Design of High Speed MAC (Multiply and Accumulate) Unit Based On "Urdhva Tiryakbhyam Sutra".

Parth S. Patel, Khyati K. Parasania

Abstract— **The multiplication and multiply-accumulate operations are expensive to implement in hardware for Digital Signal Processing, video, and graphics applications. A standard multiply-accumulator has three inputs and a single output that is equal to the product of two of its inputs added to the third input. The goal of this thesis is to investigate algorithms and architectures used to design multipliers and multiply-accumulators, and to create a multiply-accumulator. Often times in high speed designs the most time-consuming operations are pipelined to meet the system timing requirements. Therefore, the goal is to create a multiply-accumulator that is comparable in speed, but requires less area than a design using an industry standard multiplier and multiply-accumulator.**

Index Terms— **Adders, Multiply Accumulate Unit, Vedic mathematics, Vedic multiplier, Verilog**

I. INTRODUCTION

With the recent rapid advances in multimedia and communication systems, real-time signal processing like audio signal processing, video/image processing, or large-capacity data processing are increasingly being demanded .The multiplier and multiplier-and-accumulator (MAC) are the essential elements of the digital signal processing such as filtering, convolution, transformations and Inner products 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. The main motivation behind this work is to achieve high speed through VLSI design of MAC unit architecture using multipliers based on Vedic mathematics.

Hence these functional units must work in an optimized way consuming less area and power, and operating in higher speed.

Manuscript received June, 2015

***Parth S. Patel**, Electronics and communication, Gujarat Technological University). Ahmedabad, India,, +919638209114*

***Prof Khyati k Parasania**, Electronics and communication, Gujarat Technological University Ahmedabad, India,,*

II. MULTIPLY ACCUMULATE 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 obtained from the memory location and given to the multiplier block.

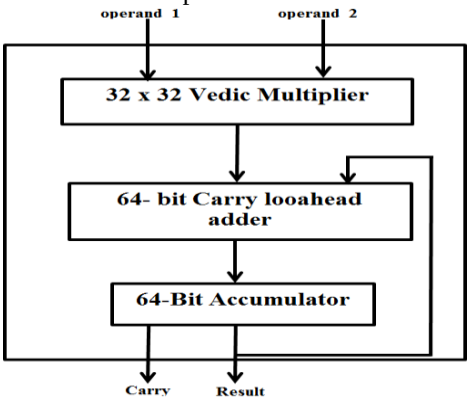

Figure 1: Architecture of MAC unit

1. **Multiplier**

   A multiplier can be divided into three steps. The First step is partial product from the multiplier and multiplicand. The Second is adder which adds all the partial products and convert in to the form of sum and carry. The last stage is Final addition in which final multiplication result is generated by adding the sum and carry. for ex. z = a x b + z.

2. **Accumulator**

   Accumulator basically consist of register and adder. Register hold the output of the previous clock from adder Holding output in Accumulator register can reduce additional add instruction. An accumulator should be Fast in response so it can be implemented with one of the fastest adder like carry look ahead adder , carry skip adder or carry select adder.

III. ADDITION ALGORITHMS

A. Ripple Carry Adder

The Ripple Carry Adder is made utilizing full adders just-bit information data is straight forwardly connected to the N-bit Adders for the

 All Rights Reserved © 2015 IJARCET

expansion operation and in the parallel addition is being created. In the event that the approaching is missing in the framework, first full snake is supplanted with the half viper. In swell convey viper the convey yield from every stage is undulated to the following stage. Figure 2 demonstrates the general piece graph of the carry Ripple Adder(RAC).

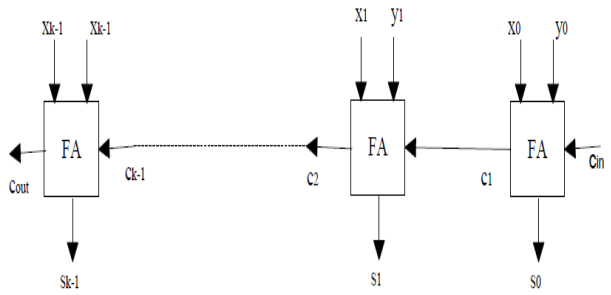

Figure 2 :Block Diagram of Ripple Carry Adder

The latency of n-bit determined by using the most pessimistic scenario propagation path. As demonstrated in to the above Figure the discriminating way as a rule starts at the x_0 and y_0 into continues through the propagations of carry chain to the Full Adder and ends at the n-1 output.

B. **Carry Look Ahead Adder**

In this Type of Adder figures gathering create motions and in addition gathering spread signs to abstain from sitting tight for a swell to figure out whether the combination of first part will produces the carry or not. On the view of point the propagation of carry and the system of a carry configuration of the real operand digits are not important. However its important when the carry is being produces propagated and obliterated.

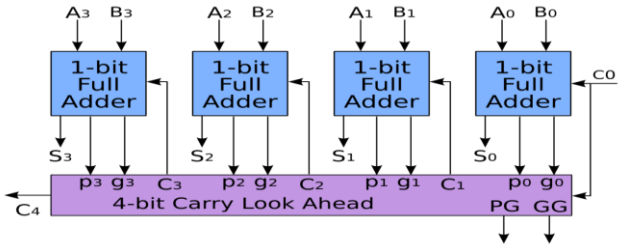

Figure 3 : Block Diagram of Carry Look Ahead Adder

On account of binary addition the produce,

engender and destroy signals are characterized by

the accompanying following comparisons.

$C_0 =$ $\underline{A * B}$ + $\underline{(A \text{ Xor } B)C_{in}}$

Carry Generator Carry Propagator

C. Carry Save Adder

Carry save adder is best suitable on adding more then 3- bits. The carry save adder is just a set of full adders and half adders. For an n-bit adder implementation, still carry is rippled to the next stage, of the same row, even though inputs to the lower next stage is ready, but kept in a wait state, until the sum and carry output didn't come from the above stage. This induces delay, now it can be optimized, if the carry out is passed diagonally to the lower next stage, instead of rippling to the next stage of the same row. Cary computation is not performed, but it is saved up to last row, where the results are obtained finally, in the last bottom row, carry is rippled however, but is significantly reduces the amount of delay occurred due to rippling operation. Fig 7 represents the carry save adder structure for a 4-bit addition.

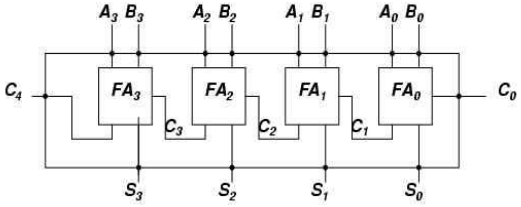

Figure 4 : Block level representation of carry save adder

IV. MULTIPLICATION ALGORITHMS

A. Booth Multiplier

Corner's calculation analyzes nearby combines of bits of the N-bit multiplier Y in marked two's supplement representation, including a certain bit beneath the minimum critical bit, y-1 = 0. For every bit yi, for i running from 0 to N-1, the bits yi and yi-1 are considered. Where these two bits are meet, the item collector P is left unaltered. Where yi = 0 and yi-1 = 1, the multiplicand times 2i is added to P; and where yi = 1 and yi-1 = 0, the multiplicand times 2i is subtracted from P. 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.

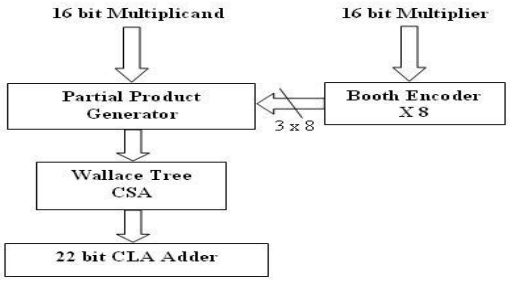

Figure 5 : Block Diagram of Booth multiplier

 All Rights Reserved © 2015 IJARCET

B. Multiplication using Vedic mathematics

I. URDHVATIRYAKBHYAM SUTRA Steps

1. We will take the right hand digit and multiply them together. This will give us LSB digit of the answer.
2. 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.
3. Multiply the LSB digit of bottom number with the MSB digit of the top one.
4. 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.
5. Finally , Simply multiply the LSB of the both number together to get the final product.

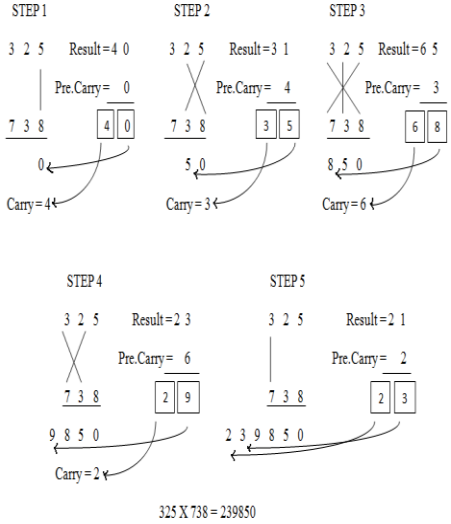

Figure 6 : Multiplication by using the Urdhva Tiryakbhyam Sutra

I. Nikhilam Sutra

II. 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 augmentation 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.

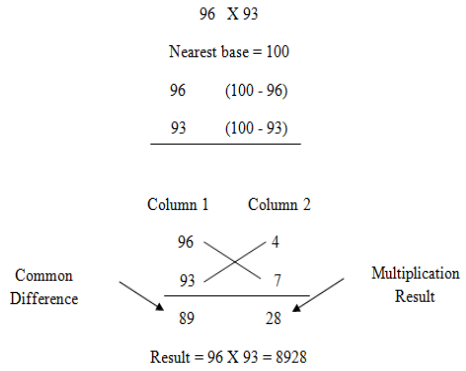

Figure 7 :- Multiplication by using Nikhilam Sutra

V. VEDIC MATHEMATICS ALGORITHM

A. 4x4 Vedic Multiplier

The 4x4 multiplication is decomposed into four 2x2 multiplications performed in parallel. This mechanism reduces the number of stages for the multiplication and thus reduces the delay of the multiplier. Fig 4 shows the block level representation of the 4x4 Vedic multiplier.

The advantages of this mechanism is that larger bit streams (say N-bits) can be divided into (N/2=n) bit length, which can be further divided into n/2 bit streams and this can be continued till we reach the bit stream width of 2-bits, and the can be multiplied in parallel, thus providing an increase in speed of operation. The selection of the adder is based on a comparative study described in section III.

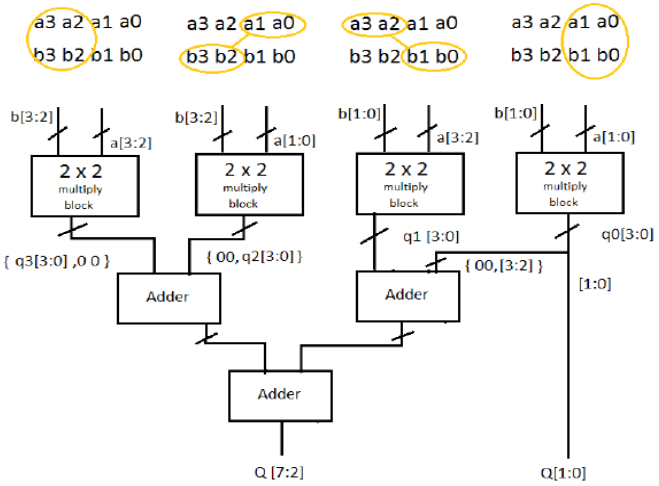

Figure 9: Block representation of 4x4 multiplier block

B. 32x32 Vedic Multiplier

The 32x32 multiplier is made by using four 16x16 multiplier blocks. The multiplicands are of bit size (n=32) where as the result is of 64-bit size. The input is broken into smaller block of size of n/2=16, for both the inputs. The newly formed 16x16 data blocks are applying to the input of 16x16 multiplier blocks. Fig6 repents the block level view of the 32x32 multiplier.

 All Rights Reserved © 2015 IJARCET

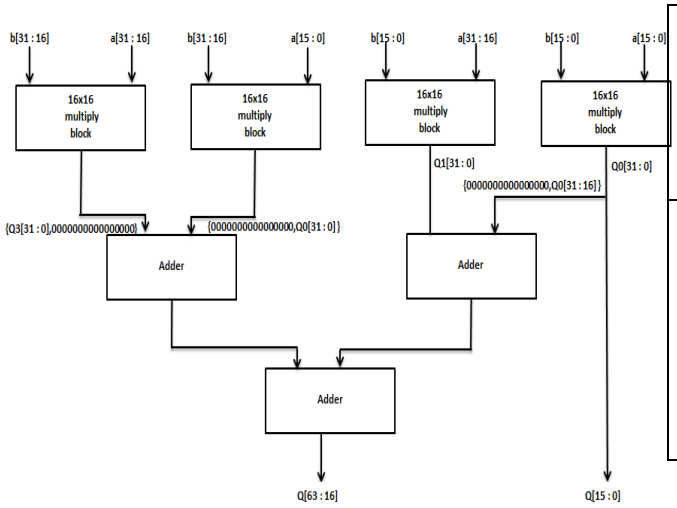

Figure 10: Block representation of 32x32 multiplier block

VI. RESULTS

A. RTL View of 32-bit MAC Unit

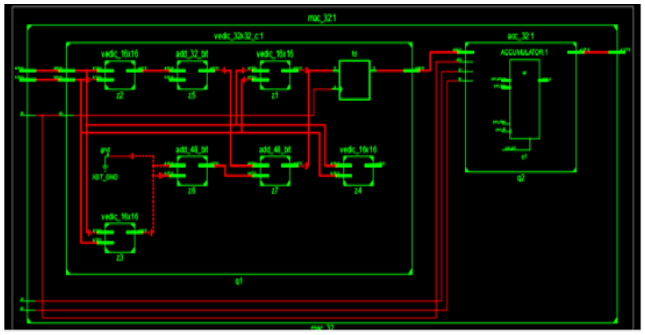

Figure 11 :- RTL View of 32-bit MAC Unit

B. Waveform of 32-bit MAC Unit

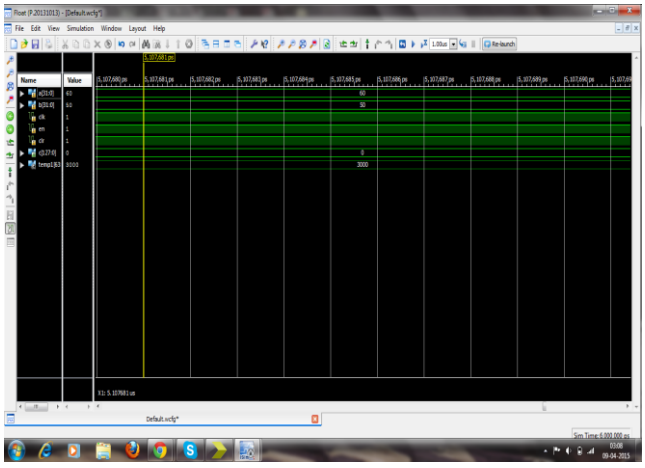

Figure 12 : wave form of 32-bit Mac Unit

C. Delay Comparisons Table

MAC	Logic Utilization	Available	Used	Delay	Utilization
32-bit MAC	Number of Slice	4656	41		30%
With out CLA	Number of 4 input LUT	9312	2614	4.31ns	28%
	Number of Bonded ios	232	195		84%
32-bit MAC With out CLA	Number of Slice	9312	1364	4.28ns	26%
	Number of 4 input LUT	9312	2486		26%
	Number of Bonded ios	232	129		55%

Table 1 :- Delay comparisons of 32-bit MAC Unit with CLA and without CLA

VII. CONCLUSION

- Vedic multipliers are much faster than the conventional multipliers. This gives us method for hierarchical multiplier design. Therefore, the design complexity has reduced for inputs of large no of bits and modularity is increased. Urdhva Tiryakbhyam, is algorithms, which can reduce the delay, power and hardware requirements for multiplication of numbers. the efficient implementation of the design various adders are studied, compared and among them carry look ahead adders is used for the final For implementation.

VIII. FUTURE SCOPE

- Future work includes the integration of the divider block, multiply and accumulate (MAC) unit, Later on, on the off chance that we need to upgrade this MAC unit for low power, we must utilize convey skip snake.

REFERENCES

[1] T.T Hoang, M. Sjalander, "Double throughput Multiply Accumulate Unit for Flexcore processor enhancements". IEEE International symposium on parallel and distributed processing, pp1-4, 2009

[2] Jagadguru Swami Sri Bharati Krishna Trithaji Maharaja, "Vedic Mathematics",Motilal Banarsidas Publishers Pvt. Ltd Delhi, 2009

[3] A.D. Booth, " A signed Binary multiplication Technique", Qrt. J.Mech. App. Math., Vol 4., no.2, pp.236-240,1951.

[4] Akhter S., "VHDL implementation of a fast NxN multiplier based on Vedic mathematic", pp. 472-475. ECCTD 2007.

[5] Shamsiah Suhaili and Othman Sidek, " Design and implementation of reconfigurable alu on FPGA", 3rd

 All Rights Reserved © 2015 IJARCET

International Conference on Electrical & Computer Engineering ICECE 2004, 28-30 December 2004, Dhaka, Bangaladesh, pp.47-56.

[6] Tam Anh Chu, "Booth multiplier with low power high performace Input circuitry", US patent, 6393454BL, May 21 2002

[7] Frank Marzona, "vedic mathematics: The Scientific Heritage of Ancient India", Proceedings of the Pennsylvania State System of Higher Education Mathematics association Conference, held at Mansfield University 1997-volume one.

[8] Aneesh R, Jijuk, Sreekumari B, "Design and Implementation of Bluetooth MAC core with RFCOMM on FPGA", proceedings of INDICON 2012.

[9] Aneesh R, Jijuk, "Design of FPGA based 8-bit RISC controller IP core using VHDL", proceedings of INDICON 2012 5

[10] Honey Durga Tiwari, Ganzorig Gankhuyag, Chan Mo Kim, Yong Beom Cho, Multiplier design based on ancient Indian Vedic Mathematics, International SoC Design Conference 2008.

[11] Sumita Vaidya and Deepak Dandekar, Delay-Power Performance comparison of Multipliers in VLSI Circuit Design, International Journal of Computer Networks & Communications (IJCNC), Vol.2, No.4, July 2010.

[12] S.S.Kerur, Prakash Narchi, Jayashree C N, Harish M Kittur and Girish V A Implementation of Vedic Multiplier For Digital Signal Processing, International conference on VLSI communication & instrumentation (ICVCI) 2011.

[13] Asmita Haveliya, A Novel Design for High Speed Multiplier for Digital Signal Processing Applications (Ancient Indian Vedic mathematics approach), International Journal of Technology and Engineering System (IJTES), Vol.2, No.1, Jan-March, 2011.

[14] Prabha S., Kasliwal, B.P. Patil and D.K. Gautam, Performance Evaluation of Squaring Operation by Vedic Mathematics, IETE Journal of Research, vol.57, Issue 1, Jan-Feb 2011.

Parth S. Patel Master of Engineering in Electronics and Communications Engineering , Hasmukh Goswami College of Engineering

Khyati Parasania Master of Engineering in Electronics and Communication Engineering , Assistant Professor at Hasmukh Goswami College of Engineering

 All Rights Reserved © 2015 IJARCET