Journal of Nonlinear Analysis and Optimization Vol. 15, Issue. 1, No.8 : 2024 ISSN : **1906-9685**



DESIGN AND IMPLEMENTATION OF MAC USING APPROXIMATE MULTIPLIER

S.Selvakumar Raja Principal & Professor, ECE Department, Kakatiya Institute of technology and science for women, Nizamabad, Telangana, India, Email: kitswnzb@gmail.com
 M.Mahipal Associate Professor & HOD, ECE department, Kakatiya Institute of technology and science for women, Nizamabad, Telangana, India, Email: <u>kitswnzb@gmail.com</u>

ABSTRACT

Monetary operators are essential arithmetic components in many applications. Usually, these applications require large amounts of multiplications, leading to significant power usage. Within error-tolerant systems, an approximation multiplier is an innovative technique used to reduce critical path time and power consumption. An approximation multiplier may sacrifice accuracy in favor of improved performance and decreased energy usage. This article describes an accurate 4-2 compressor with a configurable approximation multiplier that may truncate partial products dynamically. Furthermore, a multiplier and accumulation (MAC) unit is suggested. The suggested MAC, with an approximation multiplier, can adjust the power and precision required for real-time multiplications according to user specifications.

Key words : compressors, approximate multiplier and MAC.

1. INTRODUCTION

Multipliers are essential arithmetic functional units in various domains such as artificial intelligence, DSP, computer vision, multimedia processing, and image recognition. These applications usually require a large number of computations that use a significant amount of electricity. Power consumption poses a major challenge in deploying these apps, especially on mobile devices. Many studies have proposed methods to reduce the power consumption of multiplier circuits. If the applications involve human sensitivities or allow for error tolerance, using approximation multiplication can reduce the power consumption of the multiplier. Human sensory capacities, such as limited vision and hearing, do not necessitate exact computational results. Approximation multipliers decrease cell space, time delay, and power consumption at the cost of precision. There are two types of approximate multipliers. Dynamic voltage scaling regulates the timing of the multiplier in the first way. A reduction in voltage results in an increase in the delay along the critical path of the multiplier. Violations of the time path cause errors, which therefore lead to approximate outcomes. The second category focuses on altering the operating features of multipliers by redesigning certain multiplier circuits, as the Wallace Tree Multiplier and Dadda Tree Multiplier. Previous studies on reconstructing multipliers have presented inaccurate m-n compressor designs with n outputs and m inputs. Erroneous compressors were used to compress partial products throughout the multiplication process due to significant route delay and energy expenditure.

Early approximation multipliers mainly provided consistent output accuracy and needed power. Power consumption and precision may be adjusted dynamically, which is beneficial for artificial intelligence and other applications with changing needs. Introducing a customizable multiplier structure requires spending extra money on hardware. This article discusses a high-precision 4-2 compressor that serves as the basis for creating a high-precision approximation multiplier. Furthermore, we introduce a MAC that adapts precision and power by employing the dynamic input truncation technique.

2. LITERATURE SURVEY

"Comparing and extending approximate 4-2 compressors for low-power approximate multipliers." G. D. Meo, E. Napoli, A.G. M. Strollo, N. Petra, and D. DeCaro are the individuals in question. Recursive multipliers (RMs) are considered low-power multipliers due to the wide range of power-quality adjustment choices they provide. The basic structure of this recursive design uses 2×2 multipliers, however more advanced approximate recursive systems typically use 4×4 multipliers. Further research is needed to explore the design possibilities of AxRMs that include 2×2 multipliers. Compact, high-performance 2-bit multipliers are essential to improve the adaptability and configurability of AxRM systems. This article introduces two 2×2 multipliers with double-sided error distributions.

The suggested design outperforms the present best-approximated 2x2 multiplier by reducing area by 52% and improving latency by 25%, while still keeping restricted error behavior. Three 8x8 multipliers with different levels of precision are created by reorganizing an approximate 2x2 multiplier in different ways. AxRM1 stands out as the most precise design due to its 50% improvement in mean relative error distance (MRED) compared to the most effective MRED-optimized design currently in use. The MRED of AxRM3 closely resembles that of MACISH, the most efficient 2x2-based AxRM that preceded it. AxRM3 has a 13% greater Partially Dynamic Power (PDP) than MACISH because it builds larger multipliers using low-power, high-performance 2x2 multipliers. Convolutional neural networks exemplify sophisticated error-tolerant applications by employing the approximation multipliers currently under consideration. AxRM2 achieves an ideal equilibrium between power usage and quality, saving 32.64% energy and improving classification accuracy by 1.0%.

3.DESIGN OF MAC USING TRUNCATED MULTIPLIER



Fig. 1. Basic building block diagram (MAC)

MAC Function

MAC operation is vital for DSP, multimedia information processing, and various other applications. The MAC comprises a register/accumulator, multiplier, and adder. This paper used an approximate multiplier. The MAC inputs are retrieved from the memory address assigned to the multiplier block. This feature provides advantages to 16-bit DSPs. The input can be connected instead of using the 16-bit memory. The multiplier processes a 16-bit input and produces a 16-bit output after a successful calculation. The 16-bit data outputs from the multiplier are sent to an adder for addition

The MAC unit function is depicted by the following equation.

$$F = \sum P_i Q_i$$

The carry bit is appended to the highest output of 16 bits by the adder unit. There is a connection between the accumulator register and the related output data. The accumulator register functions

through the PIPO (parallel in parallel out) register method. Due to its Parallel-In Parallel-Out (PIPO) configuration, the data is extensive and the adder generates output values simultaneously. PIPO, or Parallel In Parallel Out, receives input bits simultaneously and generates output bits simultaneously. The output of the accumulator register transmits any input to a corresponding adder. Figure 1 illustrates the basic building block diagram of the MAC unit.

APPROXIMATE MULTIPLIER

Using approximate multipliers is highly recommended for energy-efficient computation in error-tolerant situations. Accuracy is an essential design consideration, along with power, area, and performance, making it challenging to select the most appropriate approximation multiplier. This article establishes three critical determinants that influence the selection of an approximation multiplier circuit. (4) Factors such as the architecture of the multiplier (array or tree), the organization of its efficient compressor sub-modules, and the type of compressor used in its creation. We explored the design possibilities for circuit-level implementations of approximate multipliers using these variables. Various common compressors were implemented at the circuit level.

PROPOSAL FOR A HIGH-ACCURACY 4-2 COMPRESSOR

This endeavor suggests creating a low-power, high-precision 4-2 compressor. Figure 2 depicts the proposed 4-2 approximation compressor. The next text describes the planned 4-2 approximation compressor setup. Generating W1 \rightarrow W4 necessitates four inputs: X1 \rightarrow X4. The carry bit in the proposed compressor is designed to be calculated accurately at all times because to its higher error distance compared to the sum bit. An improper carry bit results in double the error distance compared to an incorrect sum bit. Below are the equations for creating carrier bits. There are three scenarios where the carry bit is set to 1.

W1 = X1 AND X2 W2 = X1 OR X2 W3 = X3 AND X4 W4 = X3 OR X4 W5 = W1 OR W3 W6 = W2 AND W4 Carry = W5 OR W6



Fig 2. Implementation at the gate level of the proposed 4-2 compressor.

In order to make the sum bit, you should use this phrase. The 4-2 compressor uses two complete adders and four XOR gates to produce the sum bit rapidly. The signals that produce the carry bit to make the sum bit in the recommended compressor are obtained by running W2 and W4 through a two-input XOR gate.Reduced utilization of circuit space and static power can be achieved through the sharing of common symbols.

X ₃	X_3	X_2	X ₁	carry	sum	diff.
0	0	0	0	0	0	0
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	0	1	1	1	0	0
0	1	0	0	0	1	0
0	1	0	1	1	0	0
0	1	1	0	1	0	0
0	1	1	1	1	1	0
1	0	0	0	0	1	0
1	0	0	1	1	0	0
1	0	1	0	1	0	0
1	0	1	1	1	1	0
1	1	0	0	1	0	0
1	1	0	1	1	1	0
1	1	1	0	1	1	0
1	1	1	1	1	1	-1

Table 1. Truth table for our proposed 4-2 compressor design approximation.

The error distance is rather significant when using only W2 and W4 in a two-input XOR gate. Since W2 and W4 were constructed using OR gates, an error occurs when X1 and X2 or X3 and X4 are both set to 1. Despite its proper value being zero, the sum bit is currently set to one. The signal utilized to find these two conditions, W5, is sent through the XOR gate for extremely high accuracy. W2 and W5 will also be 1 if X1 and X2 are both 1. A value of "0 XOR W4" will be assigned to the sum bit, which will be known as W4. Only bits X3 and X4 require communication in this scenario. The error margin remains 1 when all four parameters are adjusted to 1. Determine the result of XORing W5, W2, and W4 with each other. Table 1 displays the truth table for the proposed approximate 4-2 compressor. When all four entries are identical, it is considered an error. The probability of a multiplicand bit and a multiplier bit being identical is $(1/2)^2$. This gives a 1/4 probability that the product will have one component. If all four inputs are 1, the probability is (1/4)4. No matter how incorrect the proper output is, there will always be exactly one difference between our output and it. The error is present in both W1 and W3. Verifying that W1 and W3 are both 1 requires an additional AND gate in order to detect errors. This is due to the fact that W1 and W3 check for the presence of 1 in X1 and X2 and X3 and X4 using AND gates. Creating the error-correction circuit for the proposed 4-2 compressor is as simple as adding an additional AND gate.



Fig 3. Modified partial product.

DYNAMIC INPUT TRUNCATION

We suggest using a dynamic input truncation technique with two 2-input AND gates to create a partial product for a configurable approximate multiplier at runtime. The equation for the partial product is based on the multiplicand (A) and multiplier (B). The Trunc signal determines if the partial product PPD is shortened. The partial product is reduced to zero when the Trunc value equals 1. Trunc signals save energy by setting the Probability of Partial Decisions (PPDs) to zero when performing multiplications. In other words, the Trunc signals can be seen as a way to disable the hardware components in their respective columns. Aj equals the product of PPDij and the conjunction of \preccurlyeq Bi and Trunc. To minimize hardware costs for an 8x8 multiplier, it is recommended to utilize shared gates together with an extra AND gate, as each bit in the multiplier represents eight bits. PPD01 is equivalent to $Trunc0 \cdot B0 \cdot A1$, while PPD00 is equivalent to $Trunc0 \cdot B0 \cdot A0$. Figure 4 illustrates the need for three AND gates, each with two inputs. Trunc0B0, a disguise, can be premeditated. The following part will offer a detailed analysis of the regulation of Trunc signals in the suggested approximate multiplier.



Fig 4. An example of gate sharing to decrease the quantity of gates.



Fig 5. Proposed estimated multiplier.

APPROXIMATE MULTIPLIER PROPOSED

Figure 5 demonstrates an estimated multiplier calculated using the suggested methods. Although the multiplier's input breadth is defined as 8 bits, the suggested methodology can be expanded to handle bigger quantities. The suggested multiplier consists of three separate steps. Figure 3 shows that the first step is to create each partial product using two 2-input AND gates. Figure 4 illustrates the gate sharing technique, which helps to decrease hardware costs. The accuracy of the partial product can be evaluated using the Trunc signal based on the unique requirements. Our suggested approximate multiplier includes a 4-bit Trunc signal to enhance control effectiveness and save hardware costs. The signal referred to as the "3-4-4-4 partition" is named based on how each bit from the most significant bit to the least significant bit corresponds to specific partial product columns, which are color-coded as khaki, sky blue, green, and black in Stage 2 of FIGURE 4. The 14th to 12th bits, 11th to 8th bits, 7th to 4th bits, and 3rd to 0th bits, respectively. If Trunc(0) is 01012, trim columns 11, 8, and 30, and avoid columns 14, 12, and 28.

4.RESULTS

RTL SCHEMATIC: The RTL schematic acts as a prototype for the architecture and is used to compare the proposed design with the ideal architecture that has not been developed. Verilog or VHDL is used to create a functional abstraction from the architecture's description. The RTL diagram aids in

the further investigation of the internal connection blocks. The diagram below shows the RTL schematic representation of the planned architecture.



Fig6. RTL Schematic of MAC using approximate multiplier

TECHNOLOGY SCHEMATIC : It shows the architecture in LUT format, where the LUT serves as the parameter used by VLSI to estimate the area in the architecture design. In FPGA, code memory allocation is represented by LUTs, which are square units.



Fig 7. View Technology schematic of MAC using approximate multiplier

SIMULATION: While each block and link is double-checked in the schematic, and the simulation final check to guarantee that everything works as expected. To open the simulation window on the tool's main screen, switch from implantation to simulation. The simulation window limits the output to wave patterns. It is versatile enough to support multiple radix number systems.



Fig8.Simaulation wave forms of MAC using approximate multiplier

PARAMETERS: VLSI considers three parameters: area, delay, and power. These measurements are suitable for comparing different architectural designs. Furthermore, area capacity and latency are taken into account. The parameter extraction is carried out using XILINX 14.7 with Verilog as the Hardware Description Language (HDL).

Summary of MAC device usage using an approximation multiplier.

Device Wilcolm Summary (editorial values)						
Logic Williandion	lted	kalabe	Utistic			
Norther of Sila Registers	Ĩ	50	ħ			
Nober of Size Life	j6	60	5			
Noter tildy und 10747 part	Ĩ	16	15			
Norter of Iconied 10%	1	30	8			
luter of BPGB/PBCRs	1	I	Ъ			

Minimum period: 2.179ns (Maximum Frequency: 458.926MHz) Minimum input arrival time before clock: 8.410ns Maximum output required time after clock: 0.735ns

CONCLUSION

This study utilizes a new approach that combines approximate 4:2 compressor designs with an estimated multiplier to create MAC units. The paper describes an accurate 4-2 compressor and an adjustable approximation multiplier that may truncate partial products dynamically to match different accuracy needs. Furthermore, a multiplier and accumulation (MAC) unit is suggested. The suggested MAC, equipped with an estimated multiplier, can adjust power and precision levels required for multiplications during runtime based on user-defined criteria. Ultimately, creating an approximation multiplier that offers clear advantages is challenging, and the best option is usually the one that best fits the specific need. The architecture of our approximation multiplier offers a rival a choice between competitive error and electrical performance trade-offs.

REFERENCES

[1] A. Bosio, D. Ménard, and O. Sentieys, Eds. Approximate Computing Techniques: From Component-to Application-Level. Cham, Switzerland: Springer, 2022. [Online]. Available: https://link.springer.com/book/10. 1007/978-3-030-94705-7

[2] A. G. M. Strollo, E. Napoli, D. De Caro, N. Petra, and G. D. Meo, "Comparison and extension of approximate 4-2 compressors for lowpower approximate multipliers," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 67, no. 9, pp. 3021–3034, Sep. 2020.

[3] T. Kong and S. Li, "Design and analysis of approximate 4-2 compressors for high-accuracy multipliers," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 29, no. 10, pp. 1771–1781, Oct. 2021.

[4] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "Design and analysis of approximate compressors for multiplication," IEEE Trans. Comput., vol. 64, no. 4, pp. 984–994, Apr. 2015.

[5] F. Sabetzadeh, M. H. Moaiyeri, and M. Ahmadinejad, "A majority-based imprecise multiplier for ultra-efficient approximate image multiplication," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 66, no. 11, pp. 4200–4208, Nov. 2019.

[6] H. Pei, X. Yi, H. Zhou, and Y. He, "Design of ultra-low power consumption approximate 4-2 compressors based on the compensation characteristic," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 68, no. 1, pp. 461–465, Jan. 2021.

[7] D. Esposito, A. G. M. Strollo, E. Napoli, D. de Caro, and N. Petra, "Approximate multipliers based on new approximate compressors," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 65, no. 12, pp. 4169–4182, Dec. 2018.

[8] U. Anil Kumar, S. K. Chatterjee, and S. E. Ahmed, "Lowpower compressor-based approximate multipliers with error correcting module," IEEE Embdded Syst. Lett., vol. 14, no. 2, pp. 59–62, Jun. 2022.

[9] X. Yi, H. Pei, Z. Zhang, H. Zhou, and Y. He, "Design of an energyefficient approximate compressor for error-resilient multiplications," in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), 2019, pp. 1–5.

[10] M. Ha and S. Lee, "Multipliers with approximate 4-2 compressors and error recovery modules," IEEE Embdded Syst. Lett., vol. 10, no. 1, pp. 6–9, Mar. 2018.