Designing a Mac Unit Using Approximate Multiplier

*

K.Prathyusha
Dept. of ECE
Gokaraju Rangaraju Institute of Engineering and Technologly(GRIET)
Hyderabad,Telangana.
prathyushakotte8@gmail.com

M.Balaraju
Dept. of ECE
Gokaraju Rangaraju Institute of Engineering and Technologly(GRIET)
Hyderabad,Telangana.
balrajumuntha@gmail.com

B.Akash
Dept. of ECE
Gokaraju Rangaraju Institute of Engineering and Technologly(GRIET)
Hyderabad,Telangana.
akashc15042002@gmail.com

K.Jamal
Dept. of ECE
Gokaraju Rangaraju Institute of Engineering and Technologly(GRIET)
Hyderabad,Telangana.
kjamal24@gmail.com

Manchalla.O.V.P.Kumar
Dept. of ECE
Gokaraju Rangaraju Institute of Engineering and Technologly(GRIET)
Hyderabad,Telangana.
pavanomkar@gmail.com

M.Suneetha
Dept. of ECE
Gokaraju Rangaraju Institute of Engineering and Technologly(GRIET)
Hyderabad,Telangana.
Suneetha645@gmail.com

***Abstract*—In numerous applications, multipliers play a crucial role as arithmetic functional units, often requiring extensive multiplications that contribute significantly to power consumption. Employing an approximate multiplier represents a novel strategy to reduce critical path time and power usage in error-tolerant systems. The trade-off involves sacrificing accuracy for enhanced performance and reduced energy consumption. To cater to varying precision requirements, this article not only introduces a highly accurate approximate 4-2 compressor but also proposes a adaptable approximation multiplier capable of dynamically truncating partial products. Additionally, a Multiplier and Accumulation (MAC) unit is recommended. Depending on user needs, the suggested MAC unit, paired with an approximate multiplier, can dynamically adjust power and accuracy for multiplications during runtime.**

***Index Terms*—Approximate multiplier,ripple carry adder,Accumulator,mac design**

I. Introduction

In a number of fields, including artificial intelligence, computer vision, multimedia processing, image recognition, and digital signal processing (DSP), multipliers play a crucial role as essential arithmetic functional units. These applications often demand a significant number of multiplications, leading to substantial power consumption, especially in mobile devices, presenting a challenge for implementation. Many studies have suggested ways to lower multiplier circuit power consumption in order to overcome this problem. When error tolerance is permissible or when applications are linked to human senses, one approach to mitigating multiplier power consumption is through the approximation of multiplication. Given the limited sensory capabilities of humans, precise computational outcomes are not always necessary, enabling the approximation-based decrease of cell space, time delay, and power consumption. Approximate multipliers come in two primary varieties: those utilizing dynamic voltage scaling to regulate the multiplier's time path and those that involve reworking specific multiplier circuits, such as the wallace Tree Multiplier and dadda Tree Multiplier, to alter operational properties. Previous research on rebuilding multipliers frequently suggested erroneous m:n compressors, in where m stands for inputs and n for outputs. During multiplication, these compressors were used to compress partial products, which resulted in high energy consumption and considerable path latency. The majority of earlier approximation multipliers had set output accuracy and power requirements. However, there is a need for dynamic adjustments in accuracy and power consumption for applications like artificial intelligence, whose requirements evolve over time. It's essential to note that achieving an adjustable multiplier architecture incurs additional hardware costs. In order to create a high-accuracy approximation multiplier, this work presents a high accuracy 4:2 compressor. Furthermore, a dynamic input truncation technique is introduced to modify the needed power and precision as necessary. The paper makes the following key contributions:

1.Proposing the construction of the recommended approximate multiplier, which is based on a high-precision approximate 4-2 compressor.

2. Presenting a straightforward circuit for error compensation intended to further minimize error lengths.

3.Presenting a dynamic input truncation method that allows for the adjustment of power and precision requirements in multiplication.

4.Developing a Multiply and Accumulate (MAC) unit that incorporates an approximate multiplier in this project.

MAC'S DESIGN

The Multiplier-accumulator (MAC) device plays a crucial role in supporting various digital signal processing (DSP) applications. It empowers the microcontroller to handle signal processing tasks, including servo and audio control. Positioned as an execution unit within the CPU, the MAC device employs a 3-stage pipelined arithmetic architecture to optimize 8x8 multipliers. It is capable of processing both signed, fixed-point fractional input operands, and signed and unsigned integers.

The MAC unit serves three primary functions:

1. Signed and unsigned integer multiplication.

2.Perform multiply-accumulate operations,supporting signed,unsigned and signed fractional operands.

3.Execute other register functions, comprising an accumulator adder, multiplier, and adder within the MAC unit.

Given the high-speed requirements of DSP applications, adders are predominantly used in the MAC unit. To enable more multiply-accumulate operations, the memory collects inputs from its location to the multiplier.The output of the MAC unit is kept in a pertinent memory location. and the entire process must be completed within a single clock cycle due to specific constraints.

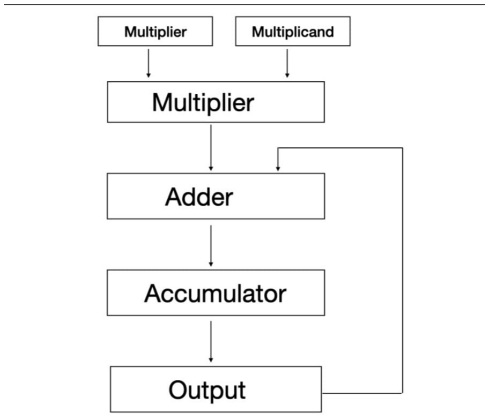


Fig. 1. Basic MAC.

II. LITERATURE SURVEY

This research introduces two novel approximate 2×2 multipliers possessing a double-sided error distribution characteristic. In contrast to the most accurate 2x2 multiplier currently in use, these designs demonstrate a remarkable 52 percent reduction in area and a significant 25 percent increase in delay, while maintaining a bounded error behavior. Additionally, the study involves the development of three 8x8 multipliers with different levels of accuracy using various configurations of the approximate 2x2 multiplier. With a Mean Relative Error Distance (MRED) improvement of 50 percent over the previous best MRED-optimized design, AxRM1 emerges as the most accurate design. On the other hand, AxRM3's MRED performance is comparable to that of the top 2x2-based AxRM. but with a notable 13 percent increase in power-delay.[1] An emerging trend in digital design is approximate computing, which puts increased speed and power efficiency ahead of precise processing requirements. In order to create effective approximation multipliers, this paper presents novel approximate compressors along with an algorithm that makes use of them. Using a 40-nanometer library, the research team has created approximation multipliers for a range of operand lengths through the use of this suggested methodology. It is clear from a comparative study with previously proposed approximated multipliers that the suggested circuits function faster or with more power for a certain precision target. The paper also presents the use of these approximation multipliers in adaptive least mean squares filtering ,CNN and picture filtering.[2] The numerous approximate 4-2 compressors that have been put forth in the literature to date are thoroughly surveyed and compared in this research. Additionally, a novel approximate compressor is introduced, resulting in the analysis of a total of twelve distinct approximate 4-2 compressors. These investigated circuits are utilized in the design of 8x8 and 16x16 multipliers, implemented in 28nm CMOS technology. Two multiplier configurations, both signed and unsigned, with different degrees of approximation are analyzed for every operand size. The study reveals that there is no singular optimal approximate compressor topology, as the most suitable solution depends on factors such as the required precision, the signedness of the multiplier, and the specific error metric considered.[3]This paper presents an approximation circuit that was created by keeping the operation intact while altering the circuit layout. Using Wallace tree reduction, 3:2 inexact additive designs for partial product synthesis and addition, as well as AND-OR logic approximation, the suggested method entails generating an approximate multiplier. The paper illustrates the suggested idea with a case study of an 8x8 bit multiplication. Moreover, it is demonstrated that the suggested multipliers result in notable improvements concerning both latency and area use. [4].According to the testing results, our designs have an approximate 18percent delay, a (43–52) percent decrease in area-delay product (ADP) when compared to the precise multiplier, and an ADP optimization of (20–55) percent when compared to compressors with identical precision. This article uses applications for image blending and matrix multiplication to further confirm the effectiveness of the suggested compressors.[5] The scientific literature demonstrates a great degree of interest in approximation multipliers and offers numerous circuits built employing approximate 4-2 compressors. With so many options available, the designer wanting to use an approximation 4-2 compressor finds it challenging to select the right architecture.[6]Compared to conventional ripple-carrier adders, the 4:2 CSA module operates by effectively adding four binary inputs while decreasing the amount of carry propagations, which lowers total latency and power consumption. The module is divided into multiple stages, each of which consists of a mix of various logic components and full-adders.[7]Approximate multiplication plays a vital role in

approximation computing algorithms, aiming to achieve both low power consumption and efficient performance. Utilizing approximate 4-2 compressors can facilitate the creation of circuits for approximative multiplication that use less power. In this letter, an error recovery module is included into a creative design that improves on a previous approximation 4-2 compressor design. thus improving its performance. Compared to previously published approximation multiplier designs relying on 4-2 compressors, the proposed architecture exhibits superior accuracy, reduced power consumption, and diminished hardware resource requirements, despite the incorporation of the error recovery module.[8]The proposed approach leverages innovative dual-stage 4:2 compressors to achieve efficient approximate multiplication. These dual-stage compressors offer enhanced performance and flexibility compared to traditional single-stage compressors. The study explores the design, implementation, and evaluation of the approximate multiplier, highlighting its advantages in terms of Area, Power consumption, and performance analysed with existing approaches.[9]The creation of circuit components that may dynamically modify their operating voltage levels in response to the required degree of approximation is referred to as the design of meta-functions that are Voltage-scalable for approximative computation. When trading accuracy for gains in performance, energy efficiency, or other metrics, these meta-functions come in handy in approximation computing.[10] Using FSM and Verilog coding techniques to implement a MAC transmitter aids in MAC design by offering a methodical way to specify the transmitter's behavior, guaranteeing adherence to protocol specifications, and enabling modular design and testing. Complex control logic can be modeled with the aid of FSMs, and hardware-level implementation can be achieved for effective and scalable designs using Verilog.[11]Describe ALUs and Booth multipliers in general terms, emphasizing their functions in digital systems and the importance of their ability to carry out arithmetic and logic operations quickly. Talk about how speed and efficiency are crucial for MAC design, emphasizing that high-performance parts like multipliers and ALUs are required to satisfy the needs of data processing and transmission.[12]

III. Design Of MAC Using Truncated Multiplier

A MAC unit is a common component found in numerous digital signal processing systems. that use accumulation and multiplication. It also operates digital DSP systems with high speed. DSP is utilized in a variety of procedures, including convolution, filtering, and inner products. DSP techniques often utilize nonlinear functions such as Discrete wavelet transforms (DWT) and Discrete cosine transforms (DCT). The whole speed of the arithmetic computations, which includes addition and multiplication, is established by the speed of execution and the performance of the full calculation since addition and multiplication are efficiently finished by cyclic application of addition and multiplication. Procedures that multiply and aggregate are unique to digital filters.Next, high-speed filtering and other special processing units for DSP applications are made possible by the MAC unit's basic features. To reduce the stress on the CPU, the MAC unit processes each piece of data individually, operating entirely independently of the CPU.One of the best uses is the optical communication system, which is entirely dependent on the DSP and needs a large amount of data to process digital data quickly. For the Fast Fourier Transform, the addition and multiplication operations are also necessary (FFT). The 16-bit MAC unit requires additional memory because it can handle a lot of bits. Its basic components are an accumulator unit that holds the total of the preceding sequential product phrases and a multiplier. The matching memory address that is attached to the multiplier block provides the inputs for the MAC unit.

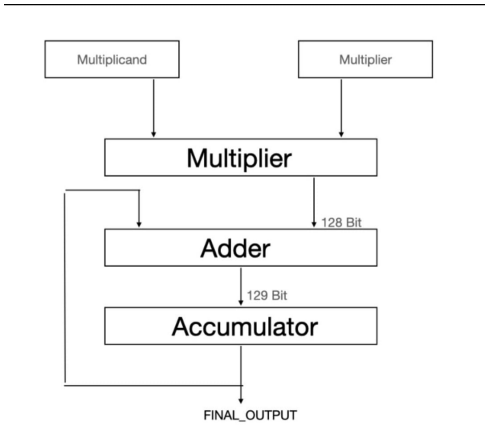


Fig. 2. MAC - Fund.Block Diagram.

A. MAC OPERATION

Furthermore, the Multiply-Accumulate (MAC) operation is important in a variety of applications related to multimedia information processing and other domains, outside of its relevance in digital signal processing (DSP). As was previously mentioned, the register/accumulator, multiplier, and adder comprise the MAC.In this work, a Vedic multiplier was used. The matching memory address that is attached to the multiplier block is where the MAC inputs are obtained. This is beneficial for DSP that is 16-bit.We can connect the input from the position of the 16-bit memory. After processing the 16-bit input, the multiplier produces a 16-bit output when the computation is successful. These 16-bit multiplier outputs are then fed into an adder to perform additional addition operations.

An extra bit is added for carry to the 16-bit output that the adder unit generates. The pertinent data in the accumulator register is then linked to this output. A Parallel in parallel out (PIPO) register method is used by the accumulator register, where input bits are received and output bits are generated simultaneously in parallel. Large sets of bits can be handled well using this PIPO technique, making it easier to generate adder output values in parallel. The output of the accumulator

register receives inputs to equivalent adders. The MAC unit's fundamental block diagram is shown in Figure 2.

B. APPROXIMATE MULTIPLIER

Approximate multipliers are heavily recommended for energy-efficient computing in areas where error is naturally tolerated. But in addition to performance, area, and power, accuracy is a crucial design parameter, which makes choosing the best approximation multiplier difficult. In this study, we delineate three primary determinants that influence the choice of an approximation multipliers circuit: (1) The kind of dual quality and approximate area efficient compressor that was utilized to build the multiplier; (2) The multiplier's design, which can be either an array or a tree; and (3) The arrangement of the approximate and exact multiplier sub-modules within the main multiplier module.We investigated the design space for circuit level implementations of approximate multipliers based on these variables. We used some of the most popular compressors implemented at the circuit level.

C. PROPOSED TRUNCATED APPROXIMATE MULTIPLIER

This proposal proposes an extremely precise and Low-Power approximation 4-2 Compressor. The Proposed compressor for 4-2 approximation as shown in Fig 3. This is a description of the design of the proposed 4-2 approximation compressor. Using four inputs, x1 x4, w1 w4 is generated. The proposed compressor's carry bit is designed to be created accurately at all times since a wrongly computed carry bit produces twice the error distance (ED) of a wrongly computed sum bit. This is because an inaccurate Carry Bit has a bigger error distance compare to Sum Bit. The equations for carry bit creation are shown below. The carry bit turns into one in three different scenarios. x1 and x2 are equivalent.Another is that x4 and x3 are equal to 1.One of (x1+x2), and one of (x3 + x4) are both 1. This is the third. (5) confirms the first two scenarios, while (6) confirms the third. (7) produces the final carry bit.

$$w1 = \ x1ANDx2 \tag{1}$$

$$w2 = \ x1ORx2 \tag{2}$$

$$w3 = \ x3ANDx4 \tag{3}$$

$$w4 = \ x3ORx4 \tag{4}$$

$$w5 = \ w1ORw3 \tag{5}$$

$$w6 = \ w2ANDw4 \tag{6}$$

$$Carry = \ w5ORw6 \tag{7}$$

Below is the suggested equation for creating the sum bit. The sum bit in a precise 4-2 compressor is generated by combining four XOR gates in the two complete adders. In our suggested compressor, on the other hand, we generate the carry bit and the sum bit using a 2-input XOR gate with the inputs w2 and w4. We can reduce the amount of circuit space and static power usage by utilizing common signals. But we found that a 2 i/p XOR gate with only w2 and w4 produces a sizable

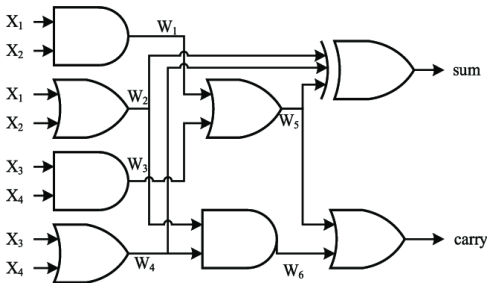


Fig. 3. Implementation of 4:2 compressor.

error margin.When either (x1 and x2) or (x3 and x4) are both 1, mistakes occur because of the OR gates that are utilized to create w2 and w4.Consequently, even though the sum bit ought to be zero, it is set to one. In order to attain extreme precision, we input the signal w5,It was fed into the XOR gate in order to detect these two circumstances. Thus, w2 and w5 will both be 1 if x1 and x2 are both 1. Consequently, the sum bit will be w4, and its value will be "0 XOR w4". Only x3 and x4 are bits that need to be considered in this case.

$$sum = \ w2XORw4XORw5 \tag{8}$$

Our proposed approximate compressor's truth table (TABLE 1) reveals errors only when all four inputs are ones. Taking into account the likelihood of a bit in the Multiplicand and Multiplier being 1 as (1/4), the likelihood of a partial product being 1 is also 1/4. Thus, the chance of all four inputs being 1 is merely (1/256). Moreover, even if an error does occur, there's only a single discrepancy between our output and the accurate output, minimizing the impact.

$$Error = \ w1ANDw3 \tag{9}$$

Since w1 and w3 both use AND gates to verify whether x1 and x2 and x3 and x4 are 1, respectively, adding one additional AND gate is sufficient for error detection to check whether both w1 and w3 are 1. Equation (9), which represents the error detection circuit,is given. Thus, incorporating an extra AND gate simplifies the planned 4-2 compressor's error compensating circuit design.

D. DYNAMIC INPUT TRUNCATION

Through the employment of two 2-input AND gates and a dynamic input truncation approach, a customizable approximate multiplier may be achieved at runtime. to generate a partial product, the equation for which is displayed. where the multiplier is B and the multiplicand is A. To decide whether to truncate the partial product, one uses the Trunc signal.The partial product is trimmed to zero if the trunc value is 1. Put another way, the Trunc signals preserve energy by setting the PPDs in the multiplications to zeros. Put differently, we can consider the Trunc signals to be acting as a means of turning off the hardware units in the respective columns.

$$PPDij = \ (\ TruncANDBi)ANDAj \tag{10}$$

Our solution is to incorporate an additional AND gate to

X4	X3	X2	X1	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

Fig. 4. Table-1 Truth Table for proposed approximate multiplier.

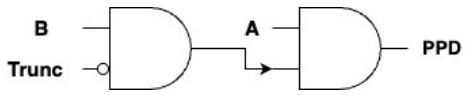


Fig. 5. Modified partial product

share gates and lower hardware costs because every bit in an 8 × 8 multiplier corresponds to 8 bits in the multiplicand. For instance, ppd01 and ppd00 are computed as trunc0·B0·A1 and trunc0·B0·A0, respectively.In this scenario, three 2-input AND gates are required, and trunc0·B0 can be precomputed to serve as a mask.

E. Ripple Carry Adder

A Ripple Carry Adder (RCA) is an essential building block in digital circuits employed in computer processors and similar digital systems for binary number addition. Its name, "ripple carry," stems from the way the carry bit, produced by adding each pair of bits, sequentially propagates through successive stages of the adder. The truth table of a complete adder, the fundamental building block of a ripple carry adder (RCA), can be used to obtain the Sum (S) and Carry-out (Cout) equations for a single stage of the RCA. For a single full adder stage, the sum (S) and carry-out (Cout) can be expressed as follows:

Sum = A xor B xor Cin (11)

Cout = (AB)+(ACin)+(BCin) (12)

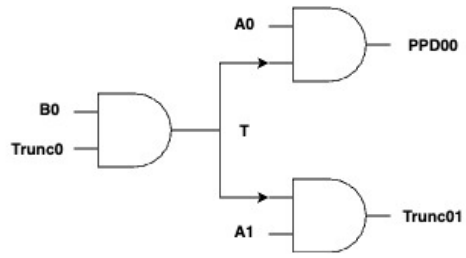


Fig. 6. An example of gate sharing to minimize the number of gates.

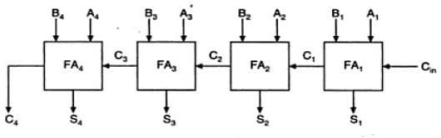


Fig. 7. ripple carry adder

The fundamental logic processes needed to calculate the sum and carry out for a single stage of a ripple carry adder are represented by these equations. These equations are applied iteratively to every step in a multi-bit RCA, where the carry-in of one stage is influenced by the carry-out of the previous one.

F. Register

A register in digital electronics is a type of data storage element used to hold binary information temporarily. It's commonly implemented using flip-flops or other similar circuitry. Registers play a crucial role in digital systems, such as microprocessors, where they are utilized for various purposes including data storage, data movement, and control signal generation. The "D" register, sometimes referred to as a "data register," is an essential part of microprocessor designs and digital systems. Its main purpose is to store binary data temporarily.It contains binary data that is being moved, processed, or altered across several system components.The D register can hold operands, intermediate results, or calculation final outcomes during arithmetic and logic operations. It offers a workspace for carrying out multiplication, division, addition, and subtraction operations.It facilitates the processing of several instructions at once by segmenting the instruction execution process into sequential steps.In digital systems and microprocessor designs, the D register is essential for enabling data processing, storage, and transfer. Because of its adaptability and flexibility, It is an essential part of the design of a wide range of digital systems and circuits.

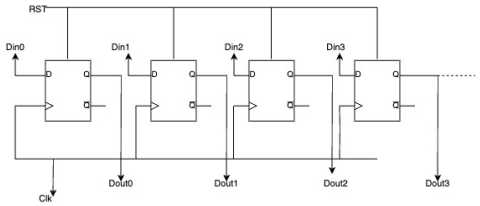


Fig. 8. Circuit diagram of register

IV. RESULTS

A. RTL Schematic

The architecture's blueprint, known as the register transfer level schematic, or RTL schematic for short, is used to contrast the current, intended design with the ideal architecture that still needs to be created.The working summary is created by utilizing the hdl language to transfer the description or summary of the architecture into a coding language, such as

verilog or vhdl. The RTL diagram even specifies the internal connection blocks for further analysis.The graphic below shows the planned architecture's RTL schematic diagram.

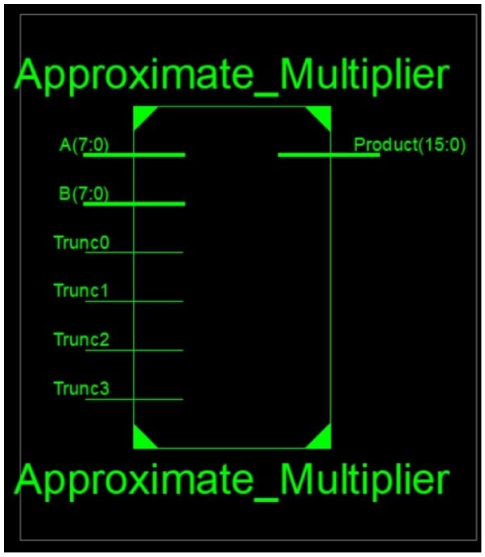


Fig. 9. RTL Schematic of Approximate Multiplier.

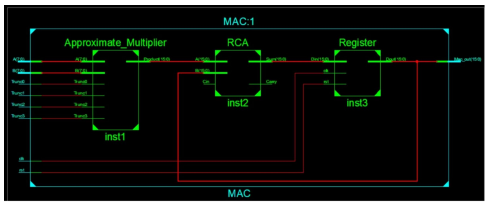


Fig. 10. RTL Analysis of MAC Using Approximate Multiplier.

B. Technology Schematic

The architectural layout is depicted in the technology diagram using the LUT (Look-Up Table) format, where LUTs serve as a measure or parameter in VLSI (Very Large Scale Integration) to gauge the design of the architecture. These FPGA LUTs, often referred to as squarunits, illustrate the memory allocation of the code.The LUT technology schematic serves as a reference for identifying potential issues or errors in the logic implementation. Designers can trace signal paths, verify connections, and diagnose any anomalies that may arise during operation.

C. Simulation

The simulation is the procedure that is thought of as the final check to make sure that everything works as planned, while the schematic confirms the connections and blocks as shown in figure 13 and figure 14. To access the simulation window, navigate from implantation to simulation on the tool's home screen. Wave patterns represent the output that is limited by the simulation window.This is where its flexibility in offering many radix number systems comes in handy. Additionally, simulations offer insights into system behavior

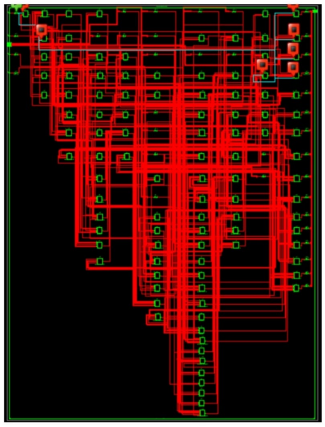


Fig. 11. Technology Schematic of Approximate Multiplier.

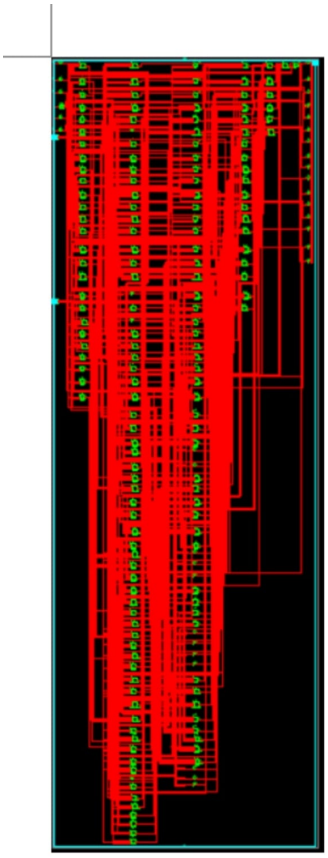


Fig. 12. Technology Schematic of MAC Using Approximate Multiplier.

under different conditions, enabling thorough validation before actual deployment.

Fig. 13. Simulation waveforms of approximate multiplier.

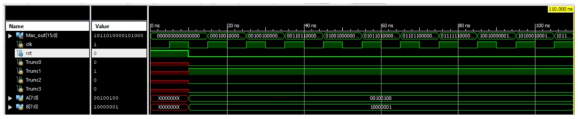
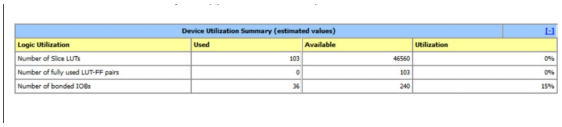


Fig. 14. Simulation waveforms of approximate multiplier.

D. Parameters

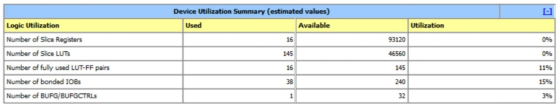
In VLSI, area, delay, and power are taken into account. These variables can be used to compare various architectures. Here, the value that considers both area power and latency is derived using the tool XILINX 14.7. Verilog is the HDL language utilized.

Approximate Multiplier and MAC Device Utilization Summary Using Approximate Multiplier



Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	103	46560	0%
Number of fully used LUT-FF pairs	0	103	0%
Number of bonded IOBs	36	240	15%

Fig. 15. Approximate multiplier.



Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	16	93120	0%
Number of Slice LUTs	145	46560	0%
Number of fully used LUT-FF pairs	16	145	11%
Number of bonded IOBs	38	240	15%
Number of BUFG/BUFGCTRLs	1	32	3%

Fig. 16. MAC Using approximate multiplier.

Delay:7.673ns
Min Period:2.179ns
(Max Freq:458.926MHz)
Arrival time of minimum input before clock::8.410ns
Maximum output required time after clock:0.735ns

CONCLUSION

This study introduces a new method involving approximate 4:2 compressor designs coupled with an approximate multiplier for constructing MAC (Multiplier-Accumulator) units. The aim is to cater to different precision requirements. We describe an extremely accurate Approximate 4-2 Compressor and suggest a flexible approximation multiplier that may dynamically truncate partial products. Additionally, we propose a MAC unit that can adjust power consumption and accuracy levels for multiplication tasks based on user requirements during runtime.In essence, designing an approximation multiplier with absolute advantage is quite difficult, and the best solution is usually the one that best fits the intended use. We provide a contender with a competitive error-electrical performance tradeoff in our approximate multiplier design.

REFERENCES

[1] O. Sentieys, D. Me nard, and A. Bosio, eds. Techniques for Approximate Computing: From the Component to Application Level. Springer, Cham, Switzerland, 2022. [Online]. The following URL is available: https://link.springer.com/book/10. 1007/978-3-030-94705-7

[2] A flexible approximate multiplier designed for use in quantized Convolutional Neural Network (CNN) applications is presented by C. Guo, L. Zhang, X. Zhou, W. Qian, and C. Zhuo in the Proceedings of the 25th Asia South Pacific Design Autom. Conf. (ASP-DAC), January 2020, pages 235–240.

[3] Low-power high-speed multiplier for error-tolerant application, K. Yin Kyaw, W. Ling Goh, and K. Seng Yeo, Proc. IEEE Int. Conf. Electron Devices Solid-State Circuits (EDSSC), Dec. 2010, pp. 1-4.

[4] "A truncated multiplier with programmable functionality and low power consumption," by M. de la Guia Solaz, W. Han, and R. Conway, IEEE Trans. Circuits Syst., vol. 59, no. 11, pp. 2555–2568, Nov. 2012.

[5] Manohar, P. S., Rohan, B., Jamal, K., Kumar, M. O., Reddy, B. V. (2023, May). Ramana, P. V. S. Using a 2-bit approximate adder, a Carry Look Ahead adder is implemented. Applied Artificial Intelligence and Computing, 2nd International Conference (ICAAIC), 2023, pp. 1543–1547. IEEE.

[6] In December of 2021, Kumar, U. A., Ahmed, S. E., Jamal, K., Kumar, S., and Chintakunta, R. K. An approximation of multiplier architectures for applications that are error resilient. Pages 89–92 in the IEEE International Symposium on Smart Electronic Systems (iSES) 2021. IEEE.

[7] "4:2 carry-save adder module," A. Weinberger IEEE Technology Development Bulletin, vol. 23, no. 8, pp. 3811–3814, 1981.

[8] "Estimated multiplier with high precision and error correction," authored by C.-H. Lin and I.-C. Lin, was published in the IEEE 31st International Conference on Computer Design (ICCD) Proceedings, October 2013, pages 33–38.

[9] "Development of approximate multiplier utilizing innovative dual-stage 4:2 compression techniques." P. J. Edavoor, S. Raveendran, and A. D. Rahulkar, IEEE Access, vol. 8, pp. 48337–48351, 2020.

[10] "Design of voltage-scalable meta-functions for approximate computing," by D. Mohapatra, V. K. Chippa, A. Raghunathan, and K. Roy, was published in Proc. Design, Autom. Test Eur., Mar. 2011, pp. 1–6.

[11] Jamal, K., and Nayeema, S. (2013). Using FSM and Verilog coding techniques, a MAC transmitter is designed and implemented to transmit UDP packets. 338–342, Int. J. Eng. Res. Appl., 3(1).

[12] In October 2020, Supritha, B., Mannem, K., Reddy, B. V., Jamal, K., Kumar, M. O. ALU with a customised booth multiplier that operates quickly and effectively. IoT in Social, Mobile, Analytics, and Cloud: Fourth International Conference on I-SMAC (2020) (pp. 1196-1200). IEEE.