Optimized MAC Unit using Binary Carry Select Adder and Counter-Based Modular Wallace Tree Multiplier

Dr. Vidyasaraswathi H N[1], Navaneetha H D[2]

[1] Assistant Professor, Department of Electronics and Communication Engineering, Bangalore Institute Of Technology,Bangalore, India

[2.]Student, Department of , Department of Electronics and Communication Engineering, Bangalore Institute Of Technology,Bangalore, India

***Abstract*— The multiply accumulate unit are crucial components in high-performance computing devices, particularly embedded systems. These units consist of a accumulator, multiplier, and adder, and perform operations by reading input from memory cells, multiplying it within MAC multiplier block, adding the result, and storing it back in memory—all within one clock cycle.To increase the efficiency of these units, a novel design MFA-MAC is proposed that improves speed while consuming less power. This design employs three separate blocks are combined with the high-speed binary carry select adders inside the MAC units: final carry generators (FCG), final sum generators (FSG), and half sum with carry generators (HSCG). A CMWTM was used for the multiplier design, incorporating power-saving techniques. Additionally, the application of MFA significantly reduces power usage by simply turning off at certain areas of the circuit when they're not in use, thus achieving a more power-efficient and faster MAC unit suitable for embedded systems.**

***Keywords*— Multiply-accumulate unit ,counter based modular Wallace tree multiplier, partial product reduction.**

I. INTRODUCTION

A new MAC unit tailored for embedded systems. is proposed, focusing on less power consumption and improving speed. Operation of a MAC unit is Inputs are accessed from memory cells,multiplication is done by the multiplier; the result is then fed to adder and ultimately saved back to memory[1].

This unit integrates a CMWTM with HSBCSA to increase the efficiency of arithmetic operations. The CMWTM is noted for its power-saving techniques, making it an ideal choice for low-power embedded systems[2].

A new high-performance MAC unit is proposed to improved efficiency by addressing consumption of power and delay issues commonly associated with traditional MAC units. The novel approach integrates a multiplier CMWTM with an adder HSBCSA to optimize arithmetic operations[1].

In this MAC unit, significant modifications are made to lower the length of carry propagations, which typically consume more power and introduce delays. Specifically, instead of handling the last addition of most significant bits during the multiplication phase as is customary, the proposed MFA-MAC design incorporates this addition into the PPR stage of the CMWTM. This adjustment allows for the shortening of the critical path delays and power loss, as the final addition is processed during the subsequent multiplication's PPR phase, resulting in shorter carry propagation paths.

II. LITRATURE SURVEY

Jeyakumar Ponraj et al. [1], the multiply accumulate retrieves inputs from memory, multiplies them with a multiplier block, and sends the result to an adder for storage, completing the process in one cycle. A new Multiply Accumulate is introduced, incorporating HSBCSA and a power-efficient CMWTM.

G. Park et al. [2], Presents a new, energy-efficient multiply-accumulate (MAC) processing approach utilizing interleaved approximate multipliers to mitigate error accumulation. The design incorporates balanced error generation and relies on a thorough analysis-driven development of positive and negative multipliers.

Munivenkatappa et al. [3], Tackles the challenges associated with multipliers, such as area, power, complexity, and speed, by introducing the Additive Multiply Module (AMM). This space-efficient MAC multiplier provides improved efficiency, reduced space requirements, and lower latency compared to conventional designs.

Sakthivel et al. [4], the analysis of both current and new designs was analyzed, and outputs demonstrate that new design 64-bit CSLA achieves an average decrease in consumption of power and lower area utilization relative to the current design.

Vidyasaraswathi. H.N et al. [5], presents the design of an area-efficient, high-performance 8x8 multiplier based on Residue Number System (RNS). Converting large numbers to RNS speeds up arithmetic-intensive algorithms in signal processing. To address speed and gate count issues, by using the Ladner Fischer parallel prefix adder for decrease delay and area.

Akarsha Yadav et al. [6], presents the design of an area-efficient, high-performance FIR filter (4-tap, 8-tap, 16-tap) using the Residue Number System (RNS). RNS enhances fast arithmetic but increases delay and logic gates. To address these issues, the Ladner Fischer parallel prefix adder is employed to reduce delay and area.

III. METHODOLOGY

The proposed methodology in the document focuses on a low-power, high-speed MAC unit designed for embedded systems, consisting of two primary parts: multiplier design and adder design. This unit features a two-stage pipeline structure. In the initial stage, the multiplication process is handled by the CMWT multiplier, which integrates the addition part to reduce carry propagation. During multiplication, a part of the addition is performed, and overflow is managed using a HSBCSA. The outcome of the multiplication are stored in registers, specifically REG 1, REG 2, and REG 3, after the final addition.

As a outcome, the register accumulation as shown in fig1 contains three parts: REG 1, the first row; REG 2, the second row, which the user can define; and REG 3, the third row, containing one bit. REG 1, REG 2, and REG 3 consist of 7 bits, 3 bits, and 1 bit, respectively. Each bit in the register accumulation initially starts with the value "0."

In the second stage, HSBCSA is utilized to produce the final accumulation result. This adder efficiently counts all carries and manages overflow effectively. Additionally, a gating technique is utilized to Conserve power by activating the second stage only during the final cycle of the multiply accumulate operations. The proposed architecture integrates a component of adder into the multiplier process, uses AND gates for unsigned MAC units, and considers the sign bit's effects for signed MAC units. This methodology ensures a reduced carry propagation length, efficient handling of overflow, and power savings during the final cycle, making it highly suitable for embedded systems.

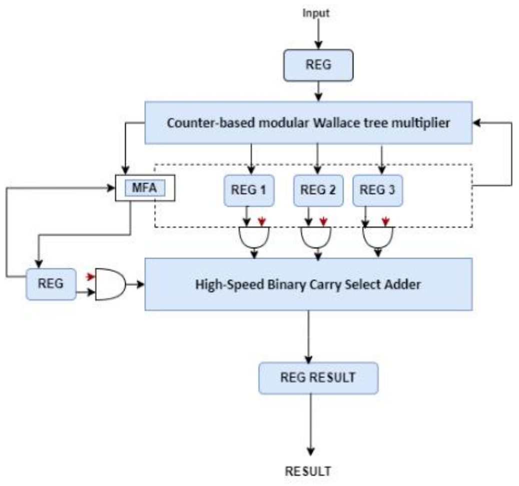
Input
REG
Counter-based modular Wallace tree multiplier
MFA
REG 1
REG 2
REG 3
REG
High-Speed Binary Carry Select Adder
REG RESULT
RESULT

FIG1: Methodology Flow

STAGE 1: In this CMWTM design is used in multiply accumulate unit.

The CMWT multiplier as shown in fig2 begins with the binary numbers to be multiplied, provided as inputs. Initially, partial products are produced through AND gates, creating a matrix of partial products. The first stage of partial product reduction employs 7:3 counters is shown in fig4, which take seven bits as input and produce three bits as output, significantly decreasing the number of partial products. The second stage uses 4:2 compressors to further reduce the partial products, with each compressor taking four bits as input and generating two bits as output. In next stage, 3:2 compressors are used to decrease the remaining partial products, where each compressor processes three bits and outputs two bits.

The last stage involves a carry-save adder (CSA) to accumulate the reduced partial products without immediate carry propagation, followed by a carry look ahead adder (CLA) to efficiently handle the final carry propagation and produce the final carry and sum outputs. And then, these outputs are put together to produce the outcome of the multiplication. The entire process, from input to the final product, highlights the stages of PPR and the efficient accumulation of results, ensuring high-speed operation with decreased consumption of power and minimal hardware complexity.

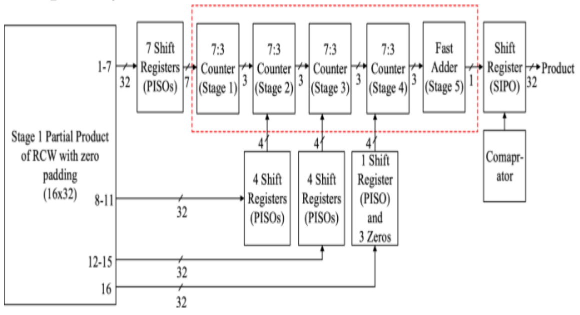

FIG2: CMWTM DESIGN

A multiplexer is a combination circuit having one output line, 'n' selection lines, and a maximum of 2n data inputs. The values of the selection lines will determine which of these data inputs is connected to the output. There are two feasible combinations of ones and zeros because there are 'n' selection lines. Thus, just one data input will be chosen for each combination. Another name for a multiplexer is a mux. The 4x1 Multiplexer comprises two selection lines (s1 & s0), one output (Y), and four data inputs (I3, I2, I1, & I0).

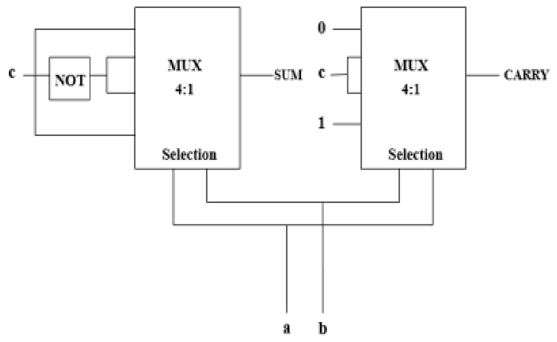

Fig3: Modified Full Adder

This modified full adder design as shown in fig3, two 4:1 multiplexers are used, with `a` and `b` serving as the selection lines for both multiplexers. The input `C` is fed into one multiplexer, producing the sum as the output. The inputs `0`, `C`, and `1` are provided to the other multiplexer, which outputs the carry. By analyzing the conventional full adder design, the multiplexer-based design is developed through its truth table and input-output behavior when logic 0 and logic 1 are given to respective input lines. This multiplexer-based adder design demonstrates a significant reduction in gate count, resulting in a more efficient implementation.

The 7:3 counter circuits are constructed using four standard full adders. Inputs I1 to I7 represent the 7 input bits, which are partial product terms. The outputs are Sum, C1, and C2, where C1 and C2 denote the first and second carry bits, respectively. The intermediate sum as well as carry bits are represented by SFA1, SFA2, CFA1, CFA2, and Carry. Initial three partial product bits, I1, I2, and I3, are fed into MFA1. The next three bits, I4, I5, and I6, are fed into MFA2, and I7 is fed into MFA3. The sum outputs from MFA1 and FA2 are further processed in MFA3, and Carry bits from these adders are handled by MFA4. At end sum is generated by FA3, MFA4 produces the carry bits C1 and C2, anyway.

In this architecture, counters are avoided due to their use greatly increases complexity, latency, and consumption of power. Below equations are for the Sum, C1, and C2 based on AND and XOR gates:

$$\text{Sum} = w8 \oplus w1 \quad (1)$$

$$C1 = w14 \oplus w1 \quad (2)$$

$$C2 = w16 \oplus w1 \quad (3)$$

Where, $w1 = I1 \oplus I2$, $w2 = I1 \cdot I2$, $w3 = I3 \oplus w1$, $w4 = I3 \cdot w1$, $w5 = w2 + w4$, $w6 = I4 \oplus I5$, $w7 = I4 \cdot I5$, $w8 = I6 \oplus w6$, $w9 = I6 \cdot w8$, $w10 = w7 + w9$, $w11 = I7 \oplus w3$, $w12 = I7 \cdot w3$, $w13 = w8 \cdot w11$, $w14 = w12 + w13$, $w15 = w5 \oplus w10$, $w16 = w5 \cdot w10$, $w17 = w14 \cdot w15$.

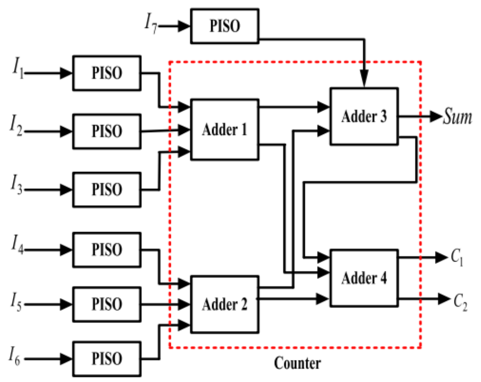

FIG4: Structure of 7:3 counter

STAGE-2: The HSBCSA is a adder design used in the multiply accumulate unit consists of three major blocks: the Half Sum Carry Generator (HSCG) , the Final Sum Generator(FSG),Final Carry Generator(FCG) Figure 5 illustrates the design of the HSBCSA adder.

2.1 The HSCG design will produce three critical signals: L, M, and N. Signal L is used to assess whether the sum of bits at specific positions is smaller than or equivalent to a predetermined value. Signal M is responsible for determining intermediate carries within a block, aiding in efficient carry propagation. Finally, Signal N serves as the complement of the half sum, contributing to the accurate calculation of the final sum. Signal L is a logical AND operation, M is a logical OR operation merged with the AND results, and N is the half-sum complement.

L0 = (P0 • Q0) (4)

L1 = (P0 • Q0)|((P1|Q1) • L0)… (5)

(P1 • Q1)|(P1|Q1) • L0 = (P1|Q1)•((P1•Q1)|L0) (6)

(P1|Q1)•((P1•Q1)|L0 = (P1|Q1)•(P1•Q1)•L0 (7)

L1 = (P1|Q1) • (P1 • Q1)•L0 (11) Ln− 1 = (Pn− 1|Qn− 1) • (Pn− 1 • Qn− 1)•Ln− 2 (8)

M0 = (P0|Q0) (9)

M1 = (P0|Q0) • M0 (10)

Mn− 1 = (Pn− 1|Qn− 1) • Mn− 2 (11)

N0 = (P0 ⊕ Q0) (12)

N1 = (P1 ⊕ Q1) (13)

Nn− 1 = (Pn− 1 ⊕ Qn− 1) (14)

2.2 This FCG will generates intermediate carries and the final carry out using NAND gates.The delay in FCG is due to two gate delays following the block’s carry-in arrival.

C1 = L0 •(M0 • carry − in) (15)

C2 = L1 •(M1 • carry − in) (16)

C3 = L2 •(M2 • carry − in) (17)

Cn = Ln− 1 •(Mn− 1 • carry − in) (18)

2.3 The FSG uses XNOR gates to give the end sum outputs. Each step uses one XNOR gate, introducing one gate delay after the carry-in arrival.

S0 = N0 ⊕carry – in (19)

S1 = N1 ⊕C1 (20)

S2 = N2 ⊕C2 (25) Sn− 1 = Nn− 1 ⊕Cn− 1 (21)

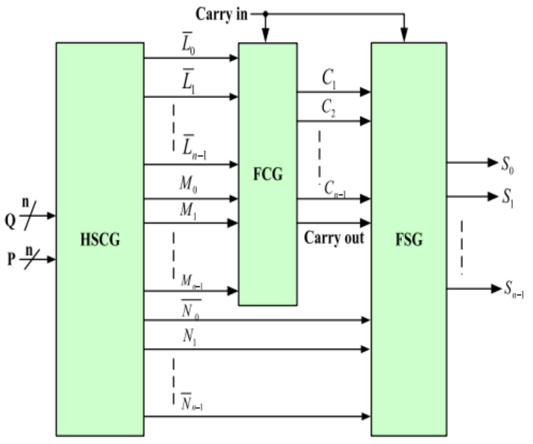

FIG5: HSBCSA Block design

IV. RESULTS

The experiment is done in Xilinx Vivado 19.1 for the purpose of synthesis, analysis, and implementation. MAC confirms through sinusoidal signal. Fig. 7 signify RTL schematic of proposed method. Fig. 6 signify simulation waveform of proposed method. Table 1 tabulates assessed values of area, power and delay.These values are compared with CMWTM-HSBCSA-ES method to proposed method. In area comparison it is reduced by 2.89%, power it is reduced to 11.41% and delay is reduced by 23.50%.

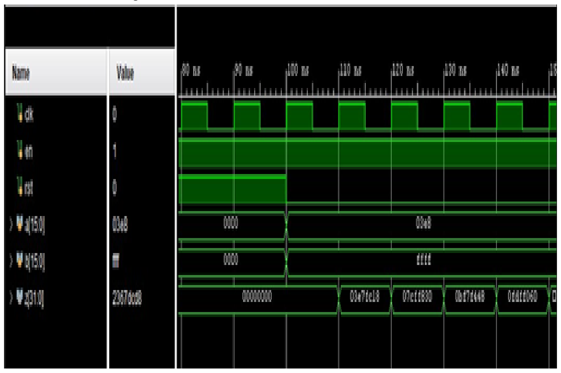

FIG 6: Simulation waveform

The above fig6 shows the waveform of the multiply accumulate unit. The waveform tracks how inputs are processed through multiplication and addition, leading to the updated result stored in the accumulator.

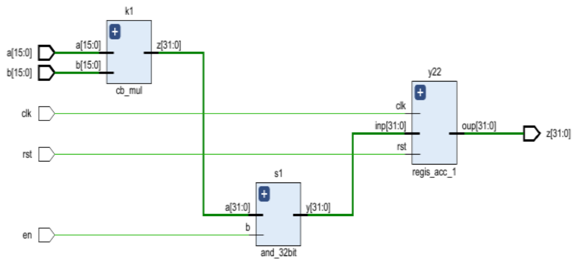

FIG7: RTL schematic of proposed MFA-MAC design.

This fig7 is the RTL schematic of the new design MFA-MAC. In this there are multiplier, adder and accumulator blocks this perform the multiplication and addition for the given two inputs and accumulates in register.

TABLE 1. COMPARISION TABLE.

MAC Unit	AREA (LUT)	POWER (in W)	DELAY
CMWTM-HSBCSA-ES method [1]	485	38.759	12.143
MFA-MAC	471	34.337	9.290

CONCLUSION

A more power-efficient and faster multiply-accumulate unit was successfully implemented for embedded systems. Utilizing Xilinx VIVADO for synthesis and Verilog for coding, the proposed design's effectiveness was evaluated based on delay, area, dynamic consumption of power and speed. The CMWTM-HSBCSA-ES method exhibited lower consumption of power compared to existing methods. Ongoing studies in optimization reveal promising strategies, and future work could expand this design by incorporating a Novel Hybrid Optimization algorithm to further reduce consumption of power and enhance the speed of MAC units by applying MFA method power or delay will reduce

REFERENCE

[1] Jeyakumar Ponraj, R. Jeyabharath, P. Veena, Tharumar Srihari, “High-performance multiply-accumulate unit by integrating binary carry select adder and counter-based modular wallace tree multiplier for embedding system”, the VLSI Journal 23 (2023).

[2] G. Park, J. Kung and Y. Lee, "Design and Analysis of Approximate Compressors for Balanced Error Accumulation in MAC Operator," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 68, no. 7, pp. 2950-2961, July 2021.

[3] Munivenkatappa, Nagabushanam & Srikanth, Skandan & Mupalla, Rushita & Kumar, Sushmitha & K, Swathi. Optimization of Power and Area Using VLSI Implementation of MAC Unit Based on Additive Multiply Module, International Journal of Electrical and Electronics Research (2022).

[4] Ramachandran Sakthive, G. Ragunath Low power area optimized and high speed carry select adder using optimized half sum and carry generation unit for FIR filter, Journal of Ambient Intelligence and Humanized Computing. (2021).

[5] Akarsha Yadav, Vijaya Prakash, and Vidya Saraswathi, Design of Modified RNS-PPA Based FIR Filter for High-Speed Application, European Journal of Engineering and Technology Research ISSN: 2736-576 (2022).

[6] Vidyasaraswathi. H.N, Akarsha S.N, Area Efficient Rns-Ppa Multiplier Design For Highspeed Application, International Journal of Research Publication and Reviews, Vol 3, no 5, pp 3165-3175, May 2022.