International Journal of Advanced Research in Engineering and Technology (IJARET)
Volume 9, Issue 3, May - June 2018, pp. 109–118, Article ID: IJARET_09_03_015
Available online at http://www.iaeme.com/IJARET/issues.asp?JType=IJARET&VType=9&IType=3
ISSN Print: 0976-6480 and ISSN Online: 0976-6499

IMPLEMENTATION OF OPTIMIZED MULTIPLIER-ACCUMULATOR (MAC) UNIT WITH VEDIC MULTIPLIER AND FULL PIPELINED ACCUMULATOR: A REVIEW

M. Nasiruddin
Research Scholar MIET, Gondia

Dr. Prashant Sharma
Assistant Professor, MIET, Gondia

Dr. Vijay Chaurasia
Assistant Professor, MIET, Gondia

ABSTRACT

As the technology is scaling down from micro scale to nano scale. At such scale a new field is evolved is called quantum computing. Quantum computation based on the principle of reversible operation, means the information is conversed and performs certain task in nanosecond. In order to implement a high speed multiplier a Vedic algorithm can be applied, because it perform simple operation and yield result quickly. The multiplication process involves two step generation of partial product and addition of partial product, these two steps are concurrently perform by the Urdhva Tiryakbhyam algorithm of Vedic Mathematics. The Multiplier and Accumulator (MAC) are the necessary elements of the digital signal processing for example filtering, convolution, and transformations. Power dissipation is recognized as a critical parameter in modern the objective of a good multiplier is to provide a physically compact, good speed and low power consuming chip. First, we introduce the concept of Vedic Multiplier. Then, we review existing methods to implement Vedic Multiplier for Accumulator design. We also discuss and analyze the advantages and disadvantages of these methods. Finally, we identify the potential challenges and future research directions in location prediction.

Key words: Adders, Multiply Accumulate Unit, Vedic mathematics, Vedic multiplier.

Cite this Article: M. Nasiruddin, Dr. Prashant Sharma and Dr. Vijay Chaurasia, Implementation of Optimized Multiplier-Accumulator (MAC) unit with Vedic Multiplier and Full Pipelined Accumulator: A Review. *International Journal of Advanced Research in Engineering and Technology*, 9(3), 2018, pp 109–118.
http://www.iaeme.com/IJARET/issues.asp?JType=IJARET&VType=9&IType=3

1. INTRODUCTION

With the recent speedy advances in multimedia and communication systems, the demand of real-time signal processing like audio signal processing, video/image processing, or large-capacity data processing are increasing day by day. The multiplier-accumulator (MAC) is the important elements of the digital signal processing such as filtering, convolution, transformations and Inner products. Power dissipation is one of the most critical parameter in modern design. The objective of a good multiplier is to provide a physically compact, good speed and low power consuming chip.

Vedic multiplier is a encouraging solution in today's world for wide range of applications because of its simple architecture and increased speed which forms an unmatched combination for serving any complex multiplication computations. The Vedic multiplier needs very less area as compared to other multiplier architecture. Hence the Vedic multiplier is faster than the array multiplier, booth multiplier and Wallace multiplier. MAC always lies in the critical path that determines the speed of the overall hardware systems. Due to its regular and parallel structure it can be realized easily on silicon as well. [1]

2. MAC UNIT

In Digital Signal Processing the Multiplier-Accumulator (MAC) operation is the very critical operation not only in DSP applications but it is also critical in multimedia information processing and various other applications. MAC unit consist of multiplier, adder and accumulator. The MAC unit determines the speed of overall system as it is always lies in the critical path. To develop high speed MAC unit essential for real time DSP application. In order to improve the speed of the MAC unit, there are two major factors that need to be considered. The first one is the fast multiplication network and the second one is the accumulation. Both of these stages require addition of large operands that involve long paths for carry propagation. In recent MAC, accumulation and addition are merging to save the time and power. The MAC unit mostly do the multiplication of two numbers, multiplier and multiplicand, and add this product with result stored in the accumulator. For high speed MAC unit, faster adder and multiplier circuits are required. Figure 1 shows the basic Structure of MAC unit.

General Architecture of a MAC unit is shown in to the Figure 1. MAC unit consists of 1. A multiplier 2. An accumulator containing the sum of the previous successive products. The MAC inputs are taken from the memory location and then given to the multiplier block. [2]

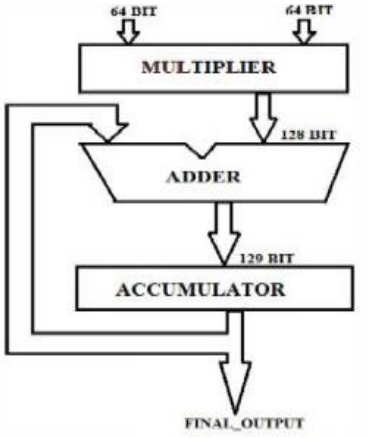

Figure 1 Basic Structure of MAC unit

3. PRINCIPLES OF VEDIC MATHEMATICS

The Sanskrit word Veda is derived from the root Vid, meaning of this is to know without limit. The word Veda describes all the Veda-sakhas known to humans. The Veda is a source of all knowledge, incalculable, ever revealing as it is investigated shallower. Swami Bharati Krishna Tirtha who was Sankaracharya of Goverdhan matha puri from 1925-1960, picked a set of 16 Sutras (aphorisms) and 13 Sub - Sutras (corollaries) from the Atharva Veda. He established methods and techniques for intensifying the principles contained in the aphorisms and their corollaries, and called it Vedic Mathematics.

These Sutras can apply to every branch of Mathematics and also covers all branches of Mathematics. These sutras can be applied even to the complex problems including a large number of mathematical operations. Application of these Sutras saves a lot of time and effort in solving the complex mathematical problems as compared to the formal methods presently in use. Though the solutions of vedic mathematics seems like magic, the application of the Sutras is perfectly logical and rational. The calculations made on the computers follows, in a way, the principles underlying the Sutras. The Sutras provides methods of calculation as well as it also improves the ways of thinking for their application. [3]

4. MULTIPLICATION ALGORITHMS

4.1. Booth's Multiplier

Booth's multiplication algorithm was developed by Andrew Donald Booth in the year 1950 while doing research. This algorithm performs the multiplication of two signed binary numbers in two's complement representation. Booth's algorithm is very useful in the study of computer architecture.

Booth's algorithm inspects neighboring pairs of bits of the 'N'-bit multiplier Y, in signed two's complement representation, including an inherent bit below the least significant bit, y-1 = 0. For each bit y_i, for i goes from 0 to N-1, the bits y_i and y_{i-1} are considered. Where these two bits are equal, the product accumulator P is left unchanged. Where $y_i = 0$ and $y_{i-1} = 1$, the multiplicand times 2i is added to P; and where $y_i = 1$ and $y_{i-1} = 0$, the multiplicand times 2i is subtracted from P. The final value of P is the signed product. The last estimation of P is the marked item. The representation of the multiplicand and item are not determined; commonly, these are both additionally in two's supplement representation, similar to the multiplier, yet any number framework that backings expansion and subtraction will fill in too. As expressed here, the request of the strides is not decided. Regularly, it continues from LSB to MSB, beginning at i = 0; the duplication by 2i is then commonly supplanted by incremental moving of the P collector to the directly between steps; low bits can be moved out, and resulting increments and subtractions should then be possible just on the most elevated N bits of P. There are numerous varieties and advancements on these points of interest. The calculation is regularly portrayed as changing over strings of 1's in the multiplier to a high-arrange +1 and a low-arrange –1 at the closures of the string. At the point when a string goes through the MSB, there is no high-arrange +1, and the net impact is translation as a negative of the fitting quality.

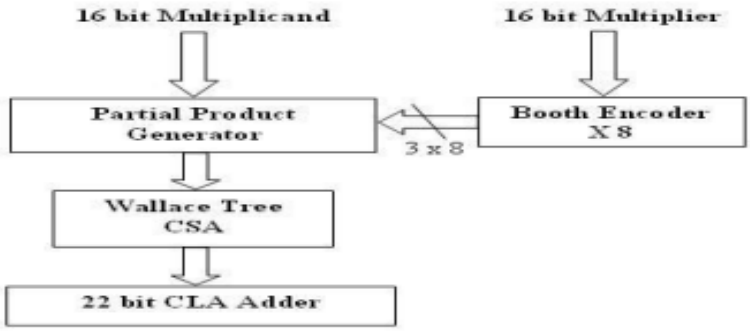

Figure 2 Block Diagram of Booth's Multiplier

4.2. Urdhva Tiryakbhyam Sutra

- Take the right hand digit and multiply them together. This will give LSB digit of the answer.
- Multiply LSB digit of the top number by the second bit of the bottom number and the LSB of the bottom number by the second bit of the top number. Once we have those value add them together.
- Multiply the LSB digit of bottom number with the MSB digit of the top one.
- This step is Similar to the second step, just move one place to the left. We will multiply the second digit of one number by the MSB of the other number.
- Finally, simply multiply the LSB of the both number together to get the final product.

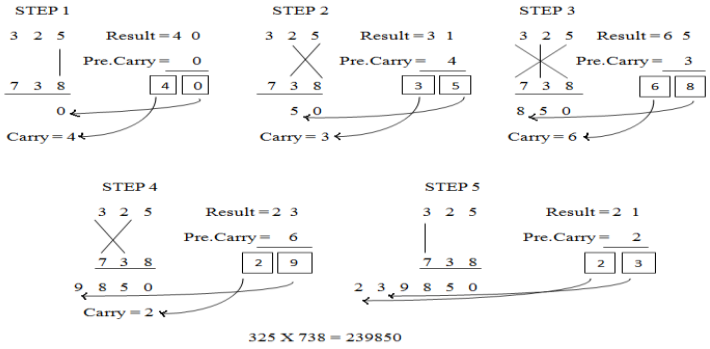

Figure 3 Multiplication Example of two decimal numbers using Urdhva Tiryakbhyam Sutra

4.3. Multiplication using Nikhilam Sutra

NIKHILAM Sutra truly signifies "all from 9 and last from 10". Despite the fact that it is appropriate to all instances of increase, it is more productive when the numbers included are extensive. Since it figures out the compliment of the huge number from its closest base to perform the expansion operation on it, bigger is the first number, lesser the intricacy of the increase. We first show this Sutra by considering the duplication of two decimal numbers (96 * 93) where the picked base is 100 which is closest to and more prominent than both these two numbers. [4]

5. MULTIPLIER ARCHITECTURE

Here we propose, "Urdhva-Tiryagbhyam" (Vertically and Crosswise) sutra to compare architectures for the multiplication of two binary numbers. The beauty of Vedic multiplier is that here partial product generation and additions are done concurrently. Hence, it is well

adapted to parallel processing. The feature makes it more attractive for binary multiplications. This in turn reduces delay.

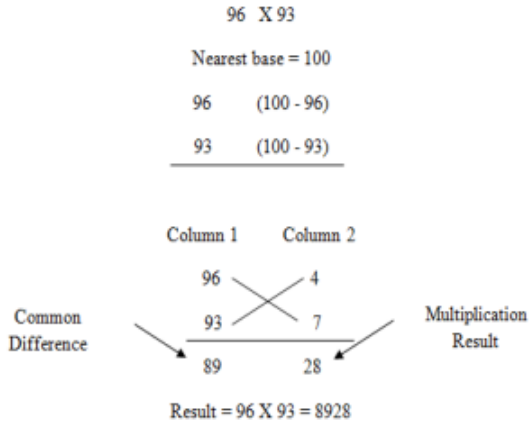

Figure 4 Multiplication Example using Nikhilam Sutra

5.1. 2x2 bit Module for Vedic Multiplier

The method is explained below for two, 2 bit numbers A and B where A = a_1 a_0 and B = b_1 b_0 as shown in Fig. 5. Initially, the least significant bits are multiplied which gives the least significant bit of the final 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 final product and the carry is added with the partial product obtained by multiplying the most significant bits to give the sum and carry. The sum is the third corresponding bit and carry becomes the fourth bit of the final product.

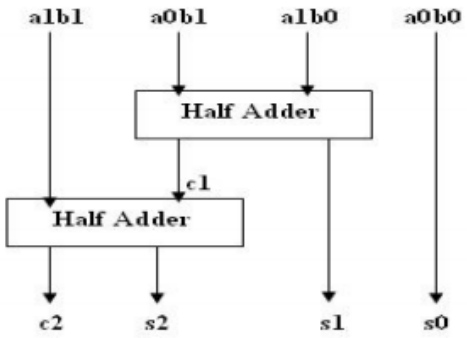

Figure 5 Block Diagram of 2x2 bit Vedic Multiplier

5.2. 4x4 bit Module for Vedic Multiplier

The 4x4 bit Vedic multiplier module is designed using four 2x2 bit Vedic multiplier modules as discussed in Fig. 3. Let's analyze 4x4 multiplications, say A=A_3 A_2 A_1 A_0 and B=B_3 B_2 B_1 B_0. The output line for the multiplication result is – S_7 S_6 S_5 S_4 S_3 S_2 S_1 S_0 Let's divide A and B into two parts, say A_3 A_2& A_1 A_0 for A and B_3 B_2& B_1 B_0 for B. Using the fundamental of Vedic multiplication, taking two bit at a time and using 2 bit multiplier block, we can have the following structure for multiplication as shown in Fig. 6.

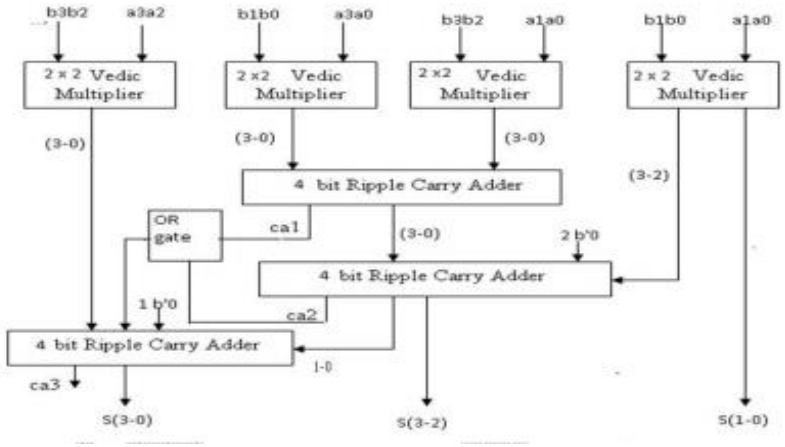

Figure 6 Block Diagram of 4x4 bit Vedic Multiplier

Each block as shown above is 2x2 bit Vedic multiplier. First 2x2 bit multiplier inputs are A_1A_0 and B_1B_0. The last block is 2x2 bit multiplier with inputs A_3 A_2 and B_3 B_2. The middle one shows two 2x2 bit multiplier with inputs A_3 A_2& B_1 B_0 and A_1 A_0& B_3 B_2. So the final result of multiplication, which is of 8 bit, S_7 S_6 S_5 S_4 S_3 S_2 S_1 S_0 and three 4-bit Ripple-Carry (RC) Adders are required. The proposed Vedic multiplier can be used to reduce delay. The arrangements of RC Adders shown in Fig. 6, helps us to reduce delay.

Each block as shown above is 2x2 bit Vedic multiplier. First 2x2 bit multiplier inputs are A_1A_0 and B_1B_0. The last block is 2x2 bit multiplier with inputs A_3 A_2 and B_3 B_2. The middle one shows two 2x2 bit multiplier with inputs A_3 A_2& B_1 B_0 and A_1 A_0& B_3 B_2. So the final result of multiplication, which is of 8 bit, S_7 S_6 S_5 S_4 S_3 S_2 S_1 S_0 and three 4-bit Ripple-Carry (RC) Adders are required. The proposed Vedic multiplier can be used to reduce delay. On the other hand, we proposed a new architecture, which is efficient in terms of speed. The arrangements of RC Adders shown in Fig. 6, helps us to reduce delay.

5.3. 8x8 bit Module for Vedic Multiplier

The 8x8 bit Vedic multiplier module as shown in the block diagram in Fig. 7 can be easily designed by using four 4x4 bit Vedic multiplier. Let's analyze 8x8 multiplications, say A= A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0 and B= B_7 B_6 B_5 B_4 B_3 B_2 B_1B_0. The output line for the multiplication result will be of 16 bits as – S_{15} S_{14} S_{13} S_{12} S_{11} S_{10} S_9 S_8 S_7 S_6 S_5 S_4 S_3 S_2 S_1 S_0. Let's divide A and B into two parts, say the 8 bit multiplicand A can be decomposed into pair of 4 bits AH-AL. Similarly multiplicand B can be decomposed into BH-BL.

Using the fundamental of Vedic multiplication, taking four bits at a time and using 4 bit multiplier block as discussed we can perform the multiplication. The outputs of 4x4 bit multipliers are added accordingly to obtain the final product. Here total three 8 bit Ripple-Carry Adders are required as shown in Fig. 7.

5.4. 16x16 bit Module for Vedic Multiplier

The 16x16 bit Vedic multiplier module as shown in the block diagram in Fig. 8 can be easily designed by using four 8x8 bit Vedic multiplier. The 16x16 multiplications, say A= A_{15} A_{14} A_{13} …A_3 A_2 A_1 A_0 and B= B_{15} B_{14} B_{13}...B_3 B_2 B_1 B_0. The multiplication result will be of 32 bits as – S_{31} S_{30} S_{29} S_{28} …S_3 S_2 S_1 S_0. Let's divide A and B into two parts, say the 16 bit multiplicand A can be decomposed into pair of 8 bits AH-AL. Similarly multiplicand B can be decomposed into BH-BL.

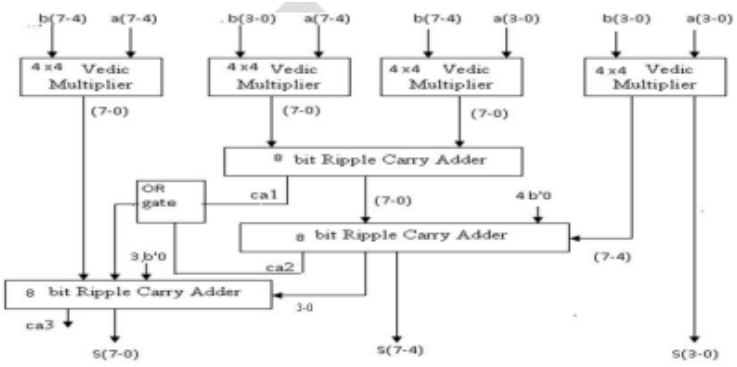

Figure 7 Block Diagram of 8x8 bit Vedic Multiplier

Using Vedic multiplication, taking four bits at a time and using 8 bit multiplier block. The outputs of 8x8 bit multipliers are added accordingly to obtain the final product. Here total three 16 bit Ripple-Carry Adders are required as shown in Fig. 8.

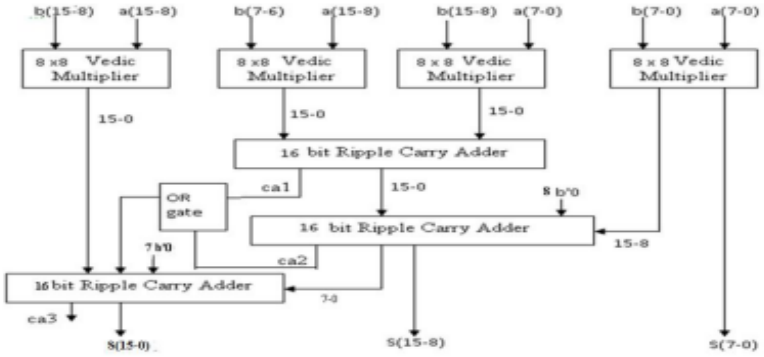

Figure 8 Block Diagram of 16x16 bit Vedic Multiplier

5.5. 32x32 bit Module for Vedic Multiplier

The 32x32 bit Vedic multiplier module as shown in the block diagram in Fig. 9, it can be designed by using four 16x16 bit Vedic multiplier modules as discussed. Let's analyze 32x32 multiplications, say A=A_{31} A_{30} A_{29} A_{28}…A_3 A_2 A_1 A_0 and B= B_{31} B_{30} B_{29} B_{28}…B_3 B_2 B_1B_0. The output line for the multiplication result will be of 64 bits as – S_{63} S_{62} S_{61} S_{12}…S_4 S_3 S_2 S_1 S_0. Let's divide A and B into two parts, say the 32 bit multiplicand A can be decomposed into pair of 16 bits AHAL. Similarly multiplicand B can be decomposed into 16 bits BH-BL.

The outputs of 16x16 bit multipliers are added accordingly to obtain the final product. Here total three 32 bit Ripple-Carry Adders are required as shown in Fig. 9.

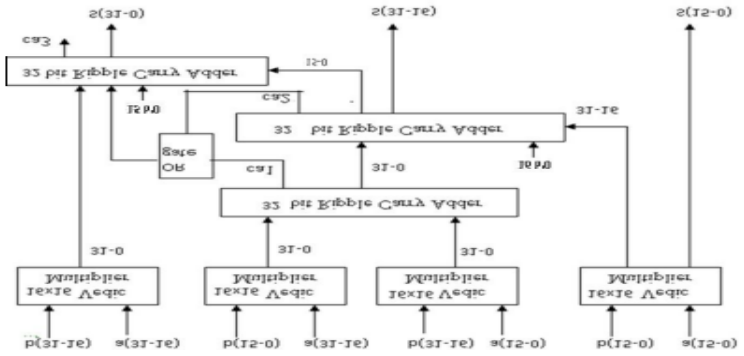

Figure 9 Block Diagram of 32x32 bit Vedic Multiplier

5.6. 64x64 bit Module for Vedic Multiplier

The 64x64 bit Vedic multiplier module as shown in the block diagram in Fig. 10 by using four 32x32 bit Vedic multiplier. The 64x64 multiplications, say A=A_{63} A_{62} $A_{61}A_{60}$…A_3 A_2 A_1 A_0 and B=B_{63} B_{62} B_{61} B_{60}…B_3 B_2 B_1 B_0. The output line for the multiplication result will be of 128 bits as – S_{128} S_{127} $S_{126}S_{125}$…S_3 S_2 S_1 S_0. Let's divide A and B into two parts, say the 64 bit multiplicand A can be decomposed into pair of 32 bits AH-AL. The multiplicand B can be decomposed into 32 bits BH-BL. The outputs of 32x32 bit multipliers are added accordingly to obtain the final product. Here total three 64 bit Ripple-Carry Adders are required as shown in Fig. 10. [5]

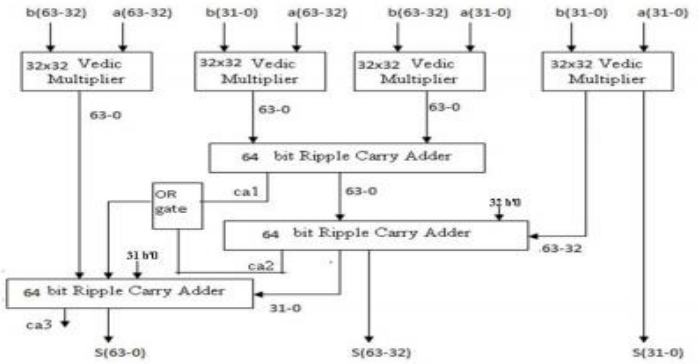

Figure 10 Block Diagram of 64x64 bit Vedic Multiplier

6. LITRATURE REVIEW

In this section we will discuss the efforts taken by the researchers on submission of Vedic mathematics.

The judgment between conventional and Vedic mathematics realized in VLSI for RSA algorithm, ALU, curve encryption etc. with respect to effectiveness analysis and complexity has been offered in [6]. It displays that Vedic mathematical method is fast and modest. 'Urdhva Tiryagbhyam Sutra' and 'Nikhilam Sutra' multiplication methods are planned in [7].

A low power Multiplier is offered in [8]. The realized multiplier is based on the ancient Vedic Multiplication Method. At this point the 'Urdhva tiryakbhyam sutra' and 'Nikhilam sutras' are used for multiplication. The multiplier founded on this technique is equated with the modern multiplier to highpoint the power and speed rewards in the Vedic Multipliers. To check the Vedic multiplier BIST (Built inSelf-Test) is realized and it is establish Fault free. The outcomes are compared with the Booth's Multiplier in terms of constraints like power and time delay. The multiplier is realized using VHDL and Spartan 2G FPGA. The simulation outcomes are offered based on power and time delay.

Reference [9] offerings a squarer based high enactment multiplier for which Vedic multipliers and two flexible constant coefficient multipliers are castoff. Outcomes are stored in ROM which rises power consumption.

The arrangement proposed in [10] attains increased speed and compact area as equated to array multipliers. Conferring to authors its only drawback is increment in dynamic power.

Additional work can be carried out to minimalize delay and to advance the speed. The efficacy comparison between Karatsuba multiplier by polynomial multiplication with multiplier realizing 'Nikhilam Sutras' have been offered in [11] which states that Karatsuba multiplier displays speed improvement as equated to Vedic multiplier. Animproved 'Urdhva

tiryakbhyam' algoriothm has been realized on new multiplier for low power, high speed uses. The new algorithm creates concurrent carry for succeeding stage which is built on generation and addition of coexisting partial sums formed within matrix architecture [12].

The core calculating unit of any DSP processor is ALU which is built on multiplication operation is time overwhelming and complex. The speed can be amplified and complexity can be concentrated by use of Vedic mathematics in application of multiplication operation. The actions like increment, decrement are built on 'Ekadhikina Purvena' and 'Ekanyunena Purvena' Sutras [13]

7. CONCLUSION AND FUTUR SCOPE

Vedic Multiplier is understood to be efficient in speed, power and area in digital plans with respect to further multipliers. Seeing all the designs of it debated above, we can accomplish that the Vedic multiplier with Urdhva Tiryakbhyam sutra is observed as auspicious technique in terms of speed and area. The work can be further extended with the usage of such multiplier in arithmetic logical unit, multiply accumulator unit projects and comparing the outcomes with existing designs for the same to boost-up the speed and reduce power dissipation and power consumption, so that efficiency of the unit should get increased.

REFERENCES

[1] Muhammad H. Rais, Mohammed H. Al Mijalli, Mohammad Nisar. 2012. FPGA Realization of Braun's Multiplier. Proceedings of International Multi-Conference of Engineers and Computer Scientists Vol. I, IMECS 2012, Hong Kong.

[2] P. Jagadeesh, Mr. S. Ravi and Dr. Kittur Harish Mallikarjun, "Design of High Performance 64 bit MAC Unit" IEEE International conference on Circuits, power and Computing Technologies [ICCPCT-2013]

[3] C Ranjit Kumar, G Rahul Ram, N Chandu Reddy, "Design of Square and Multiply and Accumulate(MAC) Unit by using Vedic Multiplication Techniques'' International Journal of Scientific & Engineering Research, Volume 4, Issue 12, December-2013

[4] Parth S. Patel, Kyati K. Parasania, "Design of High Speed MAC (Multiply and Accumulate) Unit Based on Urdhva Tiryakbhyam Sutra" International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), Volume 4 Issue 6, June 2015

[5] Sumit C. Katkar, Pragati Kene, Shubhangini Ugale, "Design of Efficient 64 bit MAC Unit using Vedic Multiplier for DSP Applications-A Review" International Journal of Advanced Information and Communication Technology (IJAICT), Volume 1, Issue 9, January 2015

[6] S. M. Khairnar, Sheetal Kapade, Naresh Ghorpade 2012 "Vedic mathematics - The cosmic software for implementation of fast algorithms", IJCSA-2012

[7] Manorajan Pradhan, Rutuparna Panda, Sushant Kumar Sahu 2011, "Speed comparison of 16 X 16 vedic multipliers", International journal of computer applications, Vol. 21, No.6, May 2011

[8] Aniruddha Kanhe, Shishir Kumar Das, Ankit Kumar Singh, 2012 "Design and implementation of low power multiplier using Vedic multiplication technique", International Journal of computer science and communication techniques, Vol. 3 No.1 Jan-June 2012, 131-132

[9] L. Sriraman, T. N. Prabakar 2012, "Design and Implementation of Two Variable KCM using Multiplier using KCM and Vedic Mathematics ", in 1st International Conference on Recent Advancements in Information Technology, 2012.

[10] L. Sriraman, T. N. Prabakar 2012, "FPGA implementation of high performance multiplier using squarer", International Journal of Advanced Computer Engineering & Architecture Vol.2, No.2, June-December 2012

[11] Sudhanshu Mishra, Manoranjan Pradhan 2012, "Synthesis comparison of Karatsuba multiplier using polynomial multiplication, Vedic multiplier and classical multiplier", International journal of computer applications (0975- 8887) Vol. 41 No. 9, March 2012.

[12] Prashant Nair, Darshan Paranji, S. S. Rathod, "VLSI implementation of matrix diagonal method of binary multiplication", Proceedings of SPIT-IEEE Colloquium and International Conference, Mumbai, India, Vol. 2, 55

[13] V. Vamshi Krishna, S. Naveen Kumar 2012, "High Speed, Power and Area efficient Algorithms for ALU using Vedic Mathematics" International Journal of Scientific and Research Publications, Volume 2, Issue 7, July 2012.