex Multiplier using the techniques of Ancient Indian Vedic Mathematics that have been modified to improve performance. Vedic Mathematics is the ancient system of mathematics which has a unique technique of calculations based on 16 Sutras. The vedic method used here for complex multiplication is – "Urdhva Triyagbhyam" (vertically and Cross wise). Urdhva tiryakbhyam Sutra is the most efficient Sutra (Algorithm) that gives minimum delay for multiplication of small or large types of numbers. This paper also presents comparison of 8bit, 16bit, 32 bit complex multiplier on various performance parameters like power and delay. The proposed system is designed using VHDL and Verilog and is implemented through Xilinx ISE 14.2 navigator and modelsim v6.3 software's.**

***Keywords*—Vedic Mathematics, Urdhva Triyakbhyam sutra, Complex Multiplier, Ripple Carry Adder(RCA).**

I. INTRODUCTION

This template, Swami Bharati Krishna Tirthaji Maharaj, Shan-karacharya of Goverdhan Peath introduces the ancient system of Vedic Mathematics to the world. Vedic mathematics helps to reduce the complexity in calculations that exist in conventional mathematics. Generally there are sixteen sutras available in Vedic mathematics. Among them only two sutras are for multiplication operation. They are "Urdhva Triyakbhyam" sutra (is a Sanskrit word means vertically and crosswise) used for smaller number multiplication and "Nihilam Navatascaramam Dasatah" Sutra (Sanskrit word means all from 9 and last from 10) used for large number multiplication and subtraction. The logic behind Urdhva Triyakbhyam sutra is very much similar to the ordinary array multiplier.

Multiplication is one of the most important arithmetic operation functions, especially when implemented in Programmable Logic. Complex multiplication are key components for many high performance systems such as Microprocessors, Digital Signal Processors (DSP), Image Processing (IP), FIR filters etc. The multiplier generally determines the performance of a system as it is generally the slowest element in the system. Also it is the most area consuming module. Hence, optimizing the speed and area of the multiplier is a major design issue. Complex number multiplication is performed using four real number multiplications and two additions/ subtractions. In algorithmic and structural levels, various multiplication techniques has been developed to enhance the efficiency of the multiplier, which encounters the reduction of the partial products and/or the methods for their partial products addition, but the principle behind multiplication is same in all cases.

In this work we formulate this mathematics for designing the complex multiplier architecture. Here we have designed the 32 bit complex multiplier using Urdhva Triyakbhyam sutra. And
the rest of the paper is organized as follows - In section II we have described the design of 32 bit complex multiplier, section
III presents the experimental results, we conclude in section IV.

II. DESIGN OF COMPLEX MULTIPLIER

A. 32x32bit vedic multiplier

For any complex number multiplier design, the most important procedure is Multiplication. Here we have designed the 32 bit multiplier using vedic algorithm (Urdhva-tiryakbyham sutra). Fig.1 shows the block diagram of 32x32bit vedic multiplier module which is easily designed using four 16x16bit Vedic multipliers modules. Ripple Carry Adders(RCA) are used for the addition of the output of these four 16x16bit multiplier modules.

Suppose the two 32bit numbers are X [31:0] and Y [31:0].

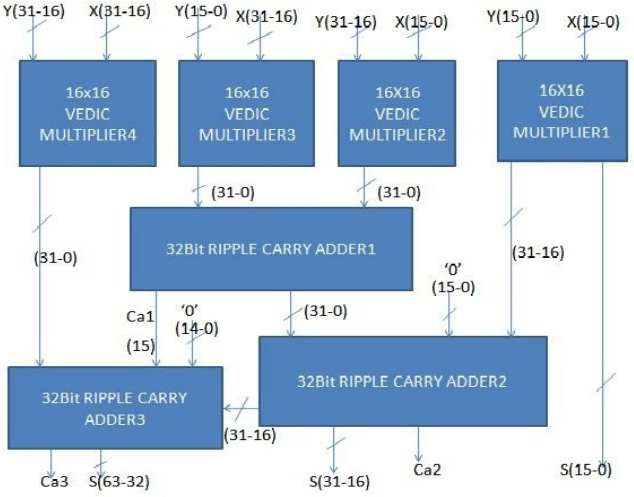

Fig. 1. Block Diagram of 32x32bit vedic multiplier

Each of these numbers are divided into two 16-bit numbers X [31:16]-X[15:0](XH-XL) and Y[31:16]-Y[15:0](YHYL) and given as a input to the 16x16bit vedic multiplier module. The input combinations for 1 to 4 16x16bit multiplier modules are XL-YL, XL-YH, XH-YL and XH-YH respectively. Each

multiplier gives the intermediate output of 32-bit. This intermediate output is then added using ripple carry adders (RCA). The output of second and third multiplier module is added using RCA1. The output of RCA1 (32-bit) and the higher order bits of first multiplier is then added using the RCA2 which gives the output S[31:16]. Finally the higher order bits of RCA2, the output of fourth multiplier and the carry from the RCA1 (the fifteenth bit position) are added to get the higher order bits S[63:32] of the final output of 32x32bit multiplier. The LSB bits S[15:0] of final output are obtained directly by taking the lower order bits of the output of first 16x16bit multiplier.

B. 32x32bit complex multiplier

Fig.2 shows the block diagram of 32x32 complex number multiplier. It requires four 32x32bit vedic multiplier modules and adders/substractors. Let (re_a+j im_a) and (re_b+j im_b) be the two 32 bit complex numbers.

$$re\_root+jim\_root = (re\_a+j\ im\_a)(re\_b+j\ im\_b) \quad (1)$$

Gauss's algorithm for complex number multiplication gives two separate final results to calculate real and imaginary part. From equation (1) the real part of the output can be computed using (re_a.re_b - im_a.im_b), and the imaginary part of the result can be computed using (re_a.im_b + re_b.im_a). Thus four separate multiplications and addition/subtraction are required to produce the real as well as imaginary part numbers.[5]

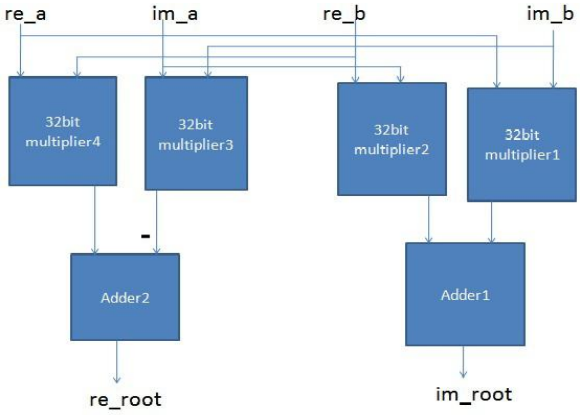

Fig. 2. Block Diagram of 32x32bit complex multiplier

III. RESULT AND COMPARISON

The 32x32bit complex multiplier using vedic algorithm is implemented using VHDL and verilog and functionally verified using Xilinx ISE 14.2 and Modelsim v6.3 simulators. Fig3 shows the RTL schematic of 32bit complex multiplier. The implemented 8bit and 16bit complex multiplier results are shown in Fig.4 and Fig.5 respectively. Fig6 shows the simulation result of implemented 32bit complex multiplication. The 8-bit, 16-bit and 32-bit complex multipliers are analyzed based on various performance factors such as delay and power. The comparison results and the device Utilization summary of these three multipliers are shown in Table I.

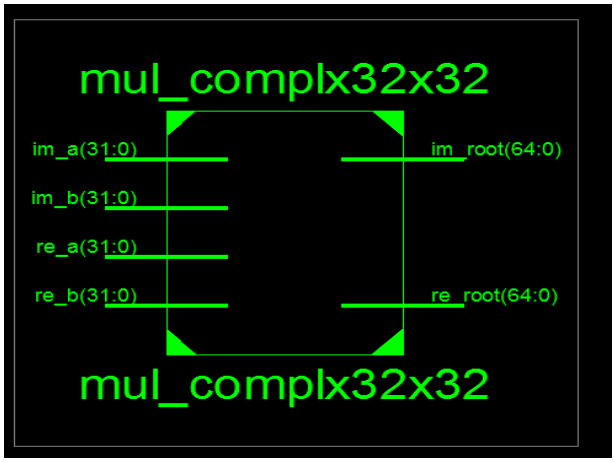

Fig. 3. RTL Schematic of 32bit Complex Multiplier

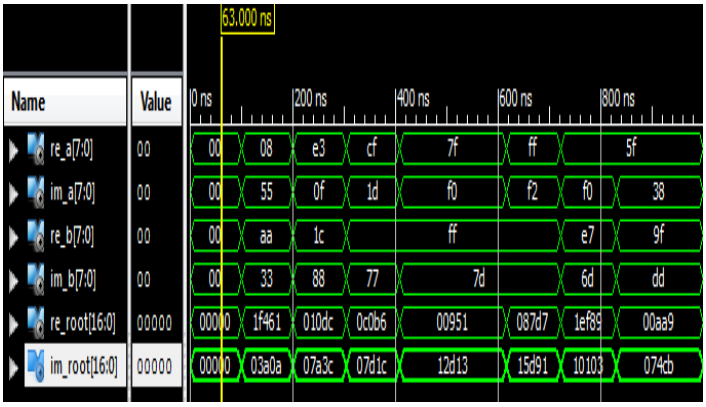

Fig. 4. Simulation result of implemented 8bit Complex multiplication

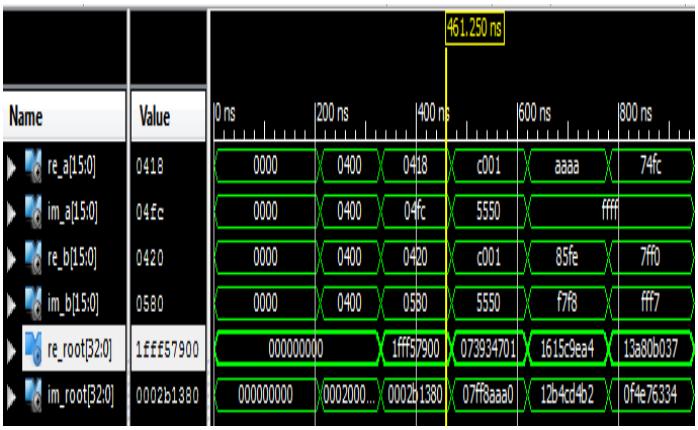

Fig. 5. Simulation result of implemented 16bit Complex multiplication

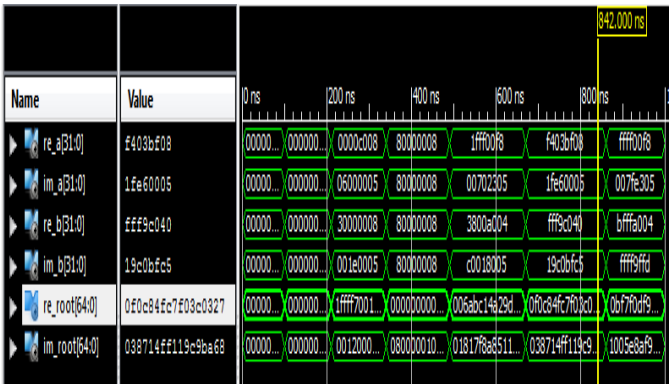

Fig. 6. Simulation result of implemented 32bit Complex multiplication

TABLE I. COMPARATIVE ANALYSIS AND DEVICE UTILIZATION SUMMARY OF 8BIT, 16BIT AND 32BIT COMPLEX MULTIPLIER

Parameters	8Bit	16Bit	32Bit
Delay(ns)	10	17.2	29.84
Power(mW)	47	52	62
No. of Slice LUTs	352	1,514	7,874
No. of bonded IOBs	66	130	258

VI. CONCLUSION

In this paper we have proposed architecture, capable of multiplying two 32bit complex numbers using Vedic Multiplier Algorithm. The multiplier algorithm is basically based on fundamentals of Ancient Indian Vedic Mathematics (Urdhva Tiryakbhyam sutra) and is implemented using Verilog and VHDL. The comparison of 8bit, 16bit, 32bit complex multiplier on various performance parameters like power and delay are discussed from which we infer that there is 10% and 19% increase in power as we go from 8bits to 16 bits and 16bits to 32bits respectively. Delay constrained varies nonlinearly as we increase the number of bits. The above results shows that Urdhva Tiryakbhyam sutra is used to implement complex multiplier efficiently in DSP algorithms by decreasing propagation delay (ns).

REFERENCES

[1] Saha, P., Banerjee, A., Bhattacharyya, P., and Dandapat, A. (2011, January). High speed ASIC design of complex multiplier using vedic mathematics. In Students' Technology Symposium (TechSym), 2011 IEEE (pp. 237-241). IEEE.

[2] Tiwari, H. D., Gankhuyag, G., Kim, C. M., and Cho, Y. B. (2008, November). Multiplier design based on ancient Indian Vedic Mathematics. In SoC Design Conference, 2008. ISOCC'08. International (Vol. 2, pp. II-65). IEEE.

[3] Ramalatha, M., Dayalan, K. D., Dharani, P., and Priya, S. D. (2009, July). High speed energy efficient ALU design using Vedic multiplication techniques. In Advances in Computational Tools for Engineering Applications, 2009. ACTEA'09. International Conference on (pp. 600 - 603). IEEE.

[4] Mehta, P., and Gawali, D. (2009, December). Conventional versus Vedic mathematical method for Hardware implementation of a multiplier. In Advances in Computing, Control, and Telecommunication Technologies, 2009. ACT'09. International Conference on (pp. 640-642). IEEE.

[5] Kong, M. Y., Langlois, J. P., and Al-Khalili, D. (2008, May). Efficient FPGA implementation of complex multipliers using the logarithmic number system. In Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on (pp. 3154-3157). IEEE.

[6] Paramasivam, M. E., and Sabeenian, R. S. (2010, February). An efficient bit reduction binary multiplication algorithm using Vedic methods. In Advance Computing Conference (IACC), 2010 IEEE 2nd International (pp. 25-28). IEEE.

[7] Patil, S., Manjunatha, D. V., and Kiran, D. (2014, October). Design of speed and power efficient multipliers using vedic mathematics with VLSI implementation. In Advances in Electronics, Computers and Communications (ICAECC), 2014 International Conference on (pp. 1-6) IEEE.