.

# Design and implement 64 bit MAC(Multiplier and accumulator)

KALIND MEHTA [1], GAURAV PATEL [2]

[1] *M.E. Student, VLSI and Embedded System Design, GTU PG School, Ahmedabad, Gujarat, India.*
[2] *Project Manager ,SAI Education and management services, university of ballaret, ,Austalia.*

## ABSTRACT

*In this thesis I am going to design 64-bit multiply and Accumulator using Vedic multiplier Technique based on Urdhva Tiryagbhyam "-vertically and crosswise sutra. reduce partial product with help of adders during multiplication operation. Vedic multiplier is higher speed compared to other multipliers. It is observe simulation waveform and Compared delay, area and speed. Reduce the delay with help of adders and to improve speed of hardware module.*
*Design of Vedic multiplier with help of adders I want to analysis of adders and comparison of delay between to adders and which adders better for delay in observation of Simulation waveform in Xilinx Software and implementation in FPGA kit. it is comparison of performance parameter like delay, area and speed. Finally design of high speed MAC unit.*

**Keyword**: - *MAC, Vedic Multiplier, Carry save adder*

## 1. INTRODUCTION

Multiply-Accumulate (MAC) unit is extensively used in microprocessors and digital signal processors for data-intensive applications, such as filtering, convolution, FFT transform and inner products. [10] Because they are basically accomplished by repetitive application of multiplication and addition, the speed of the multiplication and addition determines the execution speed and performance of the entire computation. As the multiplier exhibits inherently long delay among the basic operational blocks in digital system, the multiplier determines the critical path.
In order to improve the speed of the MAC unit, there are two major bottlenecks. The first is the partial products reduction network that is used in the multiplication block and the second is the accumulator. Both of these stages require addition of large operands that involve long paths for carry propagation. The main key to the proposed architecture is using the Vedic multiplier to design the MAC unit and compare the performance with the conventional MAC units Using Booth Multiplier, Wallace Multiplier in terms of area, speed and number of resources.
The Vedic multiplier uses "Urdhva Tiryagbhyam" algorithm. The authors in use Vedic multiplier based on the ancient algorithms (sutras) for multiplication.[10]

### 1.1 Multiply and accumulator

Multiplication Accumulation is an important part of real-time digital signal processing (DSP) with applications ranging from digital filtering to image processing. Multiply and accumulate is a very common basic-level operation seen in many DSP designs/algorithms. Two numbers are multiplied together, and added into an accumulator register. the basic MAC unit consists of multiplier, adder and accumulator. In general MAC unit uses the conventional multiplier unit, which consists of multiplication of multiplier and multiplicand based on adding the generated partial products and to compute the final multiplication. This results to adding the partial products.[10] The key to the proposed MAC unit is to enhance the performance of MAC using Vedic Multiplier and to compare the Vedic, Booth and Wallace tree multiplier in terms of computation required to generate the partial products and add the generated partial products to get the final result of the multiplication.[10]

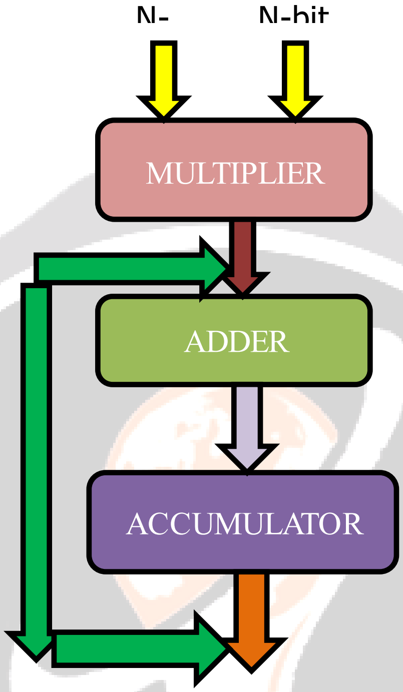


**Fig -1**: Architecture of MAC unit

### 1.2 Vedic Multiplier

The main purpose of Vedic Mathematics is to be able to solve complex calculations by simple techniques. The formula being very short makes them practically simple in implementation. Urdhva-tiryagbhyam (Vertically and crosswise) sutra is general formula applicable to multiplication operation. The Vedic multiplier using Urdhva-Tiryakbhyam sutra of width N×N will generate the 2N-1 cross products of different widths which when combined forms (log2N + 1) partial products. The partial products are obtained by vertical and crosswise operations using the Sutra. Hence the delay is equal to adder delay. Critical path would consist of adders adding the maximum number of bits in cross product. In all cases it will be the cross product in which all bits of multipliers are considered.[10] The strategy applied for developing a 64 x 64-bit Vedic multiplier is to design a 2 x 2- bit Vedic multiplier as a basic building module for the system. In the next stage of development a 4 x 4-bit multiplier is designed using 2 x 2-bit Vedic multiplier. Further in same manner 8 x 8, 16 x 16 and32 x 32- bit Vedic multiplier is designed. For the partial product addition for all stages of development a fast adders is used.Multiplier plays an very important role in today's digital circuits. The multiplier is based on an algorithm Urdhva Tiryagbhyam (Vertical and crosswise ).This sutras shows how to handle multiplication of larger number (N X N bits) by breaking it into smaller sizes.[5] For multiplier, first the basic blocks, that 2x2 multiplier are made and then, 4x4 block , 8x8 block and 16x16 block have been made A 4x4 multiplication is simplified into 4,2x2 multiplication that can be performed in parallel. This reduces the number of stages of logic and thus reduces the delay of the multiplier. This example illustrates a better and parallel implementation style of Urdhva Tiryagbhyam sutra.

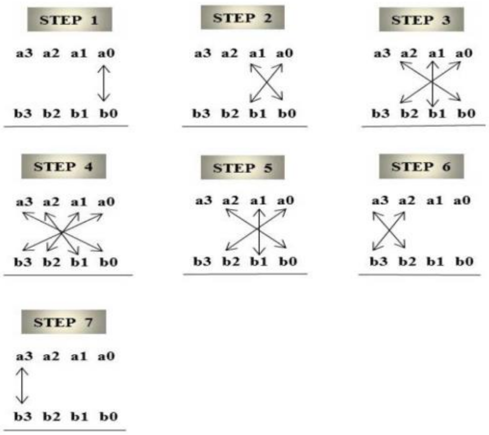
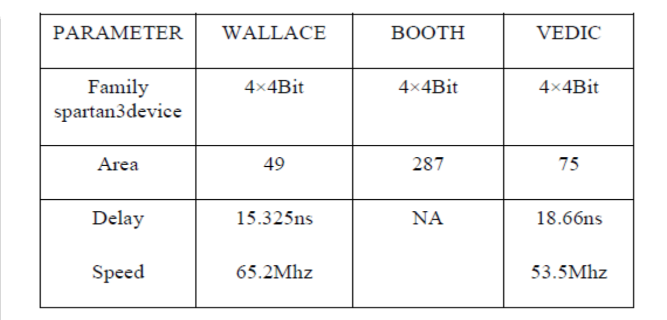


**Fig -2**: Multiplication of 4x4 bit Vedic multiplier

### 1.3 Comparison of different multiplier



| PARAMETER | WALLACE | BOOTH | VEDIC |
|---|---|---|---|
| Family spartan3device | 4×4Bit | 4×4Bit | 4×4Bit |
| Area | 49 | 287 | 75 |
| Delay<br>Speed | 15.325ns<br>65.2Mhz | NA | 18.66ns<br>53.5Mhz |

**Fig -3**: Comparison of different multiplier

## 2. Carry save Adder

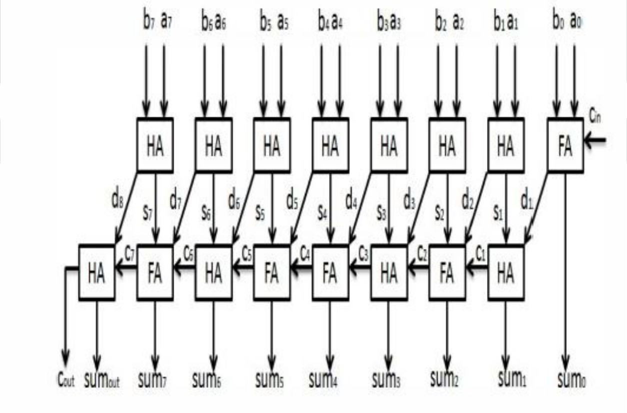


**Fig -4**: 8 bit carry save adder

In this design 128 bit carry save adder [6] is used since the output of the multiplier is 128 bits (2N). The carry save adder minirnize the addition from 3 numbers to 2 numbers. The propagation delay is 3 gates despite of the number of bits. The carry save adder contains n full adders, computing a single sum and carries bit based mainly on the respective bits of the three input numbers. The entire sum can be calculated by shifting the carry sequence left by one place and then appending a 0 to most significant bit of the partial sum sequence. Now the partial sum sequence is added with ripple carry unit resulting in n + 1 bit value. The ripple carry unit refers to the process where the carryout of one stage is fed directly to the carry in of the next stage. This process is continued without adding any

intermediate carry propagation. Since the representation of 128 bit carry save adder is infeasible , hence a typical 8 bit carry save adder is shown in the figure 3[6].Here we are computing the sum of two 128 bit binary numbers, then 128 half adders at the first stage instead of 128 full adder. Therefore , carry save unit comprises of 128 half adders, each of which computes single sum and carry bit based only on the corresponding bits of the two input numbers. If x and y are supposed to be two 128 bit numbers then it produces the partial products and carry as S and C respectively.
Si = xi 1\ yi
Ci = xi & yi
During the addition of two numbers using a half adder, two ripple carry adder is used. This is due the fact that ripple carry adder cannot compute a sum bit without waiting for the previous carry bit to be produced, and hence the delay will be equal to that of n full adders. However a carry-save adder produces all the output values in parallel, resulting in the total computation time less than ripple carry adders. So, Parallel In Parallel Out (PIPO) is used as an accumulator in the final stage.

## 3. Results

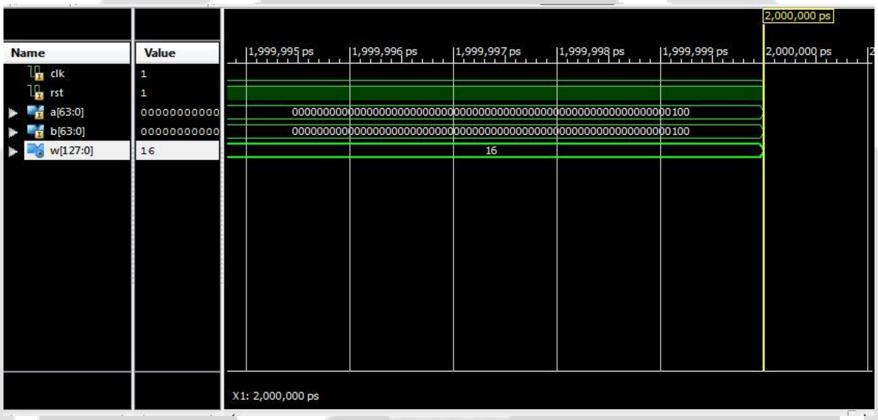


**Fig -5**: 64X64 bit simulation waveform

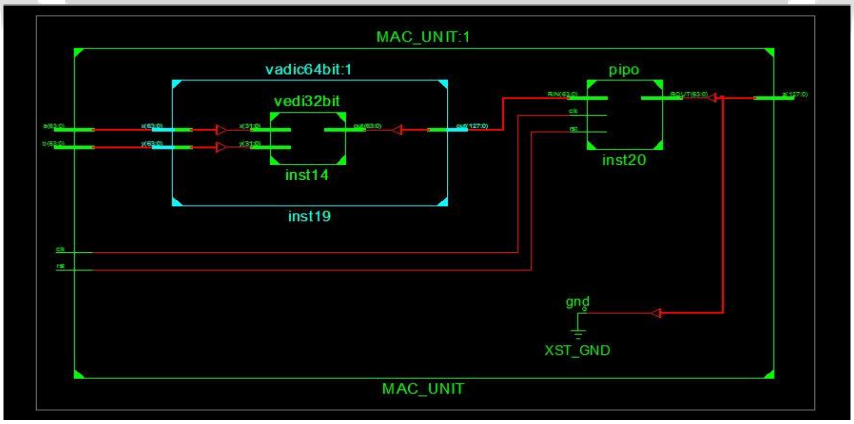
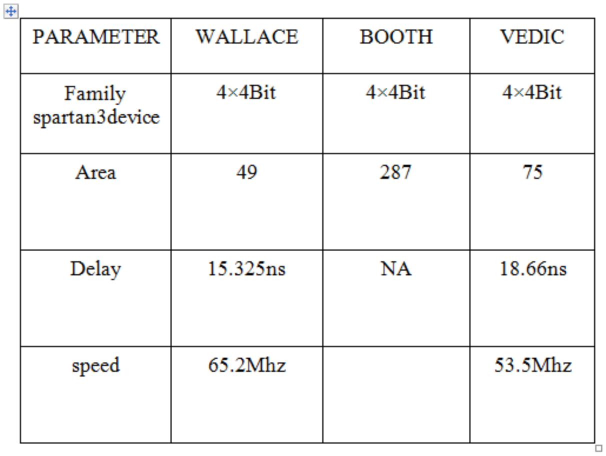


**Fig -6**: 64X64 bit RTL synthesis



| PARAMETER | WALLACE | BOOTH | VEDIC |
|---|---|---|---|
| Family spartan3device | 4×4Bit | 4×4Bit | 4×4Bit |
| Area | 49 | 287 | 75 |
| Delay | 15.325ns | NA | 18.66ns |
| speed | 65.2Mhz |  | 53.5Mhz |

**Table 1**: Comparison of different multiplier

## 4. CONCLUSIONS

Vedic multiplier is faster than other multiplier. The carry save adder vedic mathematics based MAC unit proves to be highly efficient in terms of speed and area. The area needed for vedic multiplier is very small as compared to other multiplier architecture.

## 5. REFERENCES

[1] Devika Jaina , Kabiraj Sethi and Rutuparna Panda, “Vedic Mathematics Based Multiply Accumulate Unit “ , 2011 International Conference.

[2] Teffi Francis, Tresa Joseph and Jobin K Antony , “Modified MAC unit for low power high speed DSP application using multiplier with bypassing technique and optimized adders”,2013

[3] Deepak S and Binsu J Kailath, “Optimized MAC unit design” ,2012 IEEE International Conference

[4] P.Jagadeesh ,Mr.S.Ravi and Dr.Kittur Harish , “Design of high performance 64 bit MAC unit”2013 International Conference IEEE

[5] Shamin Akhter “VHDL implementation of fast NxN multiplier based on vedic mathematic.” 2007IEEE

[6] Parth Mehta, Dhanashri Gawali , Conventional versus Vedic Mathematical Method for Hardware Implementation of a Multiplier.” 2009 IEEE

[7] 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 large scale integration (vlsi) systems, vol. 18, no. 2,february 20 10

[8] Ron S. Waters and Earl E. Swartzlander, Jr., "A Reduced Complexity Wall ace Multiplier Reduction, " IEEE Transactions On Computers, vol. 59, no. 8, Aug 20 10

[9] Shanthala S, Cyril Prasanna Raj, Dr.S.Y.Kulkarni, "Design and VLST Implementation of Pipelined Multiply Accumulate Unit," IEEE International Conference on Emerging Trends in Engineering and Technology, ICETET-09

[10] B.Ramkumar, Harish M Kittur and P.Mahesh Kannan, "ASIC Implementation of Modified Faster Carry Save Adder ", European Journal of Scientific Research, Vol. 42, Issue 1, 2010.

[11] R.UMA, Vidya Vijayan, M. Mohanapriya and Sharon Paul, "Area, Delay and Power Comparison of Adder Topologies", International Journal of VLSI design & Communication Systems (VLSICSj Vo1.3, No.1, February 2012

[12] V. G. Oklobdzija, "High-Speed VLSI Arithmetic Units: Adders and Multipliers", in "Design of High-Performance Microprocessor Circuits", Book edited by A.Chandrakasan,IEEE Press

[13] C. S. Wallace, "A suggestion for a fast multiplier," iEEE Trans. ElectronComput., vol. EC-13, no. I, pp. 14-17, Feb. 1964