Design of Complex Multiplier WITH High Speed ASIC Using Vedic Mathematics

P.SREENIVASA RAO
M.Tech [VLSI], E.C.E
VITS, PRODDATUR

Mr.C.Md.ASLAM(phd)
Asst.professor,Dept.of ECE
VITS,PRODDATUR

ABSTRACT

Vedic Mathematics is the ancient methodology of Indian mathematics which has a unique technique of calculations based on 16 Sutras (Formulae). A high speed complex multiplier design (ASIC) using Vedic Mathematics is presented in this paper. The idea for designing the multiplier and adder Subtractor unit is adopted from ancient Indian mathematics "Vedas". On account of those formulas, the partial product 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 in comparison on with DA based architecture and parallel adder based Implementation which are most commonly used architectures. The functionality of these circuits was checked and performance parameters like propagation delay and dynamic power consumption were calculated by spice spectra using standard90nm CMOS technology. The propagation delay of the resulting(16, 16)x(16, 16) complex multiplier is only 4ns and consume 6.5mW power. We achieved almost 25% improvement in speed from earlier reported complex multipliers, e.g. parallel adder and DA based architectures

Keywords- Complex Multiplier, Exponent Determinant, High Speed Radix Selection Unit, Vedic Formulas.

I. INTRODUCTION

Complex multiplication is of immense importance in Digital Signal Processing (DSP) and Image Processing (IP).To implement the hardware module of Discrete Fourier Transformation (DFT), Discrete Cosine Transformation(DCT),Discrete Sine Transformation (DST) and modern broadband communications; large numbers of complex multipliers are required. Complex number multiplication is performed using four real number multiplications and two additions/ subtractions. In real number processing, carry needs to be propagated from the least significant bit (LSB) to the most significant bit (MSB) when binary partial products are added [1]. Therefore, the addition and subtraction after binary multiplications limit the overall speed. Many alternative method had so far been proposed for complex number multiplication [2-7] like algebraic transformation used implementation[2],bit-serial multiplication using offset binary and distributed arithmetic [3], the CORDIC (co ordinate rotation digital computer) algorithm [4], the quadratic residue number system (QRNS) [5], and recently, the redundant complex number system (RCNS) [6].Blahut et. al [2] proposed a technique for complex number multiplication, where the algebraic transformation was used. This algebraic transformation saves one real multiplication, at the expense of three additions as compared to the direct method implementation. A left to right array [7] for the fast multiplication has been reported in 2005, and the method is not further extended for complex multiplication. But, all the above techniques require either large overhead for pre/post processing or long latency. Further many design issues like as speed, accuracy, design overhead, power consumption etc., should not be addressed for fast multiplication [8].In algorithmic and structural levels, a lot of multiplication techniques had 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 was same in all cases.

Vedic Mathematics is the ancient system of Indian mathematics which has a unique technique of calculations based on 16 Sutras (Formulae). "Urdhva-tiryakbyham" is a Sanskrit word means vertically and crosswise formula is used for smaller number multiplication. "Nikhilam NavatascaramamDasatah" also a Sanskrit term indicating "all from 9 and last from 10", formula is used for large number multiplication and subtraction. All these formulas are adopted from ancient Indian Vedic Mathematics. In this work we formulate this mathematics for designing the complex multiplier architecture in transistor level with two clear goals in mind such as: i) Simplicity and modularity multiplications for VLSI implementations and ii) The elimination of carry propagation for rapid additions and subtractions. Mehta et al. [9] have been proposed a multiplier design using "Urdhva-tiryakbyham" sutras, which was adopted from the Vedas. The formulation using this sutra is similar to the modern array multiplication, which also indicating the carry propagation issues. A multiplier design using "Nikhilam Navatascaramam Dasatah" sutras has been reported by Tiwariet. al [10] in 2009, but he has not implemented the hardware module for multiplication. Multiplier implementation in the gate level (FPGA) using Vedic Mathematics has already been reported but to the best of our knowledge till date there is no report on transistor level(ASIC) implementation of such complex multiplier. By employing the Vedic mathematics, an N bit complex number multiplication was transformed into four multiplications for real and imaginary terms of the final product. "Nikhilam grow The calculated results revealed (16,16)x(16,16) complex multiplier have propagation delay only 4ns with 6.5mW dynamic switching power.In this paper we report on a novel high speed complex multiplier design using ancient Indian Vedic mathematics. The paper is organized as follows; Mathematical fonnulation ofthe Vedic sutras and its application towards multiplier isdescribed in section II; section III consists transistor levelhardware implementation for multiplication algorithm usingVedic mathematics; section IV representing proposedalgorithm for complex multiplier design section V representing the simulation results and finally Section VI representing the conclusions.

II. MATHEMATICAL FORMULATION OF VEDIC SUTRAS

The gifts of the ancient Indian mathematics in the world history of mathematical science are not well recognized. The contributions of saint and mathematician in the field of number theory, 'Sri BharatiKrsnaThirthaji Maharaja', in the form of Vedic Sutras (formulas) [11] are significant for calculations. He had explored the mathematical potentials from Vedic primers and showed that the mathematical operations can be carried out mentally to produce fast answers using the Sutras. In this paper we are concentrating on"Urdhvatiryakbyham",and"NikhilamNavatascaramamDa satah" formulas and other formulas are beyond the scope of this paper. Vedic Mathematics is the ancient system of Indian mathematics which has a unique technique of calculations based on 16 Sutras (Formulae). "Urdhva-tiryakbyham" is a Sanskrit word means vertically and crosswise formula is used for smaller number multiplication. "NikhilamNavatascaramamDasatah" also a Sanskrit term indicating "all from 9 and last from 10", formula is used for large number multiplication and subtraction. All these formulas are adopted from ancient Indian Vedic Mathematics.

A. "Urdhva-tiryakbyham " Sutra

The meaning of this sutra is "Vertically and crosswise" and it is applicable to all the multiplication operations. Fig.1represents the general multiplication procedure of the 4x4multiplication. This procedure is simply known as array multiplication technique [12]. It is an efficient multiplication technique when the multiplier and multiplicand lengths are small, but for the larger length multiplication this technique is not suitable because a large amount of carry propagation delays are involved in these cases. To overcome this problem we are describing Nikhilam sutra for calculating the Multiplication of two larger numbers.

B. "Nikhilam Navatascaramam Dasatah" Sutra

Nikhilam sutra means "all from 9 and last from 10".Mathematical description of this sutra can be formulated as: Assuming A and B are two n-bit numbers to be multiplied and their product is equals to P.

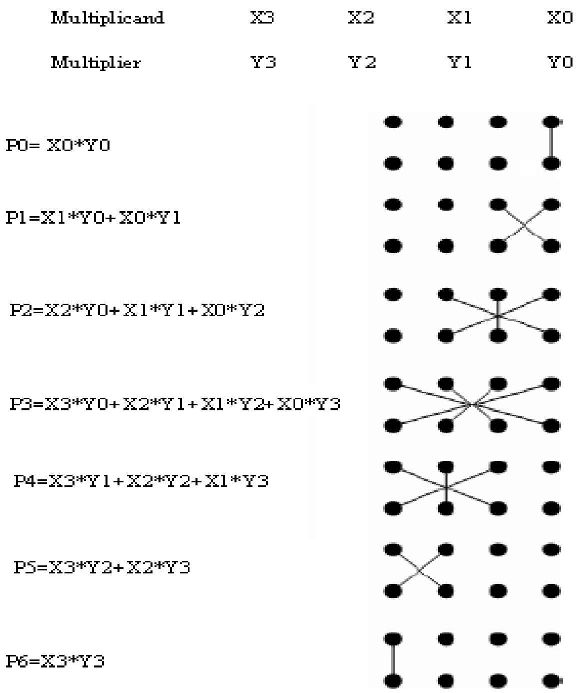
Multiplication Rule:

P=AB

The serious drawback of Nikhilam sutra can be summarized as:

(i) Both the multiplier and multiplicand are less or greater than the base.

(ii) Multiplier and multiplicand are nearer to the base



Fig, I: Multiplication procedure using "Urdhva-tiryakbyham " sutra

III.PROPOSEDMULT1PLlER ARCHITECTURE DESIGN

The mathematical expression for the proposed algorithm is shown below. Broadly this algorithm is divided into three parts. (i) Radix Selection Unit (ii) Exponent Determinant (iii)Multiplier. Consider two n bit numbers X and Y. kl and k2 are the exponent of X and Y respectively. X and Y can be represented

as:

X = Zk1 ± Zl

y = Zk2 ± Z2

For the fast multiplication using Nikhilam sutra the bases of the multiplicand and the multiplier would be same, (here we have considered different base) thus the equation can be

$$Y \times 2^{k_1-k_2} = 2^{k_1} \pm z_2 2^{k_1-k_2}$$

$$X \times Y \times 2^{k_1-k_2} = (2^{k_1} \pm z_1)(2^{k_1} \pm z_2 2^{k_1-k_2})$$

$$= 2^{2k_1} \pm z_1 2^{k_1} \pm z_2 2^{2k_1-k_2} \pm z_1 z_2 2^{k_1-k_2}$$

$$= 2^{k_1}(2^{k_1} \pm z_1 \pm z_2 2^{k_1-k_2}) \pm z_1 x_2 2^{k_1-k_2}$$

$$= 2^{k_1}(X \pm z_2 2^{k_1-k_2}) \pm z_1 z_2 2^{k_1-k_2}$$

$$P = XY = 2^{k_2}(X \pm z_2 2^{k_1-k_2}) \pm z_1 z_2$$

Hardware implementation of this mathematics is shown in Fig. 2. The architecture can be decomposed into three main subsections: (i) Radix Selection Unit (RSU) (ii) Exponent Determinant (ED) and (iii) Array Multiplier. The RSU is required to select the proper radices corresponding to the input numbers. If the selected radix is nearer to the given number then the multiplication of the residual parts (Zl xz2) can be easier to compute. The Subtractor blocks are required to extract the residual parts (ZI and Z2). The second subsection(ED) is used to extract the power (kl and k2) of the radix and it is followed by a subtractor to calculate the value of (k1-k2).The third subsection array multiplier [10] is used to Calculate the product (Zl xZ2). The output of the subtractor (klk2)and Z2 are fed to the shifter block to calculate the value ofZ2 x Zk1-k2.The first adder-subtractor block has been used to calculate the value of X ± Z2 x Zk1 -k2• The output of the first adder-subtractor and the output of the second Exponent Determinant (k2) are fed to the second shifter block to compute the value of Zk2 x (X ± Z2 X Zk1-k2). The output of the multiplier (ZI xz2) and the output of the second shifter(Zk2 x (X ± Z2 x Zk1-k2))are fed to the second adder subtractor block to compute the value of (Zk2 x (X ± Z2 xZk1-k2)) ± ZlZ2'

A. Mathematical expression for RSU

Consider an 'n' bit binary number X, and it can berepresented as

A=2n-1+2n/2

= zn-2 X 3

If X >A Then radix = 2n

If X< A Then radix = 2n-1

xy

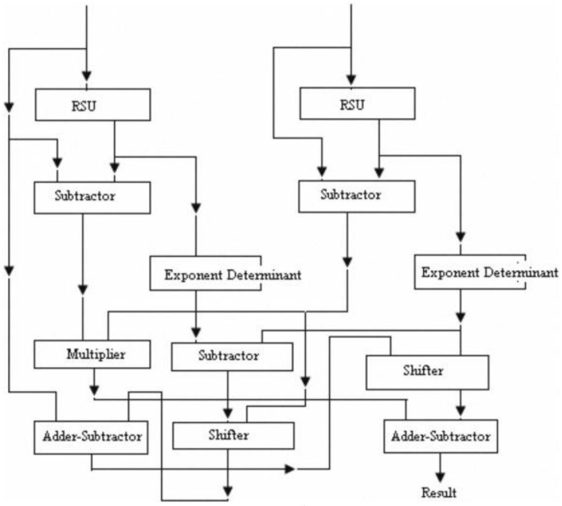


Fig. 2: Hardware implementation of "NikhiIamSutra "

The Block level architecture of RSU is shown in Fig.3.RSU consists of three main subsections: (i) Exponent Determinant(ED),(ii)MeanDeterminant(MD)and(iii)comparator. 'n' number bit from input X is fed to the ED block. The maximum power of X is extracted at the output which is again fed to shifter and the adder block. The second239input to the shifter is the (n+ I) bit representation of decimal'1 '.If the maximum power of X from the ED unit is (n-I) then the output of the shifter is i"-I). The adder unit is needed to increment the value of the maximum power of X by 'I'. The second shifter is needed to generate the value of 2".Here n is the incremented value taken from the adder block. The Mean Determinant unit is required to compute the mean of (zn-l+Zn). The Comparator compares the actual input with the meanvalue of (zn-l + zn). If the input is greater than the meanthen 2" is selected as the required radix. If the input is lessthan the mean then 2"-1 is selected as the radix. The select input to the multiplexer block is taken from the output of the comparator

B.ExponentDeterminant

The hardware implementation of the exponentdeterminant is shown in Fig. 4.The integer part or exponent of the number from the binary fixed point number can be obtained by the maximum power of the radix. For the non zero input, shifting operation is executed using parallel in parallel out (PIPO) shift registers. The number of select lines (in Fig A it is denoted as S], So) of the PIPO shifter is chosen as per the binary representation of the number (N-1)IO. 'Shift' pin is assigned in PIPO shifter to check whether the number isto be shifted or not (to initialize the operation 'Shift' pin is initialized to low). A decrementer [13] has been integrated in this architecture to follow the maximum power of the radix. A sequential searching procedure has been implemented here to search the first 'I' starting from the MSB side by using shifting technique. For an N bit number, the value (N-I)is fed to the input of decrementer. The decrementer is decremented based on a control signal which is generated by the searched result. If the searched bit is '0' then the control signal becomes low then decrementer start decrementing the input value (Here the decrementer is operating in active low logic). The searched bit is used as a controller of the decrementer. When the searched bit is 'I' then the control signal becomes high and the decrementer stops further decrementing and shifter also stops shifting operation. The output of the decrementer shows the integer part (exponent) of

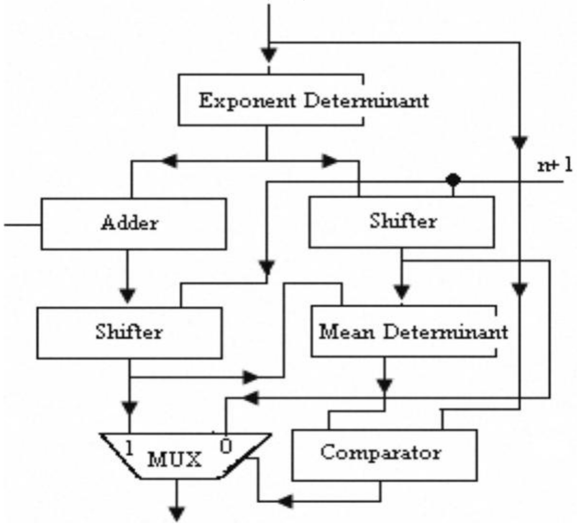


Fig. 3: Hardware implementation of RSU

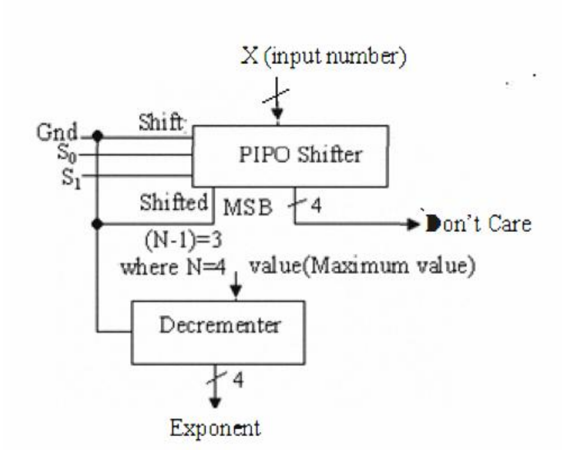


Fig. 4: Hardware implementation of exponent determinant

IV. COMPLEX MULTIPLIER

Complex multiplier design by using parallel adders and subtractors has been designed by Saha [I]. Fig. 5 shows the direct method for implementation of the complex multiplier design. In this paper complex multiplier design is done using Vedic Multipliers and Vedic subtractors. Multiplication Algorithm

<Input>

A and B : Multiplicand and multiplier respectively. Both are complex numbers .A=Ar +jA; and B =Br +jB;(here all are N Bit unsigned numbers).Stepl. Select the appropriate base using RSU.Step2. Multiply the numbers as shown in Fig 15 and then add or subtract them to obtain the real and the imaginary part of the result .

<Output>

Result :Cr and C; are the real and imaginary part ofcomplex number.

V. SIMULATION RESULT ANALYSIS

All the algorithm of this paper was simulated and their functionality was examined by using Spice Spectre. Performance parameters such as propagation delay and power consumptions analysis of this paper using standard 90nmCMOS technology. To evaluate the performance parameters, we give the values of the computational effort using array multiplier and Vedic multiplier. As shown, the application of the Vedic method for multiplication cuts the amount of the hardware as well as increases the performance parameters such as propagation delay, dynamic switching power consumptions, and dynamic leakage power consumptions. The performance parameters analysis using array multiplication and Vedic multiplication . Input data is taken as a regular fashion for experimental purpose. We have kept our main concentration for reducing the propagation delay, dynamic switching power and dynamic leakage power consumption and energy delay product.

A comparison between different architecture in terms of propagation delay and dynamic switching power consumption

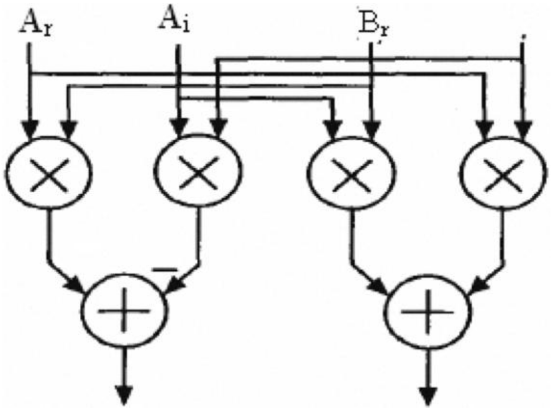


Fig.5: Implementation method of complex multiplier

The proposed complex number multiplier offered 20% and 19% improvement in terms of propagation delay and power consumption respectively, in comparison with parallel adder based implementation. Whereas, the corresponding improvement in terms of delay and power was found to be 33% and 46% respectively, with reference to the algebraic transformation information

VI. CONCLUSION

In this paper we report on a novel complex number multiplier design based on the formulas of the ancient Indian Vedic Mathematics, highly suitable for high speed complex arithmetic circuits which are having wide application in VLSI signal processing. The implementation was done in Spice spectra and compared with the mostly used architecture like distributed arithmetic, parallel adder based implementation, and algebraic transformation based implementation. This novel architecture combines the advantages of the Vedic mathematics for multiplication which encounters the stages and partial product reduction. The proposed complex number multiplier offered20% and 19% improvement in terms of propagation delay and power consumption respectively, in comparison with parallel adder based implementation. Whereas, the corresponding improvement in terms of delay and power was found to be33% and 46% respectively, with reference to the algebraic transformation based implementation. Future work is in progress regarding the layout of (16,16)x(16,16) bit complex number multiplier.

REFERENCES

[I] P. K. Saha, A. Banerjee, and A. Dandapat, "High Speed Low PowerComplex Multiplier Design Using Parallel Adders and Subtractors,"
International Journal on Electronic and Electrical Engineering,(/JEEE), vol 07, no. II, pp 38-46, Dec. 2009.

[2] R. E. Blahut, Fast Algorithms for Digital Signal Processing, Reading,
MA: Addison-Wesley, 1987.

[3] S. He, and M. Torkelson, "A pipelined bit-serial complex multiplier
using distributed arithmetic," in proceedings IEEE International
Symposium on Circuits and Systems, Seattle, W A, April 30-May-03,

1995,pp. 2313-2316.
[4] J. E. VoIder, "The CORDIC trigonometric computing technique," IRE Trans. Electron. Comput., vol. EC-8, pp. 330-334, Sept. 1959.
[5] R. Krishnan, G. A. Jullien, and W. C. Miller, "Complex digital signal
processing using quadratic residue number systems," IEEE Trans. Acoust., Speech, Signal Processing, vol. 34, Feb. 1985.
[6] T. Aoki, K. Hoshi, and T. Higuchi, "Redundant complex arithmetic andits application to complex multiplier design," in Proceedings 29th IEEE International symposium on Multiple-Valued-Logic, Freiburg, May 20-22,1999, pp. 200-207.
[7] Z. Huang, and M. D. Ercegovac, "High-Performance Low-Power Leftto-Right Array Multiplier Design," IEEE Transactions on Computers, vol 54, no. 3, pp 272-283, March 2005.
[8] S. Y. Kung, H. J. Whitehouse, and T. Kailath, VLSI and Modern SignalProcessing, Englewood Cliffs, NJ: Prentice-Hall, 1985.
[9] P. Mehta, and D. Gawali, "Conventional versus Vedic mathematical method for Hardware implementation of a multiplier," in Proceedings IEEE International Cotiference on Advances in Computing, Control, and Telecommunication Technologies, Trivandrum, Kerala, Dec. 28-29,2009, pp. 640-642.
[IO] H. D. Tiwari, G. Gankhuyag, C. M. Kim, and Y. B. Cho, "Multiplier design based on ancient Indian Vedic Mathematics," in Proceedings IEEE International SoC Design Cotiference, Busan, Nov. 24-25, 2008,
pp. 65-68.
[11] J. S. S. B. K. T. Maharaja, Vedic mathematiCS, Delhi: Motilal Banarsidass Publishers Pvt Ltd, 200 I.
[12] C. S. Wallace, "A suggestion for a fast multiplier," lEETrans.ElectronicComput., vol. EC-\3, pp. 14-17, Dec. 1964.
[I3] P. K. Saha, A. Banerjee, and A. Dandapat, "High Speed Low Power
Factorial Design in 22nm Technology," in Proceedings AlP International Cotiference on Nanomaterials and Nanotechnology,
Guwahati, 2009, pp. 294-301.
.