ISSN 2348 – 9928
IJAICT Volume 4, Issue 11, November 2017 Doi:01.0401/ijaict.2015.09.07 Published on 05 (11) 2017

DESIGN OF EFFICIENT 64 BIT MAC UNIT USING VEDIC MULTIPLIER FOR DSP APPLICATION – A REVIEW

Mr. Sumit C. Katkar
PG Scholar,

Prof. Pragati Kene
Assistant Professor,

Prof. Shubhangini Ugale
Assistant Professor,

Electronics and Communication Engineering,
G. H. Raisoni Academy of Engineering and Technology,
Nagpur, Maharashtra, India

Abstract— ***Multiplier Accumulator Unit (MAC) is a part of Digital Signal Processors. The speed of MAC depends on the speed of multiplier. The proposed MAC unit reduces the area by reducing the number of multiplication and addition in the multiplier unit. Increase in the speed of operation is achieved by the hierarchical nature of the Vedic multiplier unit. So by using an efficient Vedic multiplier which excels in terms of speed, power and area, the performance of MAC can be increased. For this fast method of multiplication based on ancient Indian Vedic mathematics is used. Among various method of multiplication in Vedic mathematics, Urdhva Tiryagbhyam is used and the multiplication is for 64 X 64 bits. Urdhva Tiryagbhyam is a general multiplication formula applicable to all cases of multiplication. It enables parallel generation of intermediate products, eliminates unwanted multiplication steps with zeros.***

Keywords— ***MAC, Vedic Multiplier, VHDL, Ripple Carry (RC) Adder.***

I. INTRODUCTION

Digital multipliers are the core components of all Digital signal processors. The speed of DSP is largely determined by the speed of its multipliers. Multiply Accumulate (MAC) operation is a commonly used operation in various Digital Signal Processing Applications. Use of a Digital Signal processor can significantly increase the performance of a MAC. Normally a multiply accumulate unit consists of a multiplier along with an accumulator which stores previous multiplication products. Since system performance widely depends on time needed to execute the instruction and multiplication being the most time consuming, any improvement to multiplication will inherently improve the system performance.

Multiplication can be designed using several algorithms such as array, Booth, carry save, Modified Booth algorithm and Wallace tree.In array multiplier multiplication of two numbers can be obtained with one micro operation. It is a fast method of multiplication since the only delay is time for the signals to propagate through the gates. But it requires larger number of gates and so it is less economical.

A new algorithm is developed that uses Vedic mathematics. The conventional mathematical algorithms can be simplified and even optimized by the use of Vedic mathematics. The Vedic algorithm is applicable to arithmetic, trigonometric, plain and spherical geometry, calculus. The whole of Vedic mathematics is based on 16 sutras. Here we use Urdhva Tiryagbhyam of Vedic mathematics. This sutra was traditionally used in ancient for the multiplication of two decimal numbers in relatively less time [2]. The architecture of urdhva tiryagbhyam is explained that any NxN multiplication can be efficiently designed by breaking it into smaller numbers of size (N/2=n) and these smaller numbers can again broken into smaller numbers (n/2) till we reach multiplicand size of (2 x2). Thus simplifying the whole multiplication process. This work present a systematic design methodology for fast and area efficient digital multiplier based on Vedic mathematics and then a MAC unit has been made which uses this multiplier [5].

1.1 MAC Operation

The Multiplier-Accumulator (MAC) operation is the key operation not only in DSP applications but also in multimedia information processing and various other applications. As mentioned above, MAC unit consist of multiplier, adder and accumulator. The MAC inputs are obtained from the memory location and given to the multiplier block. This will be useful in 64 bit digital signal processor. The input which is being fed from the memory location is 64 bit. When the input is given to the multiplier it starts computing value for the given 64 bit input and hence the output will be 128 bits. The multiplier output is given as the input to Ripple carry adder which performs addition. The function of the MAC unit is given by the following equation [1]:

$$F= sum(PiQi)……….(1)$$

The output of Ripple carry adder is 129 bit i.e. one bit is for the carry (128bits+ 1 bit). Then, the output is given to the accumulator register. The output of the accumulator register is taken out or fed back as one of the input to the Ripple carry adder. The figure 1 shows the basic architecture of MAC unit.

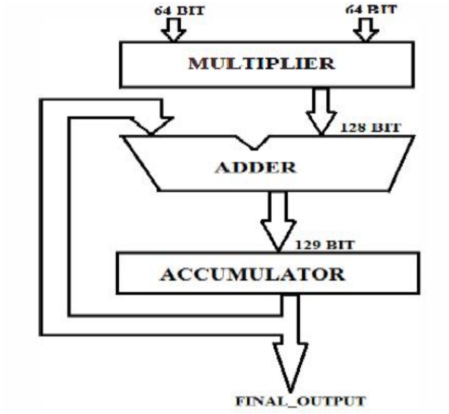


Fig 1. Basic architecture of MAC unit

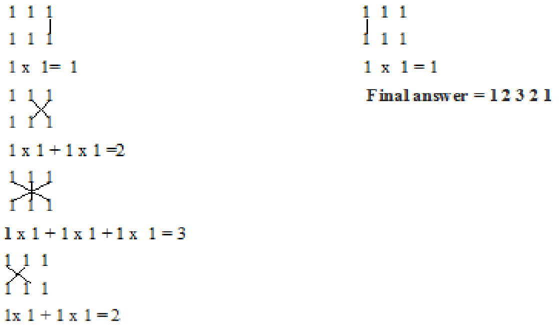
II. VEDIC MATHEMATICS PRINCIPLE

Vedic mathematics is the name given to the ancient system of mathematics, which was discovered between 1911 and 1918 by Sri Bharati Krishna Tirthaji. The word “Vedic” is derived from the word “Veda” which means the store house of all knowledge . The Vedic mathematics is based on 16 sutras which deal with various branches of mathematics. These sutras have been traditionally used for the multiplication of two numbers in the decimal number system. The possible multiplier architecture of Vedic mathematics to be designed on DSP applications is Urdhva Tiryagbhyam. Traditional Indian mathematicians used this sutra to do multiplication of two decimal numbers in less time. It multiplies the number in the vertical and crosswise fashion. It is applicable to all cases of multiplication [3].

2.1 Urdhva Tiryagbhyam Sutra

Urdhva – Triyakbhyam is the general formula applicable to all cases of multiplication and also in the division of a large number by another large number. It means vertically and crosswise. “Urdhva” and “Tiryagbhyam” words are derived from Sanskrit literature. It literally means “vertical and crosswise”.

The method of urdhva tiryagbhyam is explained that any NxN multiplication can be efficiently designed by breaking it into smaller numbers of size (N/2=n) and these smaller numbers can again broken into smaller numbers (n/2) till we reach multiplicand size of (2 x2). Thus simplifying the whole multiplication process . Ex.1. the product of 111 and 111 using Urdhva Triyakbhyam (vertically and crosswise) is given below. Methodology of Parallel Calculation



III. MULTIPLIER ARCHITECTURE

Here, “Urdhva-Tiryagbhyam” (*Vertically and Crosswise*) sutra is used to propose such architecture 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.

3.1 Vedic Multiplier for 2x2 bit Module

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. 2. Firstly, 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.

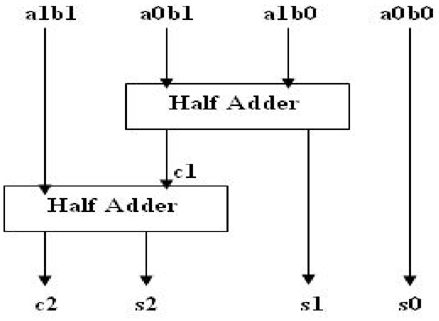


Fig 2 : Block Diagram of 2x2 bit Vedic Multiplier

3.2 Vedic Multiplier for 4x4 bit Module

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= A3 A2 A1 A0 and B= B3 B2 B1 B0. The output line for the multiplication result is – S7 S6 S5 S4 S3 S2 S1 S0 Let's divide A and B into two parts, say A3 A2 & A1 A0 for A and B3 B2 & B1 B0 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. 3.

Each block as shown above is 2x2 bit Vedic multiplier. First 2x2 bit multiplier inputs are A1A0 and B1B0. The last block is 2x2 bit multiplier with inputs A3 A2 and B3 B2. The middle one shows two 2x2 bit multiplier with inputs A3 A2 & B1 B0 and A1 A0 & B3 B2. So the final result of multiplication, which is of 8 bit, S7 S6 S5 S4 S3 S2 S1 S0 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. 3, helps us to reduce delay.

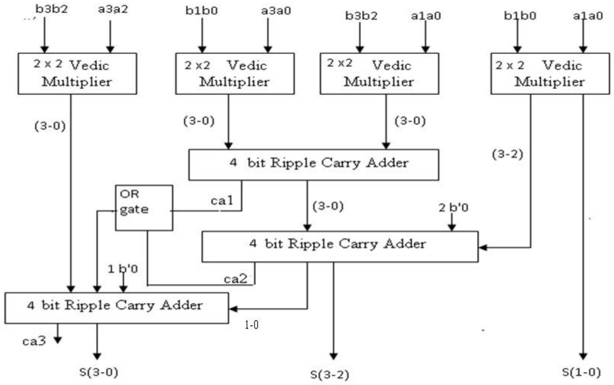


Fig 3 : Block Diagram of 4x4 bit Vedic Multiplier

3.3 Vedic Multiplier for 8x8 bit Module

The 8x8 bit Vedic multiplier module as shown in the block diagram in Fig. 4 can be easily designed by using four 4x4 bit Vedic multiplier. Let's analyze 8x8 multiplications, say A= A7 A6 A5 A4 A3 A2 A1 A0 and B= B7 B6 B5 B4 B3 B2 B1 B0. The output line for the multiplication result will be of 16 bits as – S15 S14 S13 S12 S11 S10 S9 S8 S7 S6 S5 S4 S3 S2 S1 S0. 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. The 16 bit product can be written as:

P = A x B = (AH-AL) x (BH-BL)
 = AH x BH + (AH x BL + AL x BH) + AL x 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. 4.

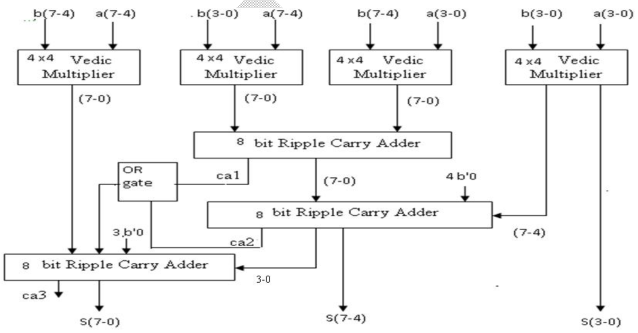


Fig 4 : Block Diagram of 8x8 bit Vedic Multiplier

3.4 Vedic Multiplier for 16x16 bit Module

The 16x16 bit Vedic multiplier module as shown in the block diagram in Fig. 5 can be easily designed by using four 8x8 bit Vedic multiplier. The 16x16 multiplications, say A= A15 A14 A13 ………..A3 A2 A1 A0 and B= B15 B14 B13.......... B3 B2 B1 B0. The output line for the multiplication result will be of 32 bits as – S31 S30 S29 S28 …………. S3 S2 S1 S0. 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.

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. 5.

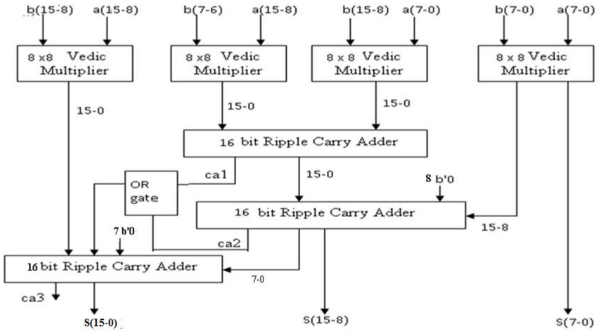


Fig 5: Bblock Diagram of 16x16 bit Vedic Multiplier

3.5 Vedic Multiplier for 32x32 bit Module

The 32x32 bit Vedic multiplier module as shown in the block diagram in Fig. 6, it can be designed by using four 16x16 bit Vedic multiplier modules as discussed. Let's analyze 32x32 multiplications, say A= A31 A30 A29 A28………………….. A3 A2 A1 A0 and B= B31 B30 B29 B28…………. B3 B2 B1 B0. The output line for the multiplication result will be of 64 bits as – S63 S62 S61 S12…………………..S4 S3 S2 S1 S0. Let's divide A and B into two parts, say the 32 bit multiplicand A can be decomposed into pair of 16 bits AH-AL. 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. 6.

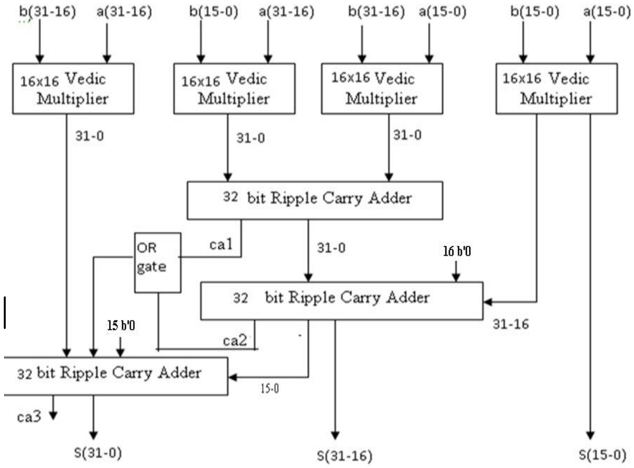


Fig 6 : block Diagram of 32x32 bit Vedic Multiplier

3.6 Vedic Multiplier for 64x64 bit Module

The 64x64 bit Vedic multiplier module as shown in the block diagram in Fig. 7 by using four 32x32 bit Vedic multiplier. The 64x64 multiplications, say A= A63 A62 A61 A60…………. A3 A2 A1 A0 and B= B63 B62 B61 B60………. B3 B2 B1 B0. The output line for the multiplication result will be of 128 bits as – S128 S127 S126 S125………………… S3 S2 S1 S0. 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. 7.

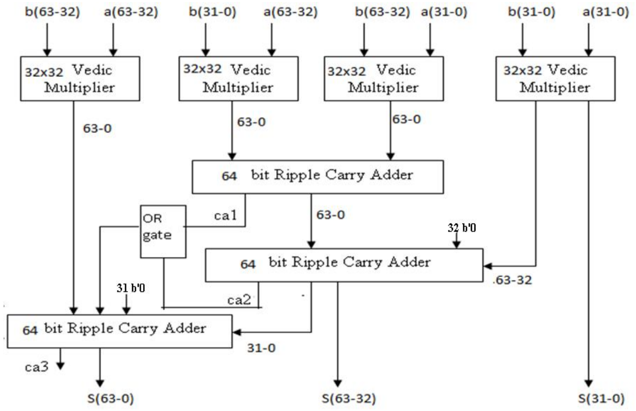


Fig 7 : block Diagram of 64x64 bit Vedic Multiplier

IV. LITERATURE REVIEW

In 2013 P. Jagadeesh, Mr.S.Ravi and Dr. Kittur Harish Mallikarjun, "*Design of High Performance 64 bit MAC Unit*" in this paper designed of high performance 64 bit Multiplier-and Accumulator (MAC). The total MAC unit operates at a frequency of 217 MHz. The total power dissipated by 64 bit MAC unit is 177.732 mW. The total area occupied by it is 542177 11m2. Since the delay of 64 bit is less, this design can be used in the system which requires high performance in processors involving large number of bits of the operation. The MAC unit is designed using Verilog-HDL and synthesized in Cadence 180nm RTL Complier.

In 2013 Shishir Kumar Das, Aniruddha Kanhe, R.H. Talwekar, *"Design and Implementation of High performance MAC Unit" in this paper implemented* 32 bit IEEE 754 Floating point multiplier based on Vedic Multiplication technique. These multipliers are implemented using VHDL. In order to get the power and delay report the multipliers are synthesized using Xilinx ISE tool and Spartan 2E FPGA is used. They gives simulation result of multipliers with Vedic Multiplier on basis of time delay and power.

In 2013 Sreelekshmi M. S., Farsana F. J., Jithin Krishnan3, Rajaram S, Aneesh R, "*Implementation of MAC by using Modified Vedic Multiplier"*in this paper they observed that for 16x16 Vedic multiplier the delay obtained is 21.4ns. Model sim is used for simulation and synthesis of the Vedic multiplier is carried out using Xilinx ISE 10.1. The delay of 16x16 Vedic multiplier is 21.4ns with nearly 8% device utilization.(number of slices: 508 out of 704) and number of 4 input LUTs: 98 out of 1408(6%). The number of bonded IOBs: 28 out of 108(25%).

V. SIMULATION RESULT

The 64x64 multiplier is designed using model sim for simulation and synthesis of the vedic multiplier is carried out using Xilinx ISE 13.1 on Spartan 7A. Hence the proposed multiplier architecture is found most efficient in terms of speed.

Table 1. SYNTHESIS RESULT

Name of multiplier	Number of Slice LUTs	Number of fully used FF pairs	Number of bonded IOBs
64 bit	178800 (available) used 9900.	9900 (available) used 0.	600 (available) used 256.
32 bit	178800 (available) used 2412.	2412 (available) used 0.	600 (available) used 128.
16 bit	178800 (available) used 574.	574 (available) used 0.	600 (available) used 64.
8bit	178800 (available) used 129.	129 (available) used 0.	600 (available) used 32.
4bit	178800 (available) used 25.	25 (available) used 0.	600 (available) used 16.
2bit	178800 (available) used 4.	4 (available) used 0.	600 (available) used 8.

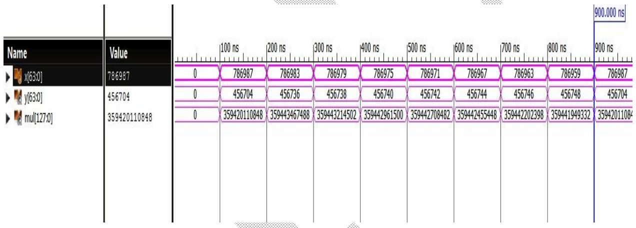


Fig 8 : Simulation Result

VI. CONCLUSION AND FUTURE WORK

Urdhva Tiryagbhyam Sutra is highly efficient algorithm for multiplication. The design of 64x64 bit Vedic multiplier has been realized on Spartan 7A. The computation delay obtained for 64x64 bit Vedic multiplier having a total delay of 42.98ns containing logic delay 3.405ns and route delay 39.578ns. This shows improvement in performance.

References

[1] 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]

[2] V.K.Karthik, Y.Govardhan, V.Karunakara Reddy, K.Praveena, " Design of Multiply and Accumulate Unit using Vedic Multiplication Techniques" International Journal of Scientific & Engineering Research, Volume 4, Issue 6, June-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] Fabrizio Lamberti and Nikos Andrikos, "Reducing the Computation Time in (Short Bit-Width) Two's Complement Multipliers '', IEEE transactions on computers, Vol. 60, NO. 2, February 2011

[5] Young-Ho Seo and Dong-Wook Kim,New "VLSI Architecture of Parallel Multiplier-Accumulator Based on Radix-2 Modified Booth Algorithm" IEEE Transactions on very largescale integration (vlsi) systems, vol. 18, no. 2,february 2010.