INTERNATIONAL JOURNAL
FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Optimization of Delay IIN Pipeline Mac Unit Using Wallace Tree Multiplier

P.Samreen Aalia[1], K. Yogitha Bali[2]

[1]PG Scholar, [2]Assistant professor, Dept. of ECE, Sree Rama Engineering College, Tirupati, A.P, India

Abstract: A prompt pipelined MAC unit is implemented in this paper. Due to carrying propagations in additions, including additions of multiplications and additions of accumulations, we will experience a significant path delay as well as a significant increase in power consumption when using the conventional methodology. To resolve these issues, we are including a portion of improvements to the partial product reduction procedure. The pipelined MAC implementation does not complete the addition and accumulation of the most weighted bits until the partial product reduction step of the subsequent multiplication. In order to manage the overspill bits during the partial product reduction procedure, we are designing the small size adder which computes the total number of carries. By comparing with the conventional methodologies, we will get the diminished area as well as power consumption for the proposed pipelined MAC implementation.
Keywords: Digital circuits, logic circuits, multiplying circuits, pipeline processing, parallel prefix adders, Dadda multiplier.

I. INTRODUCTION

In digital signal processing applications, the Multiply Accumulate unit is the primitive block. Digital Signal processors (DSP) are used widely in applications like speech and audio coding, image processing, video processing, etc., most of these applications require sequential algorithms, and the DSPs chip Architecture is designed to perform sequential tasks. Computation of these applications involves multiplication as well accumulation operations & hence Multiply Accumulate (MAC) unit is very imperative in DSP applications, to achieve these applications in real-time with high performance. In the past, several methodologies were performed in scheming MAC structural designs through great computational effectiveness as well as enhanced power consumption.

A traditional mac is mainly composed of two main blocks namely multiplier and accumulator An A n-bit mac unit will have n bit multiplier and $(2n+\propto -1)$ adders. The alpha bits are the overload bits. Multipliers's$\propto$ role in today's digital signal processing and various other applications is crucial. Addition and multiplication are the most widely used basic operations in high-end applications. If observed keenly, almost 70-75% of instructions in microprocessor and other DSP require addition and multiplication. In mac also, adder and multiplier are the basic blocks. The designer mainly concentrates on the design of a circuit that is both fast and energy-efficient.

The Wallace tree, array, and other multipliers are just a few examples of multipliers, the Wallace tree multiplier showed better performance and for addition among various adders such as ripple carry adder, carry skip adder, carry save adder, and carry select adder, parallel prefix adders are considered as best choice.

A novel MAC design for high-speed achievement is proposed. Our first idea is to incorporate the integration of adders, which includes a portion that is a component of the accumulation's final addition as well as of the multiplication's final addition. In order to cut down on power use and critical path time, there is also the partial product reduction method. The most crucial bits are not finally included during the current multiplication in the suggested MAC unit. The second stage of partial product reduction involves the final addition and aggregation of the most significant bits.

MAC will perform the product of two n-bit numbers i.e. A_{in} and B_{in} and add the product sequentially to an accumulator and repeats the procedure for n cycles and produces output Z whose size is $(2n+1)$ bits.

$$Z = Ain \times Bin + C \quad ----(1)$$

MAC unit is comprised of an accumulator and multiplier. The multiplierer has a partial products generator, the network for summation, and addition at final stage. A normal MAC design has appeared in Fig.1. It involves a multiplier register and aggregated adder. Operands X and Y are multiplied and the results are added to the accumulator. Accumulator output is taken back as one of the inputs to the same accumulator. An optimized MAC unit architecture can be achieved by combining proper multiplication schemes with an effective fast adder network.
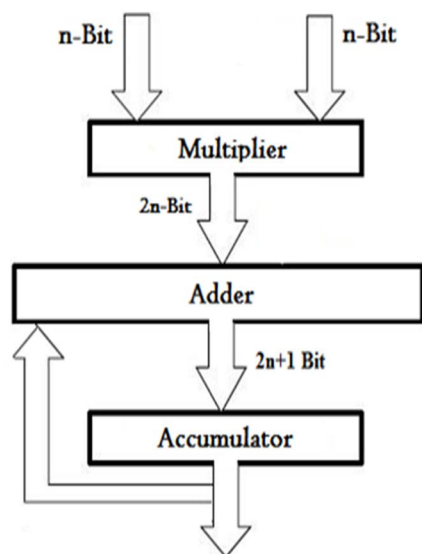
Fig. 1 Architecture of a Basic n- bit Multiply Accumulate (MAC) unit

Pipelining is the process of accumulating instruction from processor to through a pipeline. Instructions enter from one end as well as leaves from another end. The use of a pipeline improves the design's overall throughput. The pipeline itself interpreted into different stages, which would then be linked together as well as to create a pipeline configuration.

The MAC unit's entries were retrieved through memory cells as well as transmitted to a multiplication block, it conducts multiplication as well as transfers the outcome to the adder, who then preserves the outcome in a memory region. This entire procedure should be done in one clock cycle. For this configuration, the accumulator is a carry selection adder.

A. Multiplier Unit

Multipliers are used in multiply accumulate unit, generally in multiplier have three parts those are partial product generation and accumulation and final product we will get after reduced to two rows. Partial products are generated by using and gates and accumulation of partial products are done by using full adder as well as half adders. Here carry propagations are not allowed. If the carry propagations are performed here the propagation delay of overall circuit is increased drastically.

B. Adder Unit

Not only the multiplier but also adders greatly influence MAC performance. Hence design-performance full adder is needed for the day. Different adders circuit are analysed in order to choose the appropriate adder for improving the performance of MAC Unit. Initially ripple carry adder is studied. Due to the high delay it is not preferred. Later carry skip adder is analyzed. Although the speed is improved compared to ripple carry adder, it is not effectively reduced. next carry save adder is analyzed .although parallel operation is performed in the initial stages it uses carry propagation in its final stage which is not preferred. finally parallel prefix adders, from the analysis can be observed as the best choice for applications like MAC and DSP.

II. RELATED WORKS

The two stage MAC unit structure was shown in Fig. 2. It is composed of three stages termed as formation of partial products, accumulation of partial products, as well as alpha-bit addition.

The central MAC device is made up of
1) Generation of PP
2) Reduction of PP
3) Alpha-bit addition

For forming the partial products in the unsigned MAC structure and gates can be directly utilized. Various partial product generation algorithms are known to produce signed Partial product matrix for signed MAC unit. To construct the signed partial product matrix, we used the Baugh-Wooley multiplier algorithm.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue XI Nov 2022- Available at www.ijraset.com

The multiplication procedure for unsigned MAC unit:

The first stage in the MAC unit design is partial product generation and the PPM can be obtained by placing the register accumulation bits with the stage of partial product development.

The accumulation process has progressed to the second level, here we will perform the $(k+\alpha)$ bit addition. In this accumulation stage we will utilize carry propagate adder in the last row of the LSB position only. Based on Dadda multiplication approach we will reduce the partial product array to two rows without utilizing carry propagations. The proposed block diagram for this work is shown below.
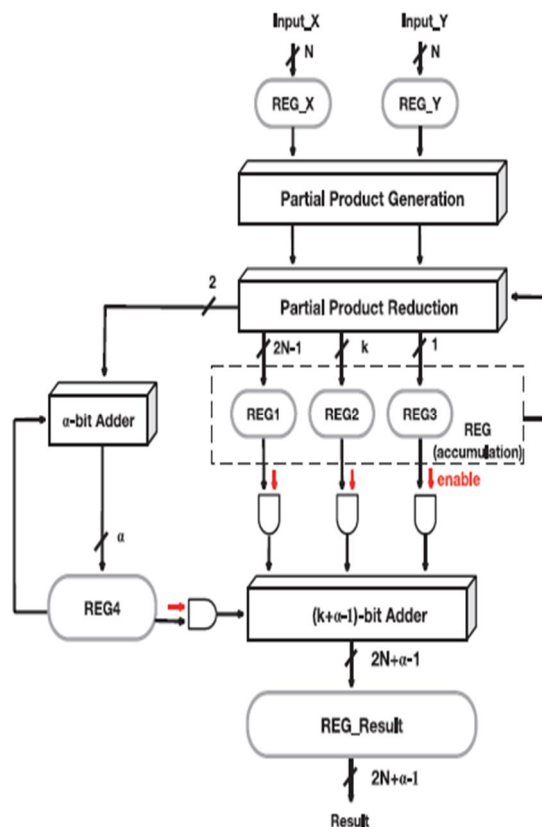

Fig. 2. MAC architecture

Overflow bits can be handled by α –bit adder. Based on number of bits considering to perform Pipelined MAC unit the α –bit adder width will alter. Depending on bit width we need to change the addition of α –bit adder design. For accumulating the partial products, full adders and half adders are used and the carries generated by full adders will not propagate to the next adder but to the next stage. To save the power, we are applying the gating technique to perform the second stage addition in the last stage only.

MSB positioned bits will not be added in the first stage of accumulation; instead, they will be added in the second. The R1, R2, and R3 values obtained in the previous stage are contributed to the partial product reduction stage's following stage. To prepare for final-stage addition, the MSB bits must be gathered in the second step.

Throughout the partial product reduction stage, designers compress the partial product matrix to 2 rows by using the Dadda tree representation. The main reason is because this partial product matrix is different from the typical partial product matrix. Also for partial product matrix, researchers expected that the Dadda tree alternative, as opposed to the Wallace tree multiplier approach, could even require less counters. We'll perform (2N-k-1) bit addition by diminishing Partial Product Matrix to two rows.

For unsigned bit, the highest significant bit was passed though the OR gate and AND gate. For signed bit, the highest significant bit was performed addition and it is subtracted with minus 1. Hence, a three row partial product matrix is also attained. In register accumulation, the three-row PPM is saved. The result is to form the next partial product matrix for the partial product generation process.

When the Partial Product Matrix is reduced to two rows, the column with the highest significance bit only produces two product names.

The outputs of this translation circuit was passed to the input of alpha-bit addition. The accumulation was done in the alpha-bit addition which is shown in Fig. 3. As a consequence, if we really need to get the final outcome by utilizing the activate signal. If the enable signal is high, it shows the final output.
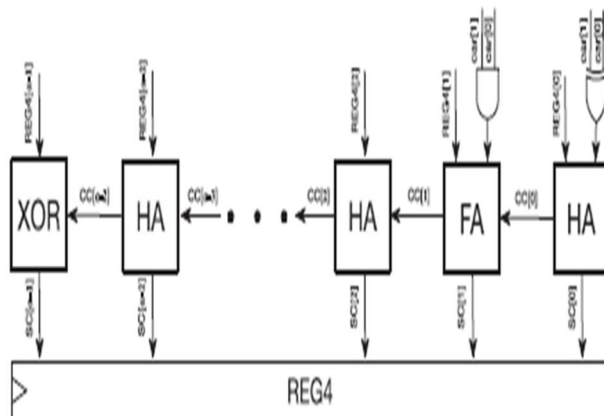

Fig.3. Alpha-bit addition for unsigned MAC unit.

III. METHODOLOGY

From the existing mac unit it can be observed that delay is more since for alpha bit addition a ripple carry adder is used and for multiplier dadda multiplier is used. Since the multipliers and adders are basic modules used in the MAC unit, the performance of these modules impact performance of MAC unit.

The proposed architecture has the following characteristic:

It minimizes the carry propagation chain by using the Wallace tree multiplier.

Instead of ripple carry adder in $(k + \alpha - 1)$ bit addition parallel prefix adder is used for minimizing the delay.
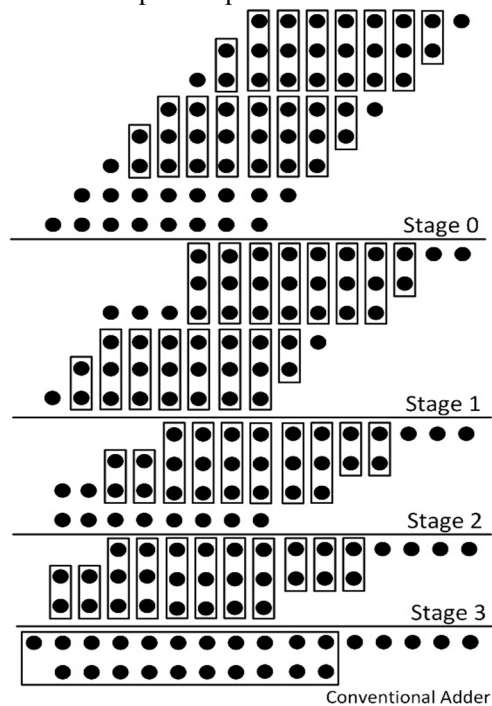

Fig. 4. 8-bits Wallace tree multiplication

Now the ripple carry adder is replaced with parallel prefix adder. The main key feature of parallel prefix adder is its high speed which is major concern in MAC unit.
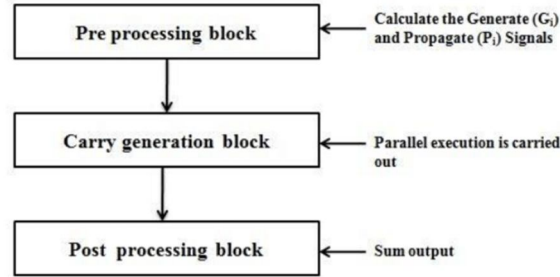
Fig. 5. General Block Diagram of Parallel Prefix Adder

1) *Pre computation:* it is the first step in parallel prefix addition. From the given inputs A and B it generates Propagate and generate values.

$$P = A \; xor \; B - - - - - (2)$$
$$G = A \; and \; B - - - -(3)$$

2) *Carry Generation:* it is the intermediate stage in the parallel prefix addition. The carry is generated and propagated to the next stage using a combination of black cells, buffers and gray cells. The black cell, buffer and gray cell are shown below.
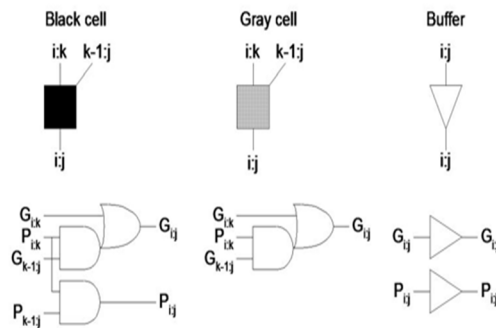

Fig.6 Block diagrams of black cell, Gray cell and buffer.
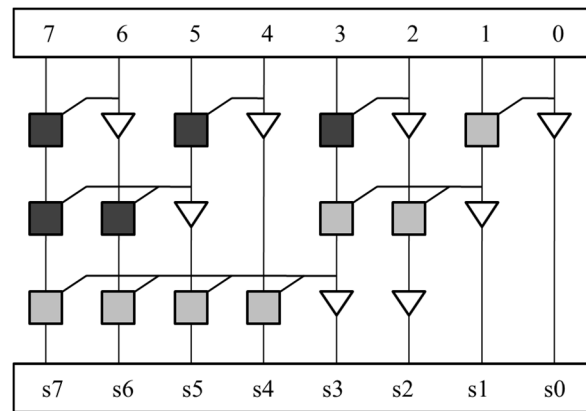

Fig. 7 .Parallel Prefix adder

3) *Final Computation:* it is the final step in the process of parallel prefix addition. In this stage final outputs sum and carry are calculated.

$$Sum = \; P_i \char`\^ C_{i-1} \quad ------ (6)$$

Finally In the mac unit for generation of partial products, a Wallace multiplier is used and in $(k + \propto -1)$ addition a parallel prefix adder is used. By implementing this proposed MAC unit, the delay is minimized.

IV. RESULTS

A pipelined MAC unit design is proposed, which includes a Wallace tree multiplier and a parallel prefix adder. The suggested MAC unit was designed and implemented in the Xilinx environment. We are able to compute both signed and unsigned pipelined MAC units by utilizing a single architecture. The comparison table, RTL schematic, the technology schematic and simulation waveform is shown below.

Table: 1 Comparison of Existing and Proposed MAC unit.

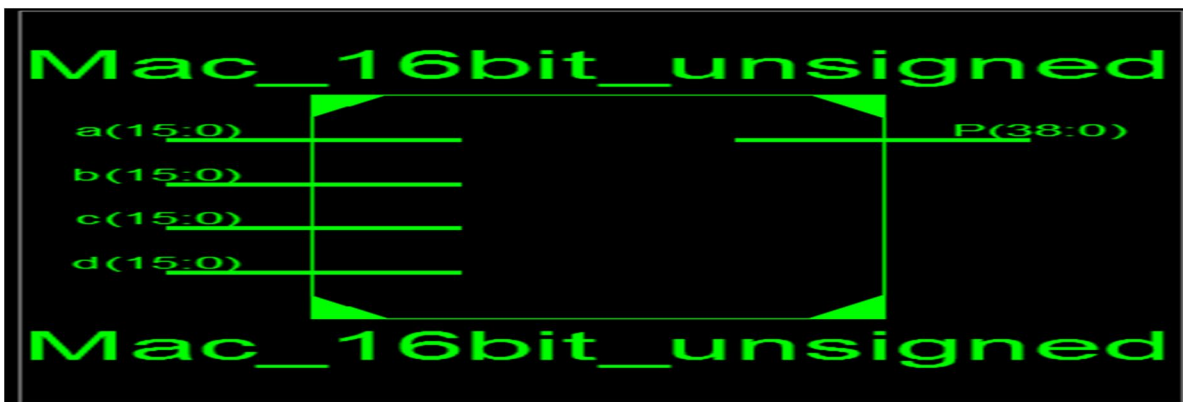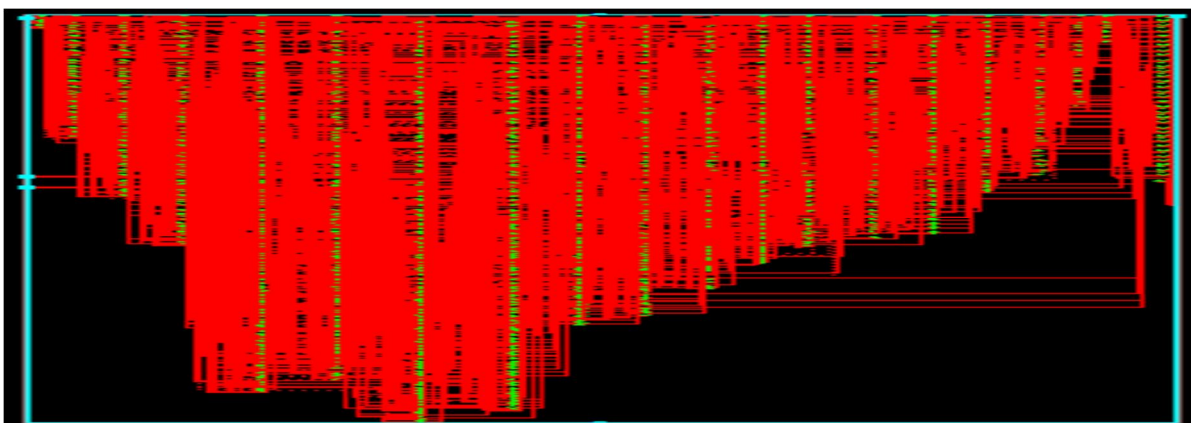| Name	Area	Delay(ns)
Existing mac	763	23.837
Proposed mac	879	14.292

Fig.8. RTL schematic
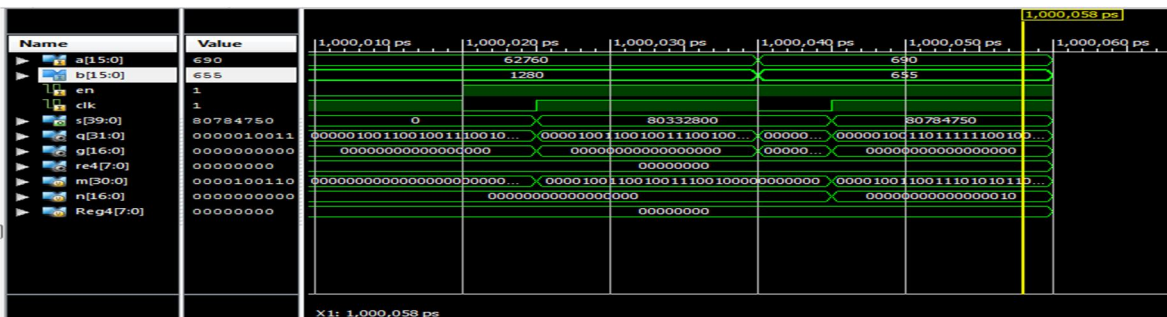

Fig.9. Technology schematic

Fig. 10.simulation waveform of proposed MAC

V. CONCLUSION

In this paper, a novel two stage pipelined MAC unit is proposed. In this mac unit for the purpose of addition a parallel prefix adder and for the purpose of multiplication a Wallace tree multiplier is used to enhance the speed which is critical in MAC unit. Further various adder architectures we can use for the trade-offs among delay, area, and power. Additionally, by using different architectures in place of the multiplier and accumulator (adder), we are able to create a variety of MAC unit models.

REFERENCES

[1] K. Paldurai, K. Hariharan, G. C. Karthikeyan and K. Lakshmanan, "Implementation of MAC using area efficient and reduced delay vedic multiplier targeted at FPGA architectures," *2014 International Conference on Communication and Network Technologies*, 2014, pp. 238-242

[2] S. Swettha, S. Rashmi, N. S. S. Reddy and R. Hemalatha, "Area and power efficient MAC unit," *2018 Conference on Signal Processing And Communication Engineering Systems (SPACES)*, 2018, pp. 202-205

[3] S. Asif and Y. Kong, "Design of an Algorithmic Wallace Multiplier using High Speed Counters", in Proc. IEEE Int. Conf. Computer Engineering & Systems, Cairo, Egypt, 2015, pp. 133-138.

[4] C.W. Tung and S.H. Huang, "Low-Power High-Accuracy Approximate Multiplier Using Approximate High-Order Compressor", in Proc. IEEE Int. Conf. Communication Engineering and Technology, Nagoya, Japan, 2019.

[5] Saini, S. Agarwal and Aditi Kansal, "Performance, Analysis and Comparison of Digital Adders", in Proc. IEEE Int. Conf. Advances in Computer Engineering and Applications, Ghaziabad, India, 2015, pp. 81-83.

[6] K. Du, P. Varman, and K. Mohanram, High performance reliable variable latency carry select addition, in Proc. Design, Automat. Test Eur. Conf. Exhib., Mar. 2012, pp. 1257–1262.

[7] R. Hegde et. al. Energy-efficient signal processing via algorithmic noise- tolerance. In Proc. ISLPED, pages 30–35, 1999.

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 ☉ (24*7 Support on Whatsapp)