Vedic Multiplier

EE18MTECH11006,EE18MTECH11007

April 25, 2019

1 Abstract

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 modem 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 . Therefore, the addition and subtraction after binary multiplications limit the overall speed. Many alternative method had so far been proposed for complex number multiplication like algebraic transformation based implementation, bit-serial multiplication using offset binary and distributed arithmetic , the CORDIC (coordinate rotation digital computer) algorithm , the quadratic residue number system (QRNS) , and recently, the redundant complex number system (RCNS) .

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 adderlsubtractor unit is adopted from ancient Indian mathematics "Vedas". On account of those 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 in comparison 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 spectre using standard 90nm CMOS technology. The propagation delay of the resulting (16, 16)x(16, 16) complex multiplier is only 4ns and consume 6.5 mW power. We achieved almost 25% improvement in speed from earlier reported complex multipliers, e.g. parallel adder and DA based architectures.

This project proposes the hardware implementation of vlsi architecture for High Speed VLSI Design of Complex Multiplier Using Vedic Mathematics that have been modified to improve performance.This project was implemented in verilog the coding is done in Verilog HDL and the FPGA synthesis is done using Xilinx Spartan library. The main advantage of implementation using Hardware based High Speed VLSI Design of Complex Multiplier Using Vedic Mathematics its inherent speed over Software based methods. This speed is due to Flexibility of Reconfigurability and Reprogrammability of FPGA. The speed advantage makes the hardware based High Speed VLSI Design of Complex Multiplier Using Vedic Mathematics is a prime candidate for real time applications.

Tools Used:

Modelsim 6.5e
Verillog HDL
Arduino

Applications:

High Speed Data Manipulations
DSP applications

2 History Of Vedic Mathematics:-

Vedic mathematics is part of four Vedas (books of wisdom). It is part of Sthapatya- Veda (book on civil engineering and architecture), which is an upa-veda (supplement) of Atharva Veda. It covers explanation of several modern mathematical terms including arithmetic, geometry (plane, co-ordinate), trigonometry, quadratic equations, factorization and even calculus. His Holiness Jagadguru Shankaracharya Bharati Krishna Teerthaji Maharaja (1884-1960) comprised all this work together and gave its mathematical explanation while discussing it for various applications. Swahiji constructed 16 sutras (formulae) and 16 Upa sutras (sub formulae) after extensive research in Atharva Veda. Obviously these formulae are not to be found in present text of Atharva Veda because these formulae were constructed by Swamiji himself. Vedic mathematics is not only a mathematical wonder but also it is logical. That's why VM has such a degree of eminence which cannot be disapproved. Due these phenomenal characteristic, VM has already crossed the boundaries of India and has become a leading topic of research abroad. VM deals with several basic as well as complex mathematical operations. Especially, methods of basic arithmetic are extremely simple and powerful. The word Vedic is derived from the word , Veda which means the store-house of all knowledge. Vedic mathematics is mainly based on 16 Sutras (or aphorisms) dealing with various branches of mathematics like arithmetic, algebra, geometry etc. These Sutras along with their brief meanings are enlisted below alphabetically.

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 Bharati Krishna Thirthaji Maharaja', in the onn of Vedic Sutras (fonnulas) [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 Urdhva-tiryakbyham", and "Nikhilam NavatascaramamDasatah" fonnulas and other fonnulas are beyond the scope of this paper.

1) (Anurupye) Shunyamanyat – If one is in ratio, the other is zero.

2) Chalana-Kalanabyham – Differences and Similarities.

3) Ekadhikina Purvena – By one more than the previous One.

4) Ekanyunena Purvena – By one less than the previous one.

5) Gunakasamuchyah – The factors of sum is equal to sum of the factors.

6) Gunitasamuchyah – The product of sum is equal to sum of the product.

7) Nikhilam Navatashcaramam Dashatah – All from 9 and last from 10.

8) Paraavartya Yojayet – Transpose and adjust.

9) Puranapuranabyham – By the completion or noncompletion.

10) Sankalana- vyavakalanabhyam – By addition and by subtraction.

11) Shesanyankena Charamena – The remainders by the last digit.

12) Shunyam Saamyasamuccaye – When sum is same that sum is zero.

13) Sopaantyadvayamantyam – The ultimate and twice the penultimate.

14) Urdhva-tiryakbhyam – Vertically and crosswise.

15) Vyashtisamanstih – Part and Whole.

16) Yaavadunam – Whatever the extent of its deficiency.

• These methods and ideas can be directly applied to trigonometry, plain and spherical geometry, conics, calculus (both differential and integral), and applied mathematics of various kinds. As mentioned earlier, all these Sutras were reconstructed from ancient Vedic texts early in the last century. Many Sub-sutras were also discovered at the same time, which are not discussed here.

• The beauty of Vedic mathematics lies in the fact that it reduces the otherwise cumbersome-looking calculations in conventional mathematics to a very simple one. This is so because the Vedic formulae are claimed to be based on the natural principles on which the human mind works. This is a very interesting field and presents some effective algorithms which can be applied to various branches of engineering such as computing and digital signal processing.

• The multiplier architecture can be generally classified into three categories. First is the serial multiplier which emphasizes on hardware and minimum amount of chip area. Second is parallel multiplier (array and tree) which carries out high speed mathematical operations. But the drawback is the relatively larger chip area consumption. Third is serial- parallel multiplier which

serves as a good trade-off between the times consuming serial multiplier and the area consuming parallel multipliers.

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 modem 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. Therefore, the addition and subtraction after binary multiplications limit the overall speed. Many alternative method had so far been proposed for complex number multiplication like algebraic transformation based implementation, bit-serial multiplication using offset binary and distributed arithmetic the CORDIC (coordinate rotation digital computer) algorithm [the quadratic residue number system (QRNS) [, and recently, the redundant complex number system (RCNS). lahut et 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 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/postprocessing or long latency. Further many design issues like as speed, accuracy, design overhead, power consumption etc., should not be addressed for fast multiplication.

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). "Urdhvatiryakbyham" is a Sanskrit word means vertically and crosswise formula is used for smaller number multiplication. "Nikhilam Navatascaramam Dasatah" 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

ii) The elimination of carry propagation for rapid additions and subtractions. Mehta et al. have been proposed a multiplier design using "Urdhvatiryakbyham" sutras, which was adopted from the Vedas. The formulation using this sutra is similar to the modem array multiplication, which also indicating the carry propagation issues.

3 ALGORITHMS OF VEDIC MULTIPLIER

Urdhva Tiryakbhyam sutra:

The multiplier is based on an algorithm Urdhva Tiryakbhyam (Vertical Crosswise) of ancient Indian Vedic Mathematics. Urdhva Tiryakbhyam Sutra is a general multiplication formula applicable to all cases of multiplication. It literally means "Vertically and crosswise". It is based on a novel concept through which the generation of all partial products can be done with the concurrent addition of these partial products. The parallelism in generation of partial products and their summation is obtained using Urdhava Triyakbhyam explained in fig 2.1. The algorithm can be generalized for n x n bit number. Since the partial products and their sums are calculated in parallel, the multiplier is independent of the clock frequency of the processor. Thus the multiplier will require the same amount of time to calculate the product and hence is independent of the clock frequency. The net advantage is that it reduces the need of microprocessors to operate at increasingly high clock frequencies. While a higher clock frequency generally results in increased processing power, its disadvantage is that it also increases power dissipation which results in higher device operating temperatures. By adopting the Vedic multiplier, microprocessors designers can easily circumvent these problems to avoid catastrophic device failures. The processing power of multiplier can easily be increased by increasing the input and output data bus widths since it has a quite a regular structure. Due to its regular structure, it can be easily layout in a silicon chip. The Multiplier has the advantage that as the number of bits increases, gate delay and area increases very slowly as compared to other multipliers. Therefore it is time, space and power efficient. It is demonstrated that this architecture is quite efficient in terms of silicon area/speed.

1) Multiplication of two decimal numbers- 325*738

To illustrate this multiplication scheme, let us consider the multiplication of two decimal numbers (325 * 738). Line diagram for the multiplication is shown in Fig.1. The digits on the both sides of the line are multiplied and added with the carry from the previous step. This generates one of the bits of the result and a carry. This carry is added in the next step and hence the process goes on. If more than one line are there in one step, all the results are added to the previous carry. In each step, least significant bit acts as the result bit and all other bits act as carry for the next step. Initially the carry is taken to be zero. To make the methodology more clear, an alternate illustration is given with the help of line diagrams in figure 1 where the dots represent bit 0 or 1.

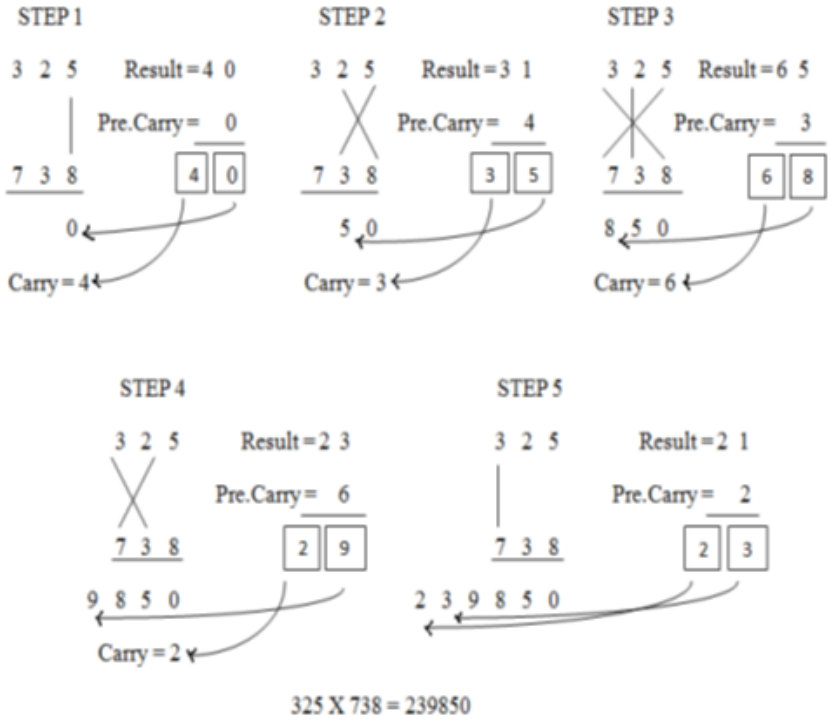

Fig. 1: Multiplication of two decimal numbers by Urdhva Tiryakbhyam.

2) Algorithm for 4 x 4 bit Vedic multiplier Using Urdhva Tiryakbhyam (Vertically and crosswise) for two Binary numbers -

CP = Cross Product (Vertically and Crosswise)
X3 X2 X1 X0 Multiplicand
Y3 Y2 Y1 Y0 Multiplier

H G F E D C B A

P7 P6 P5 P4 P3 P2 P1 P0 Product

PARALLEL COMPUTATION METHODOLOGY

1. CP X0 = X0 * Y0 = A

   Y0

2. CP X1 X0 = X1 * Y0+X0 * Y1 = B

   Y1 Y0

3. CP X2 X1 X0 = X2 * Y0 +X0 * Y2 +X1 * Y1 = C

Y2 Y1 Y0

4. CP X3 X2 X1 X0 = X3 * Y0 +X0 * Y3+X2 * Y1 +X1 * Y2 = D

Y3 Y2 Y1 Y0

5. CP X3 X2 X1 = X3 * Y1+X1 * Y3+X2 * Y2 = E

Y3 Y2 Y1

6. CP X3 X2 = X3 * Y2+X2 * Y3 = F

Y3 Y2

7 CP X3 = X3 * Y3 = G

Y3

3) Algorithm for 8 X 8 Bit Multiplication Using Urdhva Triyakbhyam (Vertically and crosswise) for two Binary numbers-

A = A7A6A5A4 A3A2A1A0

X1 X0

B = B7B6B5B4 B3B2B1B0

Y1 Y0

X1 X0
* Y1 Y0

F E D C

CP = X0 * Y0 = C
CP = X1 * Y0 + X0 * Y1 = D
CP = X1 * Y1 = E
Where CP = Cross Product.

Note: Each Multiplication operation is an embedded parallel 4x4 Multiply module.

To illustrate the multiplication algorithm, let us consider the multiplication of two binary numbers a3a2a1a0 and b3b2b1b0. As the result of this multiplication would be more than 4 bits, we express it as... r3r2r1r0. Line diagram for multiplication of two 4-bit numbers is shown in Fig. 2.2 which is nothing but the mapping of the Fig.2.1 in binary system. For the simplicity, each bit is represented by a circle. Least significant bit r0 is obtained by multiplying the least significant bits of the multiplicand and the multiplier. The process is followed according to the steps shown in figure.

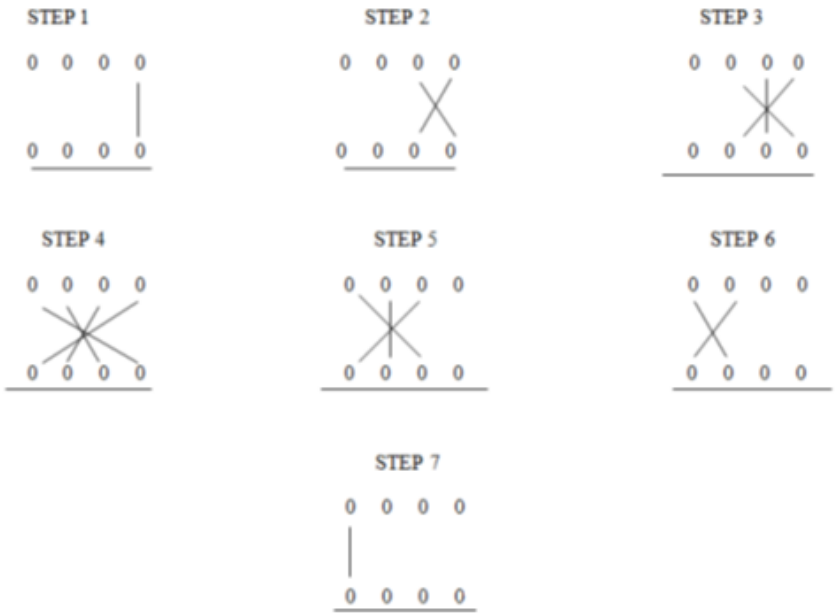

Fig. 2: Line diagram for multiplication of two 4 - bit numbers..

Firstly, least significant bits are multiplied which gives the least significant bit of the product (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. For example, if in some intermediate step, we get 110, then 0 will act as result bit (referred as rn) and 11 as the carry (referred as cn). It should be clearly noted that cn may be a multi-bit number. Thus we get the following expressions:

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

With c6r6r5r4r3r2r1r0 being the final product. Hence this is the general mathematical formula applicable to all cases of multiplication.

4 Implementation

Two inputs are given to Arduino via Keypad. From Arduino inputs are send to Ico Board. Multiplication is done here using Vedic Multiplier algorithm. Output is sent back to Arduino from Icoboard and displayed on LCD.

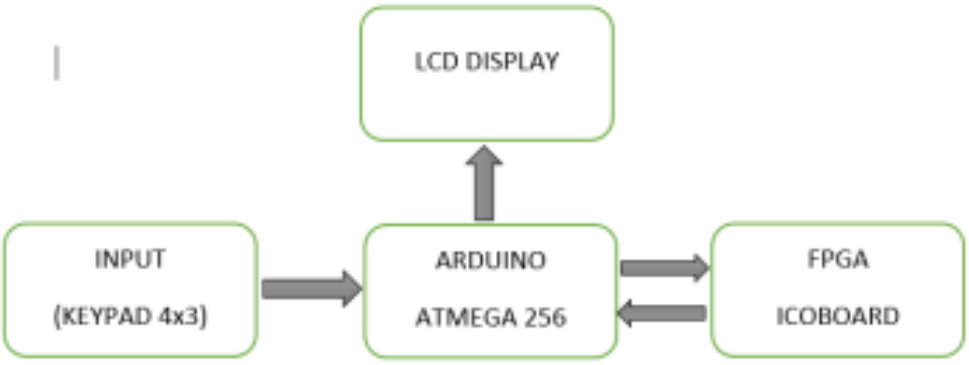

Fig. 3: Block diagram for vedic multiplier

5 References

[1] P. K. Saha, A. Banerjee, and A. Dandapat, "High Speed Low Power Complex 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 and its 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 Signal Processing, 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," lEE Trans. Electronic Comput., vol. EC-, 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.