Design & Simulation of 64-Bit Multiplier-Accumulator Unit using VHDL

Miss. Pratiksha Awate[1]Prof. P. R. Indurkar[2]
[1]M.Tech Student [2]Associate Professor
[1,2]Department of Electronics & Communication Engineering
[1,2]BDCOE, Sewagram, Maharashtra, India

Abstract*—This paper targets the Design, Synthesis and Simulation of 64-bit multiplier-accumulator unit using VHDL. Further we have introduced CSA (Carry save Adder) into the proposed design to increase the speed and performance of the proposed design. CSA is one way of improving the overall processing performance of a multiplier. Here, CSA with modified techniques are used to increase the speed of the Vedic Multiplier; also it can perform fast operations as compared to normal multiplier. Design, synthesis and simulation of 64-bit multiplier-accumulator unit using VHDL have been done using XILINX ISE 14.5 software. Coding of the proposed design has been done in VHDL (Very high Speed Integrated Circuit Hardware Description Language). After taking synthesis of the design it is found that 64-bit multiplier-accumulator unit operation can be done in 0.562ns time and 176.835 MHz frequency with only 143mW power consumption.*

Keywords—MAC Unit, Vedic Multiplier, XILINX ISE, VHDL

I. INTRODUCTION

Multipliers are most commonly used in various electronic applications e.g. Digital signal processing in which multipliers are used to perform various algorithms like FIR, IIR etc. Earlier, the major challenge for VLSI designer was to reduce area of chip by using efficient optimization techniques to satisfy MOORE'S law. Then the next phase is to increase the speed of operation to achieve fast calculations like, in today's microprocessors millions of instructions are performed per second. Speed of operation is one of the major constraints in designing DSP processors and today's general-purpose processors. However area and speed are two conflicting constraints. So improving speed results always in larger areas. Now, as most of today's commercial electronic products are portable like Mobile, Laptops etc. that require more battery back-up. Therefore, lot of research is going on to reduce power consumption. So, in this paper it is tried to find out the best solution to achieve low power consumption, less area required and high speed for multiplier operation

In most of the DSP applications, multiplier is the main of the system. The speed of that system is mainly depends upon the multiplier. If the multiplier is efficient for performing fast operations then the overall speed of the design automatically increases. So, there is need of high speed multiplier in every system which consists of multiplier. In some DSP applications, multiplications are carried out such as in FFT processor. Hence, if we replace the multipliers used by that system by most efficient multiplier based on Vedic mathematics then the speed of operation of that system will increase and the system will become more efficient. Hence, design of FFT system using Vedic mathematics is a necessary choice.

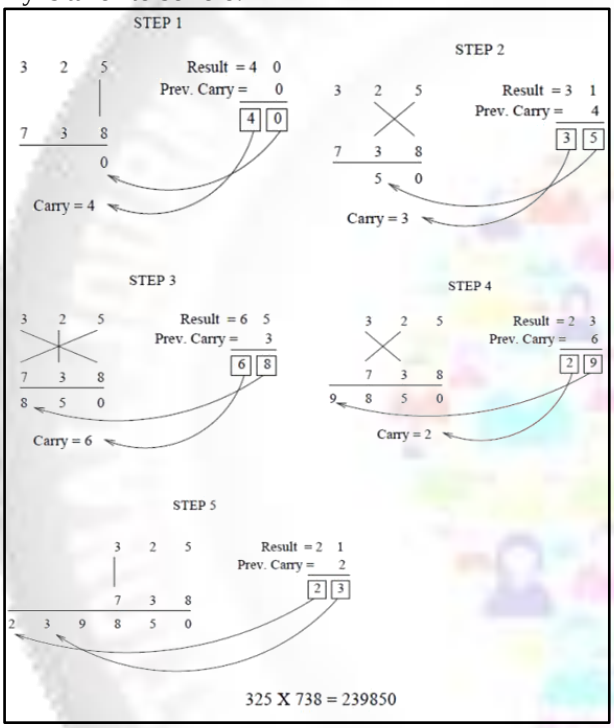
Multipliers are key components of many high performance systems such as FIR filters, Microprocessors, digital signal processors, etc. Furthermore, it is generally the most area consuming. Hence, optimizing the speed and area of the multiplier is a major design issue. However, area and speed are usually conflicting constraints so that improving speed results mostly in larger areas. As a result, a whole spectrum of multipliers with different area-speed constraints has been designed with fully parallel end of the spectrum and fully serial multipliers at the other end. The speed of multiplication operation is increased using several schemes such as Wallace-tree, Booth and CSA multipliers. The array multiplier is the simplest architecture and is most suitable for VLSI implementation because of its high degree of regularity. A system's performance is generally determined by the performance of the multiplier because the multiplier is generally the slowest clement in the system. The multiplier circuit is a core component of most of the present day digital signal processors. Therefore, the demand for multiplier-performance improvement is increasing. Multipliers are a major source of power dissipation. Reducing the power dissipation of multipliers is key to satisfying the overall power budget of various digital circuits and systems. For high speed FFT processor, multiplier has a major role, as it is one of the processing elements of FFT. To meet better performance of FFT, there should be high speed multiplier.

II. VEDIC MULTIPLIER

Vedic mathematics is mainly based on 16 Sutras (or aphorisms) dealing with various branches of mathematics like arithmetic, algebra, geometry etc. The multiplier is based on an algorithm UrdhvaTiryakbhyam (Vertical & Crosswise) of ancient Indian Vedic Mathematics. UrdhvaTiryakbhyam 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 [2]. Vedic multiplier is faster than array multiplier and Booth multiplier. As the number of bits increases from 8x8 bits to 16x16 bits, the timing delay is greatly reduced for Vedic multiplier as compared to other multipliers. Vedic multiplier has the greatest advantage as compared to other multipliers over gate delays and regularity of structures. Array multiplier, Booth Multiplier is some of the standard approaches used in implementation of binary multiplier which are suitable for VLSI implementation.

Vedic Mathematics is a methodology of arithmetic rules that allow more efficient speed implementation. It also provides some effective algorithms which can be applied to various branches of engineering such as computing. UrdhvaTiryakbhyam Sutra The proposed Vedic multiplier is

based on the "UrdhvaTiryagbhyam" sutra (algorithm). These Sutras have been traditionally used for the multiplication of two numbers in the decimal number system. In this work, we apply the same ideas to the binary number system to make the proposed algorithm compatible with the digital hardware. It is a general multiplication formula applicable to all cases of multiplication. It literally means "Vertically and Crosswise". To illustrate this scheme, let us consider the multiplication of two decimal numbers 325 x 738 by UrdhvaTiryakbhyam method. 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.



An alternative method of multiplication using Urdhvatiryakbhyam Sutra is shown in Fig. 3. The numbers to be multiplied are written on two consecutive sides of the square as shown in the figure. The square is divided into rows and columns where each row/column corresponds to one of the digit of either a multiplier or a multiplicand. Thus, each digit of the multiplier has a small box common to a digit of the multiplicand. These small boxes are partitioned into two halves by the crosswise lines. Each digit of the multiplier is then independently multiplied with every digit of the multiplicand and the two-digit product is written in the common box. All the digits lying on a crosswise dotted line are added to the previous carry. The least significant digit of the obtained number acts as the result digit and the rest as the carry for the next step. Carry for the first step (i.e., the dotted line on the extreme right side) is taken to be zero.

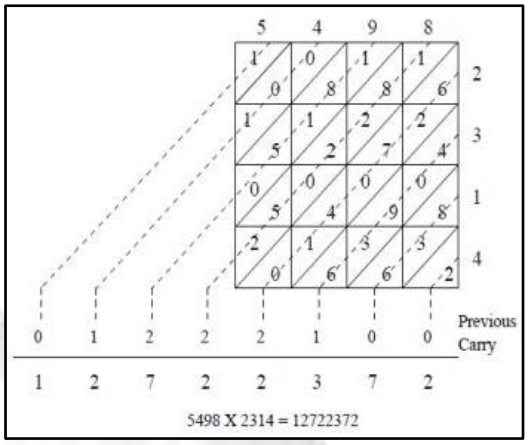


Fig. 1: Alternative Way of Multiplication by UrdhvaTiryakbhyam Sutra

The proposed Vedic multiplier is based on the "UrdhvTiryakbhyam" sutra (algorithm). These Sutras have been traditionally used for the multiplication of two numbers in the decimal number system. In this work, we apply the same ideas to the binary number system to make the proposed algorithm compatible with the digital hardware. It is a general multiplication formula applicable to all cases of multiplication. It literally means "Vertically and crosswise" [7]. 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 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. The Multiplier based on this sutra has the advantage that as the number of bits increases, gate delay and area increases very slowly as compared to other conventional multipliers.

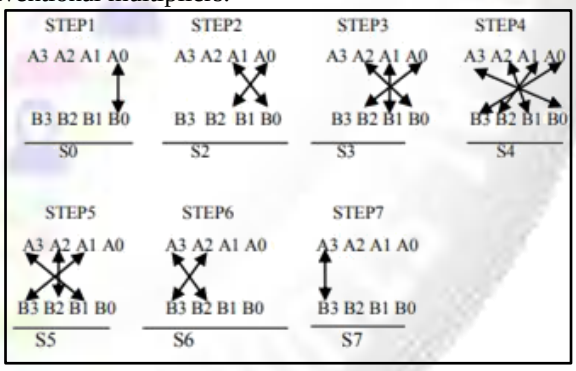


Fig. 2: Vertically & Crosswise Technique

1) Step1: S0 = A0*B0
2) Step2: S1 = A1*B0+A0*B1
3) Step3:S2 = A2*B0+A0*B2+A1*B1
4) Step4:S3 = A3*B0+A0*B3+A2*B1+A1*B2
5) Step5:S4 = A3*B1+A1*B3+A2*B2
6) Step6:S5 = A3*B2+A2*B3
7) Step7:S6 = A3*B3

Finally the output is : S0S1S2S3S4S5S6

III. CARRY SAVE ADDER

The carry-save adder reduces the addition of 3 numbers to the addition of 2 numbers. The propagation delay is 3 gates regardless of the number of bits. The carry-save unit consists of n full adders, each of which computes a single

sum and carries bit based solely on the corresponding bits of the three input numbers. The entire sum can then be computed by shifting the carry sequence left by one place and appending a 0 to the front (most significant bit) of the partial sum sequence and adding this sequence with RCA produces the resulting n + 1-bit value. This process can be continued indefinitely, adding an input for each stage of full adders, without any intermediate carry propagation. These stages can be arranged in a binary tree structure, with cumulative delay logarithmic in the number of inputs to be added, and invariant of the number of bits per input. The main application of carry save algorithm is, well known for multiplier architecture is used for efficient CMOS implementation of much wider variety of algorithms for high speed digital signal processing .CSA applied in the partial product line of array multipliers will speed up the carry propagation in the array. The design schematic of Carry Save Adder is shown in Figure.

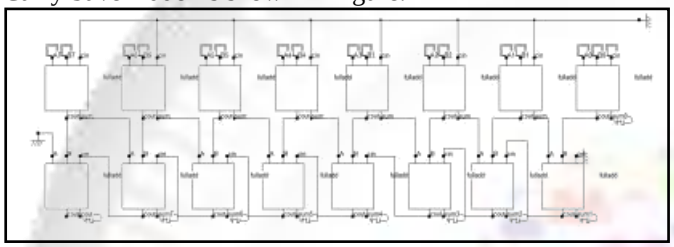


Fig. 3: Schematic of Carry save Adder

There are many cases where it is desired to add more than two numbers together. The straightforward way of adding together m numbers (all n bits wide) is to add the first two, then add that sum to the next, and so on. This requires a total of m − 1 additions, for a total gate delay of O(m lg n) (assuming lookahead carry adders). Instead, a tree of adders can be formed, taking only O(lg m · lg n) gate delays. Using carry save addition, the delay can be reduced further still. The idea is to take 3 numbers that we want to add together, x + y + z, and convert it into 2 numbers c + s such that x + y + z = c + s, and do this in O(1) time. The reason why addition cannot be performed in O(1) time is because the carry information must be propagated. In carry save addition, we refrain from directly passing on the carry information until the very last step. We will first illustrate the general concept with a base 10 example. To add three numbers by hand, we typically align the three operands, and then proceed column by column in the same fashion that we perform addition with two numbers. The three digits in a row are added, and any overflow goes into the next column. Observe that when there is some non-zero carry, we are really adding four digits (the digits of x,y and z, plus the carry).

A. Proposed 64-bit Mac Unit

The block diagram of 64-Bit MAC UNIT Using Vedic Mathematics is shown in the figure below.

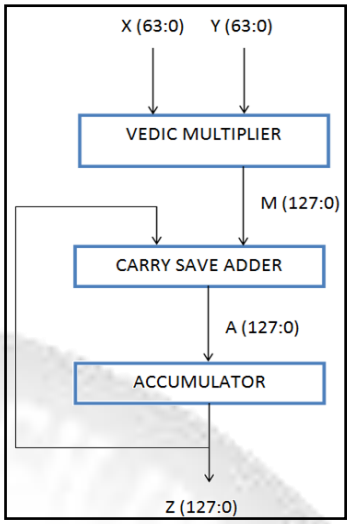


Fig. 4: Block Diagram of Proposed 64-Bit MAC Unit

For designing of 64-Bit MAC UNIT Using Vedic Mathematics; Vedic multiplier, carry save adder and Accumulator are necessary. These three modules are the essential for the design of 64-Bit MAC UNIT Using Vedic Mathematics. For the 64-Bit MAC UNIT Using Vedic Mathematics, the most important module is Vedic multiplier since the performance of 64-bit MAC Unit is depending upon Vedic multiplier unit. In this research work we have designed the 64-Bit MAC UNIT Using Vedic Mathematics.

IV. EXPERIMENTAL RESULTS

A. 64-Bit MAC Unit

1) RTL View

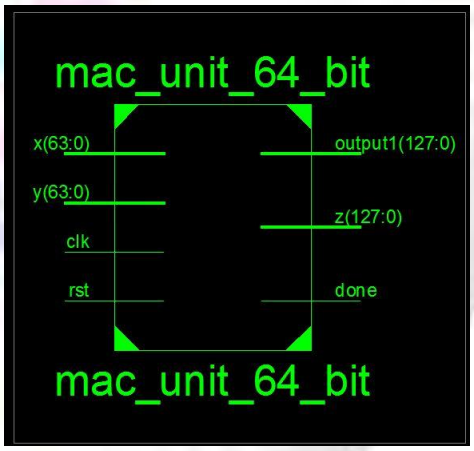


Fig. 5: RTL View of 64-MAC Unit

Figure 5 shows the RTL view of 64-bit MAC Unit using Vedic Multiplier. It shows the register transfer logic view of the 64-bit MAC Unit. It consists of Vedic multiplier, CSA adder and accumulator.

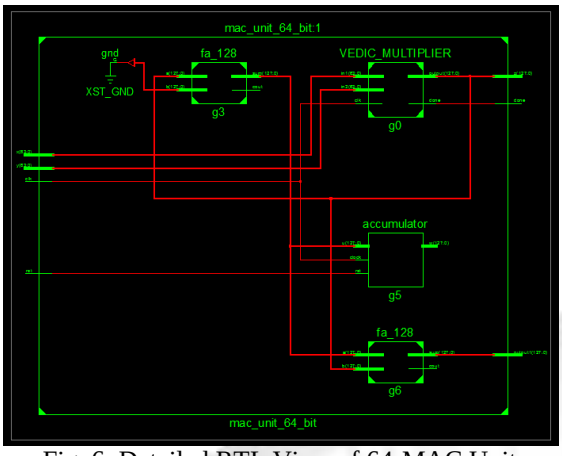


Fig. 6: Detailed RTL View of 64-MAC Unit

Figure 6 shows the Detailed RTL view of 64-bit MAC Unit using Vedic Multiplier. It shows the register transfer logic view of the 64-bit MAC Unit. It consists of Vedic multiplier, CSA adder and accumulator.

B. Simulation Result

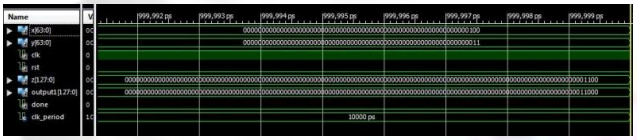


Fig. 7: Simulation Result of 64-MAC Unit

Simulation result for 64-bit MAC Unit using Vedic Multiplier is shown in figure 7. It has inputs x and y for multiplier and multiplicand, clk for clock, rst for reset operation. Outputs are z which represents multiplication of x and y, output1 represents the MAC operation and done for operation has been executed.

C. Result Analysis

The Proposed 64-bit MAC Unit using Vedic Multiplier has found less delay, low power and high frequency. The device utilization of the design shows that the design and be upgrade and integrate on the same device.

Logic Utilization	Used	Available	Utilization
Number of Slice Registers	526	408000	0%
Number of Slice LUTs	2586	204000	1%
Number of fully used LUT-FF pairs	422	2567	15%
Number of bonded IOBs	387	600	64%

Table 1: Device Utilization Table (VIRTEX 7)

Table 1 shows Device Utilization Table for the device, VIRTEX 7 for the design of proposed 64-bit MAC Unit using Vedic Multiplier.

Parameters	Ref [1] (Wallace Multiplier)	Ref [3] (Vedic Multiplier)	Ref [5] (Wallace Multiplier)	Proposed Work (Vedic Multiplier)
Delay (nsec)	4.9	---	160.19	0.562
Power (mW)	177.732	175	1203	143
Frequency (MHz)	---	---	6.24	176.835

Table 2: Comparison Table

Table 2 shows Comparison Table of proposed 64-bit MAC Unit using Vedic Multiplier with different previous papers.

V. CONCLUSION & FUTURE SCOPE

In this research design, synthesis and simulation of 64-Bit MAC unit based on Vedic mathematics Using VHDL has been presented. Synthesis results are shown by RTL view of the design whereas simulation results are shown by the graphical representation of the design. Using this design any 64-bit number multiplication can be possible. After taking synthesis of the design it is found that single MAC operation can be done in 0.562ns time. In future, this MAC unit can be use where MAC operation is required and it can also be upgraded to 128-bit also.

REFERENCES

[1] P.Jagadeesh, Mr.S.Ravi, Dr. Kittur Harish Mallikarjun, Design of High Performance 64 bit MAC Unit, 978-1-4673-4922-2/13/$31.00 ©2013 IEEE, 2013 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2013].

[2] Mohammad Shabber Hasan Khan, Abdussayeed Khan, Design and Implementation of an Efficient 64 bit MAC, Int. Journal of Engineering Research and Applications, ISSN: 2248-9622, Vol. 5, Issue 8, (Part - 2) August 2015, pp.55-57.

[3] Shaik Masthan Sharif , D.Y.V. Prasad, Design of Optimized 64 Bit MAC Unit for DSP Applications, International Journal of Advanced Trends in Computer Science and Engineering, Vol.3 , No.5, Pages : 456-460 (2014).

[4] BhavyaSree Kotte, S Saleem Malik, An Optimized Implementation of CSLA for 32-bit MAC using VHDL, IJSRD - International Journal for Scientific Research & Development Vol. 3, Issue 07, 2015.

[5] K.Kalaiselvi , H. Mangalam, PhD, Area Efficient High Speed and Low Power MAC Unit, International Journal of Computer Applications (0975 – 8887) Volume 67–No.23, April 2013.

[6] Manoranjan Pradhan, Rutuparna Panda, Sushanta Kumar Sahu, MAC Implementation using Vedic Multiplication Algorithm, International Journal of Computer Applications (0975 – 8887) Volume 21–No.7, May 2011.

[7] Rakesh Warrier, C.H. Vun, Wei Zhang, Low-Power Pipelined MAC Architecture using Baugh-Wooley based Multiplier, 978-1-4799-05145-1/14/$31.00 ©2014 IEEE.

[8] Dr. M. Jayaprakash, Dr. V. V. Karthikeyan, M. Peermohamed, LOW POWER AND AREA EFFICIENT MULTIPLIER-ACCUMULATOR UNIT FOR FIR FILTER, Int J AdvEngg Tech/Vol. VII/Issue II/April-June.,2016/405-410.

[9] MarojuSaiKumar,D.Ashok Kumar, Dr.P.Samundiswary, Design and Performance Analysis of Multiply-Accumulate (MAC) Unit, 978-1-4799-2397-7/14/$31.00 ©2014 IEEE.

[10] Naveen Khare,Divyanshu Rao, Ravi Mohan, VLSI Implementation of High Speed MAC Unit Using Karatsuba Multiplication Technique, Journal of Network Communications and Emerging Technologies (JNCET),Volume 6, Issue 1, January (2016).