IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

Design And Analysis Of Approximate Multiplier With Reconfigurable Reduction

Mrs.V.Sreelakshmi[1],G.Siva Prasad[2],B.Divya[3],Ch.N.B.Bhavani[4],B.Pratap Reddy[5]

[1]Assistant Professor,[2345]Students

[12345]Dept of ECE, Seshadri Rao Gudlavalleru Engineering College, Seshadri Rao Knowledge Village, Gudlavalleru-521356, Andhra Pradesh, India

***Abstract*— For a variety of different applications, including processing of image, expert systems, the smart things, and processing of digital systems, low-power and high-speed electronic system are now required. In a general multiplying operation account for most of the data processing time. The potential solution for hardware that operates quickly and efficiently is approximate multiplier. The approximate multiplier has developed into a key arithmetic element for many applications over the past ten years. So, Identifying a suitable multiplier which makes the trade-off between power and efficiency is important. In this Paper, We compared different Approximate Multipliers and find the suitable one.**

Keywords—Approximate Multiplier, Low Power, Image processing, Neural Networks, Internet of things.

I. INTRODUCTION

In numerous applications, including processing of digital signals, machine vision, processing of multimedia, recognition of image, and artificial intelligence, multipliers are among the most important arithmetic functional units. These applications frequently require several multiplications, which consume a lot of power. Implementing various applications is difficult due to their high usage of electricity, especially on mobile devices. Because of this, numerous studies have suggested methods to lower the usage of electricity of multiplier circuits. When the intended uses support the tolerance for error, or, to put it another way, Whether they pertain to the senses of humans, approximating multiplication is one way to lower the usage of electricity of a multiplier. Precise computer results are not required because of limitations of human perception, such as a narrow sight restricted hearing span and scale. Precision is sacrificed with approximate multipliers in order to reduce usage of electricity, time delay, and cell area.

There are two types of approximate multipliers. With the use of reconfigurable voltage scaling, the reset type can regulate the multiplier's time path. The critical route will be delayed if a multiplier is given a lower voltage. As a result, when the time path is violated, errors occurs and approximative results are produced. The second type involves redesigning the precise multiplier circuits, such as the Wallace and Dadda Tree Multiplier, in order to change the functional characteristics of multipliers. Most of the earlier research on rebuilding multipliers suggested erroneous m to n compressors, which produce n outputs from m inputs. Since the process of compressing partial products absorbed the majority of the multiplier energy and resulted in a considerable route delay, these incorrect Compressors were utilised to minimise the incomplete products during multiplication.

In this paper, we present a high precision 4 - 2 compressor, and then develop a high precision approximate multiplier based on it. We also provide a reconfigurable input Reduction method to change the precision and usage of electricity needed. A list of the paper's contributions is provided below.:

- To construct the suggested approximation multiplier and the additional approximations, we suggest a high precison approximate 4 : 2 compressor.
- For all other approximative multipliers, we suggest using a reconfigurable input Reduction approach to adapt the precision and power needed for a multiplication.
- Finally, Different Comparisons are made with different approximate compressors.

According to experimental findings, as compared to the Wallace Tree Multiplier, the suggested alterable approximate multiplier with reconfigurable input Reduction can minimize the latency and average usage of electricity and other approximate results also shown as effective when there is a usage of reconfigurable Reduction. The other approximate

compressors are named as compressor1, compressor2, compressor3, compressor4, compressor5.

The remainder of this essay is structured as follows: Further approximative multipliers are introduced in Part II, as well as the comparison metric. In Part III, we present an approximation of the multiplier. In Section IV, we compare power, area, precision, delay, and area. The paper is completed in Section V.

II. FUNDAMENTALS

A. CONCEPTS FOR APPROXIMATE MULTIPLIER

Compressor:1 (suggested), Compressor:2, Compressor:3, Compressor:4, and Compressor:5 are the approximate multiplier designs. This section mostly focuses on the various approximative multiplier designs. Compressor:2 is made up of many logic gates, including two xnor gates and four nand gates. The 4 - 2 compressors listed above are all featured in this publication. Three nand gates, two xor gates, and three or gates make up Compressor:3. Only two xnor gates and two nand gates make up Compressor:4. In addition, Compressor:5 has four and gates and two xnor gates. The approximate multiplier designs are outlined here.

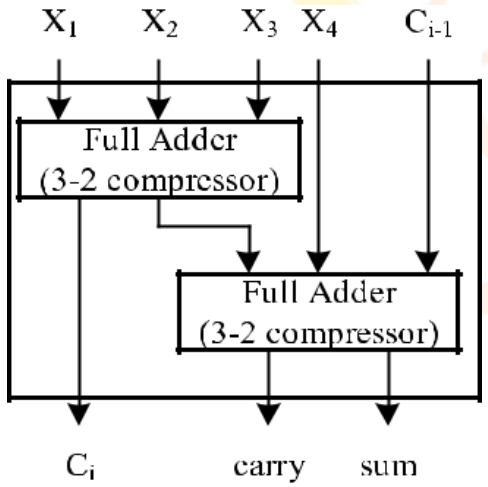

FIGURE 1.1 Precise 4 : 2 compressor.

Figure 1 depicts the overall block diagram of a precise 4 : 2 compressor. It has two cascaded full adders, three outputs, and five inputs. The exact 4 : 2 compressor's inputs are x1, x2, x3, x4 and ci, while its outputs are c0, carry, and sum. The values for co, carry, and sum are:

$$co = x3(x11 \oplus x2) + x11(x11 \oplus x2) \quad (1)$$

$$carry = ci\ (x1 \oplus x2 \oplus x3 \oplus x4) + x4(x1 \oplus x2 \oplus x3 \oplus x4) \quad (2)$$

$$sum = ci \oplus x1 \oplus x2 \oplus x3 \oplus x4 \quad (3)$$

Precise 2 - 2 compressors (half-adder) and 3 - 2 compressors (full adder) were employed in a Wallace tree multiplier to cut down on the number of rows of partial products. To create a more regular pattern, precise 4 - 2 compressors can also be employed in a Wallace Tree Multiplier. A carry propagating adder sums the reduced partial products to produce the original product. Because of its popularity, the exact 4 - 2 compressor was changed to create most earlier work's approximate 4 - 2 compressors, whereas the second approximate 4 - 2 compressor had four inputs and two outputs like other approximate 4 - 2 compressors. A 3-input max gate generated the carry bit, and the sum bit was always 1. The approximation 4 - 2 compressor was designed using a circuit stacking technique, which counted the amount of ones in the inputs. Four inputs are reduced to three by the stacking circuit, and a complete adder was utilized to calculate the carry and sum bits.

In the following subsection, the specifics of the precise and approximative compressors are discussed.

B.PRECISE AND APPROXIMATE COMPRESSOR

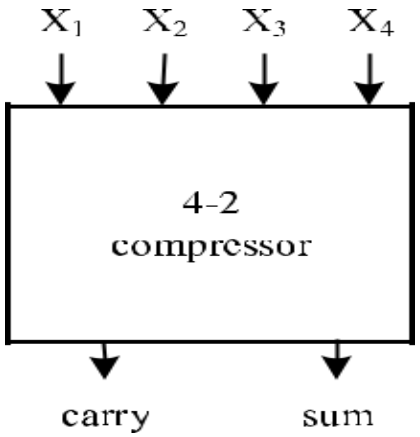

FIGURE 2. Approximate 4 : 2 compressor

These distributions taken into consideration to achieve a balanced tradeoff among throughput, usage of electricity, and precision. In comparison to the precise 4 - 2 compressor, the aforementioned approximate 4 - 2 compressors could reduce the partial products quickly and used less electricity.

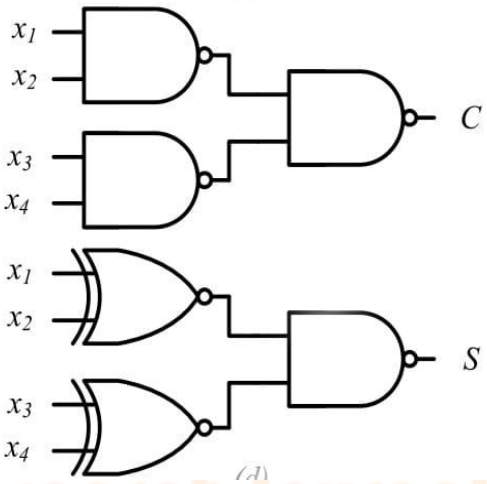

FIGURE 3. Approximate Compressor:2.

Two complete adders can be used to create a precise 4 - 2 compressor, as shown in FIGURE 1. In a multiplication, ci is the carry-in from the previous compressor for columns, and x1 and x4 are the partial products in the same column. 3 outputs—the ci, carry, and sum—will be generated by the precise 4 - 2 compressor. The block diagram for the majority of the roughly 4 - 2 compressors is shown in FIGURE 2. The approximation 4 - 2 compressor does not use ci-1 and does not produce ci in comparison to the exact 4 - 2 compressor in FIGURE 1. This results in a change of the number of inputs to four and the number of outputs to two. This considerably reduces the complexity of compressing the incomplete product.. When all four inputs are 1, however, errors are inevitable since the binary output value of '100' necessitates the use of at least three output ports.

C. ALTERABLE APPROXIMATE COMPUTING

An emerging design paradigm known as approximate computing broadly refers to a family of design methods that take advantage of inherent application resilience to create computing platforms that are quicker and use less power.

Compressor:1 refers to the four planned alterable approximate 4 - 2 compressors, each of which is made up of an approximate part and a supplemental part with a different construction. Both exact and approximate operation modes are available. The additional part is active in the precise mode and polishes the result from the approximation part to achieve precise results. The additional part is turned down with power gating while the approximation mode is active, allowing the approximate result to be achieved as the original result. The precise 2 - 2 (half-adder) compressor and the 3 - 2 compressor (full adder). In order to control whether the result is approximative or exact, the compressors contain a mask signal. We presented a programmable Reduceated multiplier and split two sub-multipliers of a multiplier to provide an alterable approximate multiplier that enables variations in the put in bit widths for multiplication. To generate the partial product elements with Reduction capability, they switched out the 2-input AND gates for 3-input AND gates. Certain incomplete product columns are trimmed at run-time when great precision is not required. In order to forecast and reconfigurableally change the precision of approximate multipliers for points with a precision preprocessor, the impact of bit-width precision on precision was examined.

D. ANALYSIS METHODS

The many performance indicators that are examined to determine the effectiveness of approximative multipliers and compressors are introduced in this section. Precision metrics and implementation efficiency measurements are two primary categories for the performance metrics.

A. RELIABILITY METRICS:

The level of precision attained by the multipliers created using suggested compressors and existing approximation compressors is measured using precision measures.

1) FAILURE DISTANCE (FD)

The term ED describes the variance between the precise and approximative 4 : 2 compressor output.

ED stands for Exactout-Approxout (4)

2) MEAN FAILURE DISTANCE (MFD)

The mean of the ED for all conceivable input combinations is referred to as the MED**.**

MFD=(1/2^2n)∑|Edk|. (5)

3)MEAN RELATIVE ERROR/FAILURE DISTANCE

(MRED) When we talk about MRED, we're talking about the average ED upon the associated Exactout for all potential input combinations.

MRFD=(1/2^2n)∑Edk/Exactoutput. (6)

(4) NORMALISED ERROR DISTANCE (NED)

For all conceivable input combinations, NED calculates the mean of ED normalised with the highest error in the suggested design.

NFD==(1/2^2n)∑Edk/Edmax. (7)

5) A TRUE COUNT OF THE OUTPUT (AOC)

AOC counts the exact outputs for each conceivable combination of inputs.

B. METRICS FOR IMPLEMENTATION EFFICIENCY

Using CMOS technology with a 45-nm process, the suggested compressors operate at 1 GHz with a 1 V supply voltage. Measures of implementation efficiency analyse using number of gates and delay. The suggested design's area refers to how effectively it optimises the hardware, making the design small. The amount of time it takes for a design to carry out its intended function is known as delay. For a comparable range of precision, an ideal design must include optimal elements including area, delay, and power.

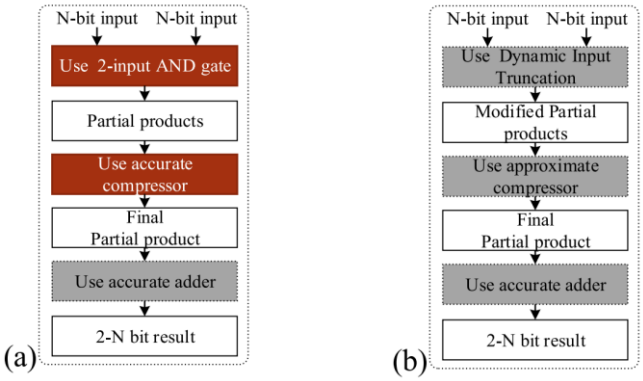

FIGURE 4. (a) Traditional and (b) our suggested

III. SUGGESTED APPROXIMATE MULTIPLIER

The traditional multiplication flow and our suggested multiplication flow are initially contrasted in this section. the high-precision 4 - 2 compressor we've suggested. The process then moves on to reconfigurable input Reduction, which is employed to create the alterable multiplier. Lastly, it discusses the suggested approximation multiplier's general architecture and explains why our suggested multiplier is appropriate for various applications.

A.SUGGESTED FLOW AND APPROXIMATE MULTIPLIER

The overall ow of the conventional ow for multiplication that yields precise results is shown in FIGURE 4(a). Using two-input AND gates, precise partial products are first created, which are then compressed using precise compressors. In order to produce the output, precise adders sum the compressed partial products. Our suggested ow for the suggested approximate multipliers is shown in FIGURE 4(b). The steps of creating partial products and compressing the partial products, which will be covered in Part III.B, are what set the suggested multiplication apart from conventional multiplication. We employ the reconfigurable input Reduction that will be covered

in Section III.C to generate the changed partial products during the stage of creating partial products.

B.SUGGESTED HIGH-PRECISION 4 - 2 COMPRESSOR

This paper proposes an approximation 4 - 2 compressor with good precision and low power. FIGURE 5 depicts the suggested 4 - 2 approximation compressor. The suggested 4 - 2 approximation compressor's design is explained in the paragraphs that follow. Eqs are used to construct w1, w4 from the four inputs x1, x4. (8)-(11). The carry bit in the suggested compressor is designed to always be correctly created since an incorrectly computed carry bit has a larger error distance than an incorrectly computed sum bit, i.e., an incorrect carry bit creates two times the ED of that produced by an incorrect sum bit. The carry bit generation equations are shown in (13)- (15).

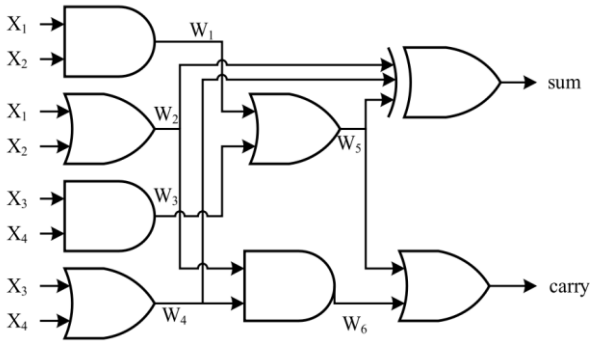

FIGURE 5. Implementation of the suggested 4 - 2 compressor at the gate level.

The carry bit for the following is x3, and x4 is also 1. The third condition is either x1 or x2 is 1 and x3 or x4 is 1. The final carry bit is produced after checking the first two conditions and the third situation.

w1 = x1 and x2 **(8)**

w2 = x1 or x2 **(9)**

w3 = x3 and x4 **(10)**

w4 = x3 or x4 **(11)**

w5 = w1 or w3 **(12)**

w6 = w2 and w4 **(13)**

sum = w5 or w2 or w4 **(14)**

carry=w5 or w6 **(15)**

The suggested equation for producing the sum bit is displayed in (14). The sum bit in a precise 4 - 2 compressor is produced using four XOR gates integrated into the two complete adders. In contrast, in our suggested compressor, we employ the signals that are used to create the carry bit to generate the sum bit by feeding them into a 2-input XOR gate. We can reduce the circuit area and static usage of electricity by using shared signals. However, we discovered that when only w2 and w4 are supplied into a 2-input XOR gate, the error distance is significant. The error happens when both x1 and x2 are 1 or when both x3 and x4 are 1, which causes the sum bit to result in 1 when it should have been 0. This is because w2 and w4 are created with OR gates. We incorporate w5, the signal used to identify these two situations, inside the XOR gate in order to obtain high precision. As an illustration, if x1 and x2 are both 1, w2 and w5 will both be 1, and the sum bit is '0 XOR w4', w4 will be the outcome. Just x3 and x4 bits must be taken into account in this situation. Yet when each of the four inputs is 1, the aggregate bit also becomes 1, which results in an error distance of 1.

The truth table for our suggested roughly 4 - 2 compressor is shown in TABLE 1. Only when all four inputs are equal to 1 will there be an error. Additionally, even if an error does occur, the difference between the precise result and our output is only 1, which is insignificant.

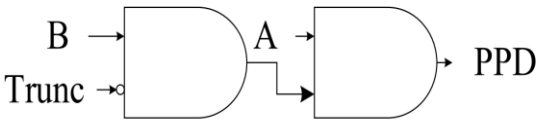

FIGURE 6:Changed Partial Product.

Error = w1 and w3 **(16)**

When W1 uses an AND gate to determine whether both x1 and x2 are 1, and W3 uses an AND gate to determine whether both x3 and x4 are 1, we simply need an additional AND gate to determine if both w1 and w3 are 1. The error detection circuit (EDC) equation is displayed in (16). As a result, by simply adding one additional AND gate, the suggested 4 - 2 compressor's error compensation circuit can be quickly built.

TABLE2: Truth table for our suggested approximate 4 - 2 compressor

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

C. RECONFIGURABLE INPUT REDUCTION

We suggest a reconfigurable input Reduction technique that combines two 2-input AND gates, as shown in FIGURE 6, to create a partial product, whose equation is shown in (17), where A is the multiplicand and B is the multiplier, in order to achieve an alterable approximate multiplier at runtime. The partial product PPD is evaluated to see if it should be shortened using the Reduce signal. The partial product is trimmed to 0 if the

Reduce value is 1. To be more exact, the Reduce signals reduce the PPDs in the multiplications to zeros in order to conserve power. In other words, we can consider the Reduce signals to be the hardware units in the appropriate columns being disabled.

PartialProducts = (Reduce, bi, and aj) (17)

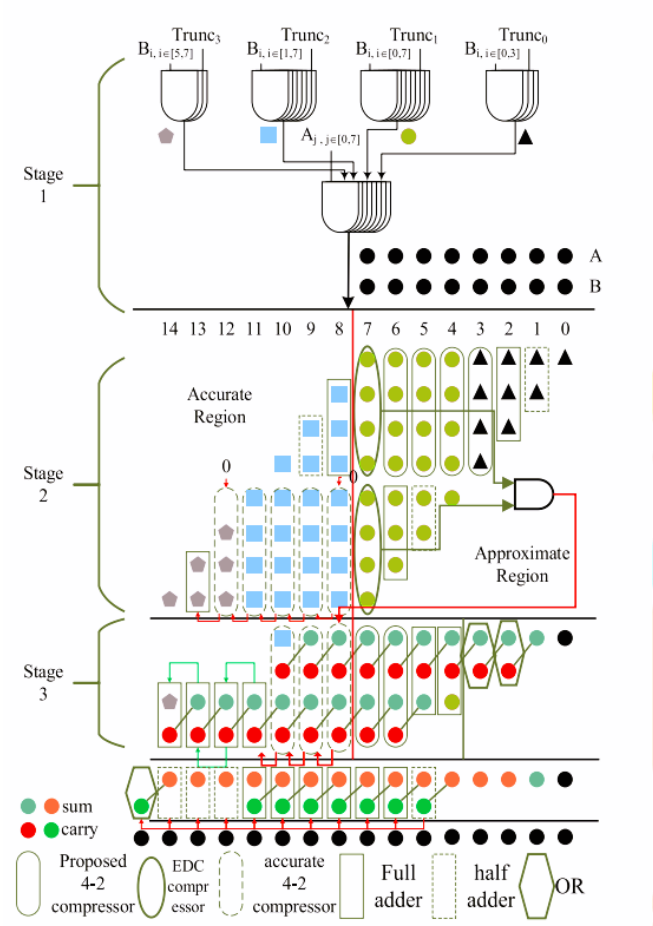

FIGURE 7: Our Suggested Approximate Multiplier.

D. THE SUGGESTED APPROXIMATE MULTIPLIER

FIGURE 7 depicts a rough multiplier using the suggested method. The suggested technique can still be applied to larger multipliers even though the multi-input plier's width is only intended to be 8 bits. Three phases make up the suggested approximate multiplier. Each partial product is created in the first stage by two 2-input AND gates, as previously illustrated in FIGURE 5. To further lower hardware costs, FIGURE 6 applies the gate sharing technique. Depending on the needs, the Reduce signal can be used to assess the incomplete product's correctness. In our suggested approximate multiplier, we build a 4-bit Reduce signal with each bit controlling multiple partial product columns, which we refer to as the "3-4-4-4 partition." In particular, each bit from the MSB to LSB controls column 14, 12, 11, 8, and 6, corresponding to the seventh, fourth. In Stage 2 of FIGURE 7, the colours of light yellow, light sky blue, green, and black are used. In this case, we selected the 3-4-4-4 partition for the roughly multiplied value mentioned above. To compare the outcomes of all other approximative multipliers, we have almost taken all 256 options. We've provided inputs a, b, and Reduce in the first stage. A, b are each 8 bits, while Reduce is a 4 bit input. There are two substages in the second stage. The partial product rows are created in two halves, one of which is approximate and the other precise, as illustrated in FIGURE 7. The operation is carried out in the third stage based on reconfigurable input Reduction. Finally, if the Reduce signal is applied, it Reduces the remaining bits. As we indicated previously, the output is kept in the provided arrays. The final two partial product rows are obtained once the third stage is complete, and these two rows are then added together precisely to yield the results.

IV. SETUP AND RESULTS OF EXPERIMENTS

The experimental setup that will be utilised to assess the approximative multipliers is initially described in this section. The approximative multipliers are then compared using several evaluation metrics. The critical path duration, error distance, and number of logic gates utilised in each design in the study are then compared.

Type of Multiplier	No of LUT's	No of IO Buffers	No of XOR's	Delay (ns)	Levels of Logic	Memory Usage(kb)	Accuracy
Exact Multiplier	98	36	66	6.912	11	4617468	100%
Multiplier 1	119	36	87	7.124	12	4617532	88.60%
Multiplier 2	115	36	85	6.779	11	4617504	79.40%
Multiplier 3	117	36	87	6.779	11	4617516	88.45%
Multiplier 4	109	36	87	6.965	11	4617504	76.32%
Multiplier 5	112	36	88	7.084	11	4617508	64.79%

TABLE 3: Comparison On Different Metrics Along With Multipliers.

We can see that, in order to preserve the trade-off between all other metrics of other approximate multipliers, Compressor:1 is the most appropriate for approximation multiplication. 11 architectural designs from the literature and an exact 4: 2 compressor are compared to the high speed, area-efficient 4: 2 compressor design that is being suggested. Xilinx is used to implement the designs.Table 3 provides precision data for the suggested high speed, area-efficient 4: 2 compressor. Observations indicate that Compressor:1, together with the others and the reconfigurable input Reduction approach, has the best precision. We can also see that Compressor:4 has the lowest precision, despite the fact that the design uses relatively few transistors. However, this means that we must compromise precision in favour of area, and the suggested compressor has a considerably smaller delay at the expense of precision and power. The approximation multipliers' RTL codes are implemented in Verilog HDL, and Xilinx is used to simulate them and create waveforms that capture the switching actions of logic gates. Just the 4 : 2 approximate compressor comparisons amongst the other approximate compressor types are shown in the above table. First off, using the Xilinx tool and VHDL, every single basic component is built from the ground up. Subsequently, error compensation circuits will be included to every design by designing the suggested approximate multiplier, Precise Multiplier, and other approximate multipliers. Finally, We can observe the simulation results by giving an input value like all possible combinations and compare all the inputs of every other approximate multipliers to calculate the precision and precision metrics in the design.

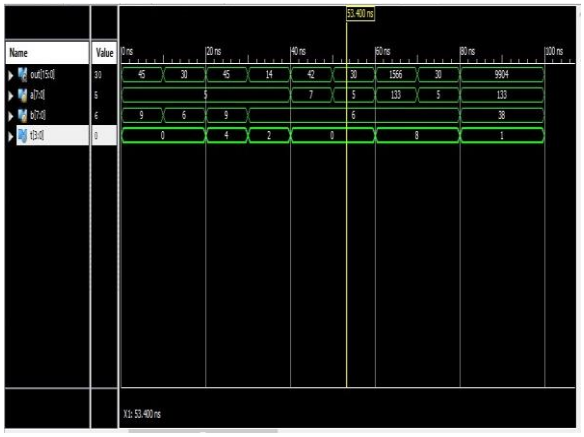

FIGURE 8 :Simulation Results Of Suggested Compressor with Multiplier

The comparison of various precision metrics between the suggested approximative multipliers and all other approximative multipliers will be covered in the section that follows.

COMPRESSOR DESIGN USED	AOC	MED	MRED	NED
Accurate	-	-	-	-
Compressor1(Proposed)	2048	800.125	0.101476	0.084473
Compressor2	2247	878.0313	0.170748	0.093647
Compressor3	2051	801.3438	0.100622	0.084601
Compressor4	2342	914.8438	0.275621	0.099787
Compressor5	257	100.438	020448	0.07655

TABLE 4: Comparison On Precision Metrics .

As mentioned above, TABLE 4 compares several precision metrics for all approximative multipliers used in this design. It can be seen that the suggested design, with one exception when compared with other approximative multipliers, is able to attain equivalent precision metrics with other designs with error rates of 25%. Thus, we can state that the suggested multiplier is appropriate for all situations in which multiplication is used.

VI. CONCLUSION AND FUTURE WORKS

The construction of an approximation multiplier using a high precision approximate 4 - 2 compressor is suggested in this research. To improve precision, the suggested approximate multiplier reconfigurableally Reduces incomplete products. From inspecting all simulations on different Approximate Compressors, We can conclude that our suggested compressor makes a better trade-off and produce best results. So, Our suggested compressor is the required one that can be used in further applications like multiplication. Our suggested multiplier has the lowest mean error distance when compared to other approximation multipliers. The error distance can be decreased in the upcoming work by using a straightforward error compensation circuit.

We haven't yet discovered the ideal formulas for dividing up the Reduce signals, though. Future study entails a more thorough examination of various partitioning techniques in order to obtain more precise or mathematical expressions for the connection between hardware costs, precision, and usage of electricity. We can see from the current study that the alterable approximate multiplier that has been provided needs different Reduce signals for various networks, or more specifically, various convolutional layers, in order to produce satisfactory results. Future work will emphasise this element. Examining the characteristics or features of various convolutional layers to determine the best a group of parameters based on the different layer kinds is a workable solution. The users can then employ the suggested multipliers after having discovered those previously investigated parameters in this way.

REFERENCES

[1]Power_and_High-Precision_Approximate_Multiplier_With_Reconfigurable_Reduction ,FANG-YI GU , (Graduate Student Member, IEEE), ING-CHAO LIN , (Senior Member, IEEE), AND JIA-WEI LIN, (Graduate Student Member, IEEE.

[2] B. Moons and M. Verhelst, ‘‘DVAS: Reconfigurable voltage precision scaling for increased energy-efficiency in approximate computing,’’ in Proc. IEEE/ACM Int. Symp. Low Power Electron. Design (ISLPED), Jul. 2015, pp. 237–242.

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

[4] M. de la Guia Solaz, W. Han, and R. Conway, ‘‘A flexible low power DSP with a programmable Reduceated multiplier,’’ IEEE Trans. Circuits Syst., vol. 59, no. 11, pp. 2555–2568, Nov. 2012.

[5] R. Zendegani, M. Kamal, M. Bahadori, A. Afzali-Kusha, and M. Pedram, ‘‘RoBa multiplier: A rounding-based approximate multiplier for high speed yet energy-efficient digital signal processing,’’ IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 25, no. 2, pp. 393–401, Feb. 2017. [6] C. S. Wallace, ‘‘A suggestion for a fast multiplier,’’ IEEE Trans. Electron. Comput., vol. EC-13, no. 1, pp. 14–17, Feb. 1964.

[7] A. Weinberger, ‘‘4 : 2 carry-save adder module,’’ IBM Tech. Discl. Bull., vol. 23, no. 8, pp. 3811–3814, 1981.

[8] 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.

[9] Z. Yang, J. Han, and F. Lombardi, ‘‘Approximate compressors for errorbresilient multiplier design,’’ in Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Nanotechnol. Syst. (DFTS), Oct. 2015, pp. 183–186.

[10] C.-H. Lin and I.-C. Lin, ‘‘High precision approximate multiplier with error correction,’’ in Proc. IEEE 31st Int. Conf. Comput. Design (ICCD), Oct. 2013, pp. 33–38.

[11] P. J. Edavoor, S. Raveendran, and A. D. Rahulkar, ‘‘Approximate multiplier design using novel dual-stage 4 : 2 compressors,’’ IEEE Access, vol. 8, pp. 48337–48351, 2020. [12] 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.

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

[14] H. Xiao, H. Xu, X. Chen, Y. Wang, and Y. Han, ‘‘Fast and high-precision approximate MAC unit design for CNN computing,’’ IEEE Embedded Syst. Lett., early access, Dec. 21, 2021, doi: 10.1109/LES.2021.3137335.

[15] O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, ‘‘Dual-quality 4 : 2 compressors for utilizing in reconfigurable precision configurable multipliers,’’ IEEE Trans. Very Large

Scale Integr. (VLSI) Syst., vol. 25, no. 4, pp. 1352–1361, Apr. 2017.

[16] T. Yang, T. Ukezono, and T. Sato, ‘‘A low-power high-speed precisioncontrollable approximate multiplier design,’’ in Proc. 23rd Asia South Pacific Design Autom. Conf. (ASP-DAC), Jan. 2018, pp. 605–610.

[17] I. Hammad, L. Li, K. El-Sankary, and W. M. Snelgrove, ‘‘CNN inference using a preprocessing precision controller and approximate multipliers with various precisions,’’ IEEE Access, vol. 9, pp. 7220–7232, 2021.

[18] C. Guo, L. Zhang, X. Zhou, W. Qian, and C. Zhuo, ‘‘A reconfigurable approximate multiplier for quantized CNN applications,’’ in Proc. 25th Asia South Pacific Design Autom. Conf. (ASP-DAC), Jan. 2020, pp. 235–240.