Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 5, May : 2024

High Speed and Area Optimized VLSI Architecture of Probability Based Approximate Multiplier

U Bharadwaja[1], Vulasala Neelavathi[2]

[1,2]Assistant Professor, Nalla Narasimha Reddy Education Society's Group of Institutions
Chowdariguda Hyderabad TELANGANA

ABSTRACT

When it comes to error-resistant applications, approximate computing has the potential to reduce the complexity of the design while simultaneously improving performance in terms of area, latency, and power efficiency. A new design technique for approximating multipliers is the topic of discussion in this particular brief. When it comes to error-resistant applications, approximate computing has the potential to reduce the complexity of the design while simultaneously improving performance in terms of area, latency, and power efficiency. With the help of this short, a unique 4-2 approximate compressor is proposed. This compressor is supplementary to other compressors that have been developed in previous work. The results of the simulation indicate that the approximation multiplier that was developed has a satisfactory level of performance.Through the use of the verilog HDL programming language, the implementation, synthesis, and simulation are carried out and recorded in the Xilinx-ise.

Key words: Approximate computing, 4-2 compressor, or gate, accurate compressors digital multiplier, compressor combination, error compensation.

1.INTRODUCTION

In the current state of the art of sophisticated sign preparation and many applications, multipliers play a vital role. The expansion and duplication of two paired integers is a fundamental and regularly used number-crunching activity that is widely utilised in advanced frameworks. According to statics, more than seventy percent of the directions in a microchip and the majority of DSP computations, expansion and duplication are performed. The execution time is therefore defeated by these jobs. Simply said, this is the reason why a rapid multiplier is required. A increasing number of applications for personal computers and sign preparation have contributed to an increase in the desire in speedy handling. Additionally, a fundamental problem in multiplier design is the utilisation of low force. In order to reduce the utilisation of enormous force, it is a good idea to reduce the amount of activity hence reducing the unique force, which is a big component of the utilisation of absolute force. As a result, the need for high velocity and low force multiplier has increased. A high velocity and low force competent circuit layout is the primary focus of Creator's attention. It is the purpose of a good multiplier to provide a device that is not only compact but also quick and requires a low amount of force to operate.

One further key challenge that arises in multiplier design is the use of low force. This will result in a reduction in dynamic force, which is a large component of absolute force utilisation. As a result, the need for a force multiplier that is both quick and low intensity has increased. Fast and low-force effective circuit planning is the primary emphasis of planners for the most part. The purpose of a good multiplier is to provide a unit that is extremely compact, moves at a fast speed, and requires a low amount of force to operate. A single cycle multiply–amass unit typically spent the vast majority of the chip region of early digital signal processors. Advanced sign preparing calculations spend the majority of their energy duplicating; computerised signal processor planners sacrifice a significant amount of chip territory in order to cause the to increase as quickly as possible from the beginning. The majority of computerised frameworks will require a significant amount of work to establish. Individuals who have received training in any subject will divide the initial project into logical subunits that serve as building blocks, and they will make use of the standard subunits whenever they arise. The building blocks that are used in computerised equipment are referred to by names such as adders, registers, and multiplexers.

A method for addressing the issue of NP-completeness in the context of an optimisation problem is known as an approximation algorithm. Using this method does not ensure that the best answer will be found. The objective of the approximation algorithm is to arrive at the ideal answer in polynomial time, or as near to it as feasible given the available information. Approximation algorithms and heuristic algorithms are the names given to these types of algorithms. To put it simply, we are utilising approximation techniques in order to boost the speed of the design. In addition, this helps to decrease the amount of space and power that the design requires.

The most popular units are approximation multipliers since they are easy to adjust. A multiplier can use approximation methods to generate partial products, accumulate and reduce them, and add them. During the process of multiplication, the second stage, which consist of lowering partial products by accumulation, uses the greatest amount of resources [2]. The acceleration of the accumulation is often accomplished by the utilisation of approximate compressors, such as 4-2 compressors. On the basis of the quantity of output mistakes, Kong and Li [3] construct a wide classification system for compressors, dividing them into high-accuracy and low-accuracy groups. Low-accuracy approximation compressors are primarily concerned with minimising the amount of power that is lost.An approximation compressor is the subject of this brief study, which demonstrates the following contributions:

- A novel compressor with a small area, around 4-2, and just four logic gates is an idea that has been offered. Specific input patterns cause output errors.
- Esposito's compressors are coupled in an innovative manner to limit the possibility of input pattern mistakes.
- The recommended compressor is used to build a low-resource approximation multiplier that reduces mistakes and improves electrical performance.
- The simulation shows that the approximation multiplier outperforms existing multipliers in PDAP balance and accuracy tradeoff.

2.LITERATURE SURVEY

G. D. Meo published their findings.Recursive multipliers (RMs) are low-power multipliers because they offer several power-quality tuning options. Although 22 multipliers are the basic building elements of this recursive architecture, most state-of-the-art approximation recursive designs use 44 multipliers. Because of this, the investigation of the design space for AxRMs that utilise 2×2 multipliers is still a subject that is actively being researched. It is necessary to have 2-bit multipliers that are both high-performing and low-area in order to include configurability and flexibility into the design of AxRMs. In this article, two approximation multipliers with dimensions of 2×2 are suggested, both of which exhibit a double-sided component of error distribution. The proposed approach reduces space by 52% and improves latency by 25% compared to the current 2x2 multiplier. Convolutional neural networks, which are considered to be the most advanced error-tolerant application, are used to implement the approximation multipliers that have been developed. The best quality-power trade-off is provided by AxRM2, which achieves a 32.64 percent reduction in power consumption while simultaneously improving classification accuracy by 1.10 percent.

3.EXISTING METHOD

APPROXIMATE 4-2 COMPRESSOR AND PROBABILITY-BASED ERROR ADJUSTMENT FOR APPROXIMATE MULTIPLIER

An error of minus one is produced by a 4-2 compressor that was invented by Kumar and colleagues. This occurs when the final two bits of the input information are 1. In order to correct for the inaccuracy, which is something that is comparable to, an AND gate that is attached to the final two bits of the compressor's input is utilised. In addition, the final four bits of the multiplier are shortened by employing an unconventional approach that replaces the conventional 0000 with the average number 0110. Taking this action will allow for a reduction in the error without requiring any more work. When compared to other multipliers that use the compressor, the approximation multiplier in has one of the greatest levels of accuracy. The truth table contains four different error possible outcomesHowever, the multiplier converts a half adder into a full adder and a full adder into an exact compressor to get the first-level error correcting AND gate output. Finishing the procedure requires this. Error correction modules also lengthen the propagation channel, increasing multiplier delay.

4-2 APPROXIMATE COMPRESSOR

We are able to correct compressor faults by utilising a straightforward logic gate, As observed in the scenario when we force the wrong output at specific input characteristics. This technique may be modified to accommodate situations when there are more than four instances of each of the sixteen input patterns available. A novel, space-saving 4-2 compressor was designed with the aim of enhancing the regularity of the Karnaugh map and minimising the discrepancy between the S (sum) and C (carry) outputs and the actual result.

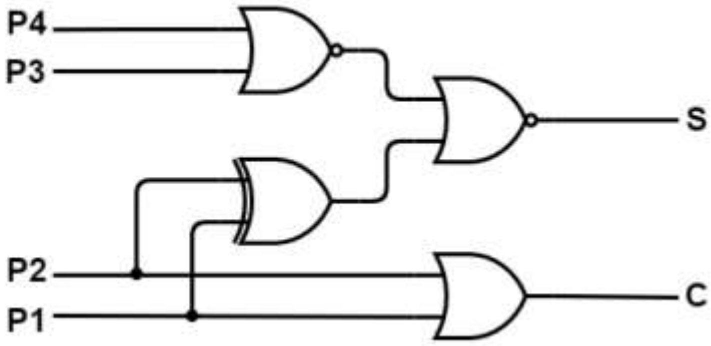


Fig. 1. Architecture of the proposed compressor

Table I displays the anticipated outputs of the compressor, namely the S and C values. Table I provides the odds ratio (OR) for variable C, as shown by equation (1). Equation (2) assumes the minimum number of logic gates necessary for the expression of S. This is in accordance with the rules and principles of De Morgan. It should be brought to your attention that the performance of NAND and NOR gates is superior to that of AND and OR gates.

Figure 1 depicts the suggested design for a 4-2 compressor. This compressor utilises a small number of transistors due to its composition of only two NOR gates, one XOR gate, and one OR gate. Despite having six flaws among the sixteen input patterns, the design remains attractive and competitive. The Esposito's 3-2 or 4-2 compressor yields an output of 11 when at least two of its inputs are set to 1. To mitigate the likelihood of errors, we may enhance the accuracy by integrating Esposito's compressors for the early partial products with the recommended compressor at the second level. We found this. See Figure 2 for the 8-bit hybrid approximation multiplier construction. Based on two approximation compressors and the error correcting module, this multiplier also known as the AND gate, as well as constant approximation from.

A ripple carry adder, also known as an RCA, is utilised for the extremely last addition. When it comes to Esposito's compressors, the first level makes use of two different types: 3-2 and 4-2. This is demonstrated in Figure 2. As seen in Table I, the chance of the outputs w1 and w2 being zero is the same for Esposito's 4-2 compressor, which is 135/256, which is around 0.527. According to Esposito's 3-2 compressor, the likelihood of one of the outputs, w1, being zero is around 0.563, while the probability of the second output, w2, being zero is approximately 0.703. C4 to C1 are the names given to the four distinct input patterns that are included inside the four inexact columns. Table II displays the probabilities of plus- and minus-one mistakes in each column. Table I uses the term "origin" to describe the likelihood of compressor input patterns at the first level with direct connections to their initial partial products.

Due to its higher negative-one error probability and lower positive-one error possibility, the suggested hybrid approximation multiplier is more suited for utilisation with an error correcting AND gate. The concept of hybridization is implemented using a 16-bit approximation multiplier. The initial and final phases of the multiplier employ Esposito compressors. At the second level, two error-correcting AND gates reduce error distance while conserving resources. The same truncation approach is used to apply several compressors from, and hence shorten, the unified Dadda structure. This represents an 8-bit approximation multiplier. Note that just the specified compressors are used. Figure 2 depicts the hybrid structure produced and compared to others.

The implementation of 16-bit approximate multipliers includes 10 approximate compressor columns and six constant approximation columns. Specifically, the authors' goals were realised by including more error correcting modules within and applying to their multipliers. Compressor multipliers only employ compressors on the first layer. This is because the second layer is quite incorrect. The quantification of precision in approximation computing is commonly done through the use of two metrics: the normalised mean error distance (NMED) and the mean relative error distance. In order to get accurate error measurements, we conduct a comprehensive analysis of 65536 instances for 8-bit multipliers, and an average of 10,000 distinct 1 million random cases for 16-bit multipliers. Esposito and Kumar's designs exhibit superior MRED (Minimum Relative Error Distance) and NMED (Normalised Mean Error Distance) due to their reduced reliance on approximation compressors and error correction units.

The hybrid approximation multiplier (ProposedH) has superior NMED and MRED values in the low-accuracy category, along with a significant 60.65% decrease in error compared to the proposed multiplier. However, the multiplier based on the proposed compressor demonstrates poor performance. Table IV illustrates that the hybrid 16-bit approximation multiplier was specifically developed to be competitive in terms of achieving a high level of accuracy. Pei's design is more precise due to the inclusion of an additional column of precise compressors and an additional layer of approximate compressors.

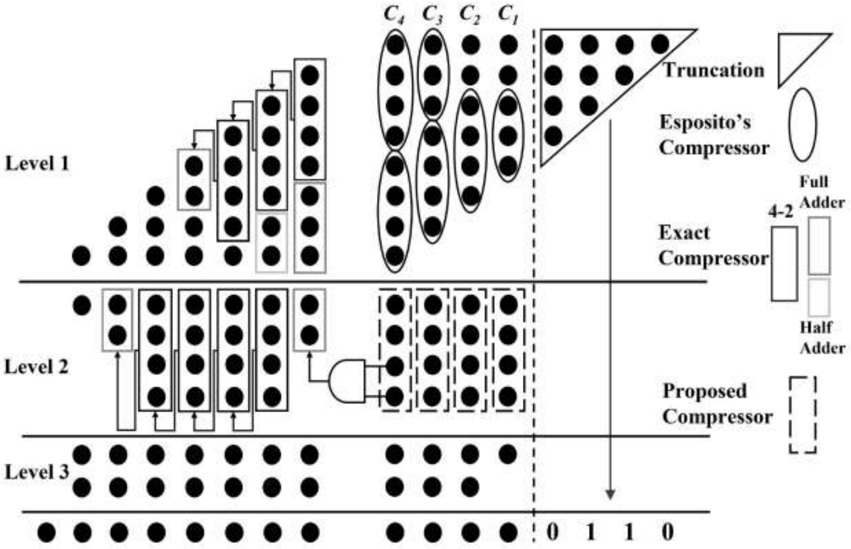


Fig. 2. Proposed hybrid approximate multiplier structure (ProposedH).

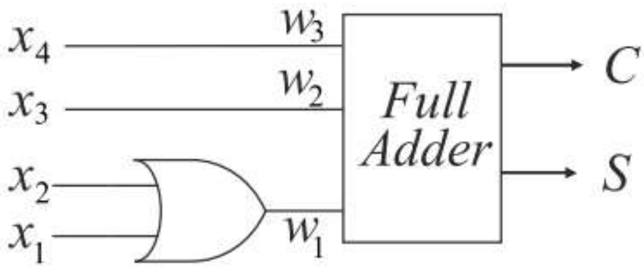


Fig 3: esposito's compressor

4.PROPOSED METHOD

APPROXIMATE MULTIPLIER DESIGN USING NOVEL 4: 2 COMPRESSORS

There is a growing agreement that approximation multipliers are the best option to reduce processing energy, especially in error-tolerant applications. The inclusion of accuracy as a design parameter together with performance, area, and power makes choosing the best approximation multiplier problematic. This study found that when choosing an approximate multipliers circuit, three main factors should be considered: (1) the type of approximate area efficient compressor and dual quality compressor used in its construction, (2) the multiplier's design, which may be an array or a tree, and (3) the positioning of sub-modules of approximate and exact multipliers within the main multiplier. We examined the circuit-level design space for approximation multipliers based on these factors. Some popular compressors were used at circuit level.

EXACT 4:2 COMPRESSOR

An identical 4:2 compressor block diagram is shown in Figure 1. Five inputs, three outputs, and two cascaded full adders comprise this system. The exact 4:2 compressor has A1, A2, A3, A4, and CIN inputs and COUT, CARRY, and SUM outputs. We provide COUT, CARRY, and SUM.

$$\mathrm{COUT} = \mathrm{A3}(\mathrm{A1} \oplus \mathrm{A2}) + \mathrm{A1}(\mathrm{A1} \oplus \mathrm{A2}) \tag{1}$$

$$\mathrm{CARRY} = \mathrm{CIN}\,(\mathrm{A1} \oplus \mathrm{A2} \oplus \mathrm{A3} \oplus \mathrm{A4}) + \mathrm{A4}(\mathrm{A1} \oplus \mathrm{A2} \oplus \mathrm{A3} \oplus \mathrm{A4}) \tag{2}$$

$$\mathrm{SUM} = \mathrm{CIN} \oplus \mathrm{A1} \oplus \mathrm{A2} \oplus \mathrm{A3} \oplus \mathrm{A4} \tag{3}$$

Figure 4 illustrates a compressor chain in its entirety. The CIN value is the input carry from the 4:2 compressor that came before it and processed the bits with a lower significance level. The input CIN is less significant than the outputs CARRY and COUT, which are of order '1' and have a larger relevance. This is the truth table for the exact compressor, which is presented in Table 1.

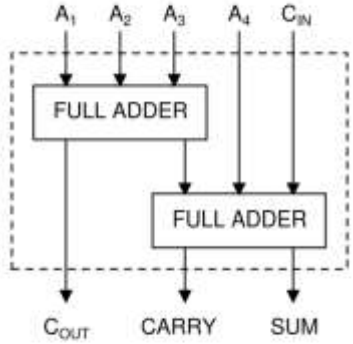
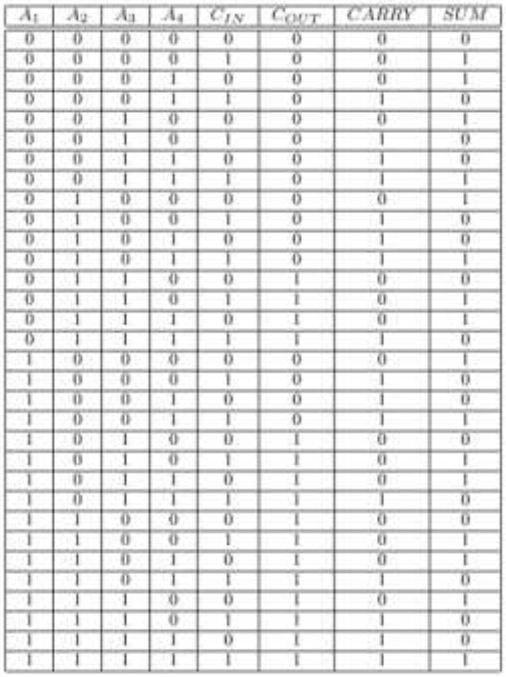


Fig4. Exact 4:2 compressor.

TABLE 1. Truth table for exact 4:2 compressor.



A_1	A_2	A_3	A_4	C_{IN}	C_{OUT}	$CARRY$	SUM
0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	1
0	0	0	1	0	0	0	1
0	0	0	1	1	0	1	0
0	0	1	0	0	0	0	1
0	0	1	0	1	0	1	0
0	0	1	1	0	0	1	0
0	0	1	1	1	0	1	1
0	1	0	0	0	0	0	1
0	1	0	0	1	0	1	0
0	1	0	1	0	0	1	0
0	1	0	1	1	0	1	1
0	1	1	0	0	1	0	0
0	1	1	0	1	1	0	1
0	1	1	1	0	1	0	1
0	1	1	1	1	1	1	0
1	0	0	0	0	0	0	1
1	0	0	0	1	0	1	0
1	0	0	1	0	0	1	0
1	0	0	1	1	0	1	1
1	0	1	0	0	1	0	0
1	0	1	0	1	1	0	1
1	0	1	1	0	1	0	1
1	0	1	1	1	1	1	0
1	1	0	0	0	1	0	0
1	1	0	0	1	1	0	1
1	1	0	1	0	1	0	1
1	1	0	1	1	1	1	0
1	1	1	0	0	1	0	1
1	1	1	0	1	1	1	0
1	1	1	1	0	1	1	0
1	1	1	1	1	1	1	1

AREA-EFFICIENT APPROXIMATE 4 : 2 COMPRESSOR

Figure 5 depicts the proposed high-speed, area-efficient 4:2 approximation compressor. All compressor inputs are A1, A2, A3, and A4, while the outputs are CARRY and SUM. SUM is created utilising multiplexers (MUX). When the MUX is activated, the chosen line serves as the XOR gate output. (A3A4) is chosen when the choose line is high, and (A3 + A4) when it is low. The suggested 4:2 compressor simplifies carry-generating logic to an OR gate. Add an error with a distance of one to the exact compressor's truth table. The logical equations for SUM and CARRY are shown here.

$$\text{SUM} = (\text{A1} \oplus \text{A2})\ \text{A3A4} + (\text{A1} \oplus \text{A2})\ (\text{A3} + \text{A4}) \qquad (4)$$

$$\text{CARRY} = \text{A1} + \text{A2} \qquad (5)$$

Table 2 shows that the suggested 4:2 compressor has produced errors for input values − {0011}, {0100}, {1000}, and {1111} to achieve equal positive and negative deviation with ED = 1 (minimum).

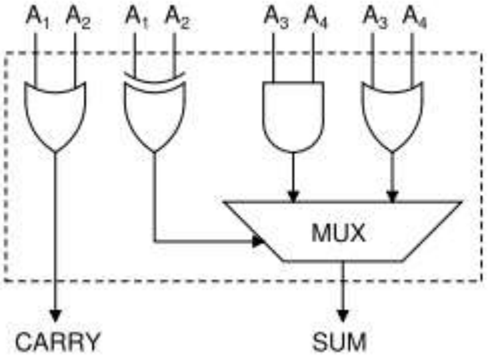
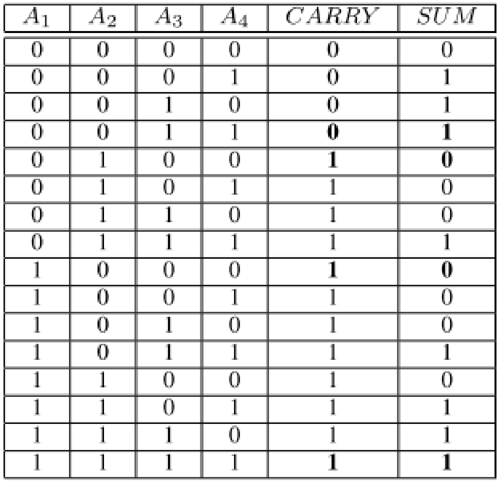


Fig 5. Area-efficient 4:2 compressor.



A_1	A_2	A_3	A_4	$CARRY$	SUM
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	0	1
0	0	1	1	**0**	**1**
0	1	0	0	**1**	**0**
0	1	0	1	1	0
0	1	1	0	1	0
0	1	1	1	1	1
1	0	0	0	**1**	**0**
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	1	1
1	1	0	0	1	0
1	1	0	1	1	1
1	1	1	0	1	1
1	1	1	1	**1**	**1**

TABLE 2. Truth table for proposed area efficient 4:2 compressor

The area-efficient 4:2 compressor, or gate, and the 4:2 precise compressor were utilised in the design of the proposed multiplier architecture during its development.In this instance, the design is split into two parts: the first is the msb component, and the second is the lsb part. approximation compressors and gates are being utilised in the LSB portion of the design. These approximation adders are helping to decrease the amount of space, latency, and power consumption that the system requires. When compared to complete adders, the latency associated with the msb component was reduced because to the utilisation of correct compressors during the design process. And lastly, the entire design was developed by utilising dadda structure. These cstructures minimise the amount of half adder usages, as seen in the image below.

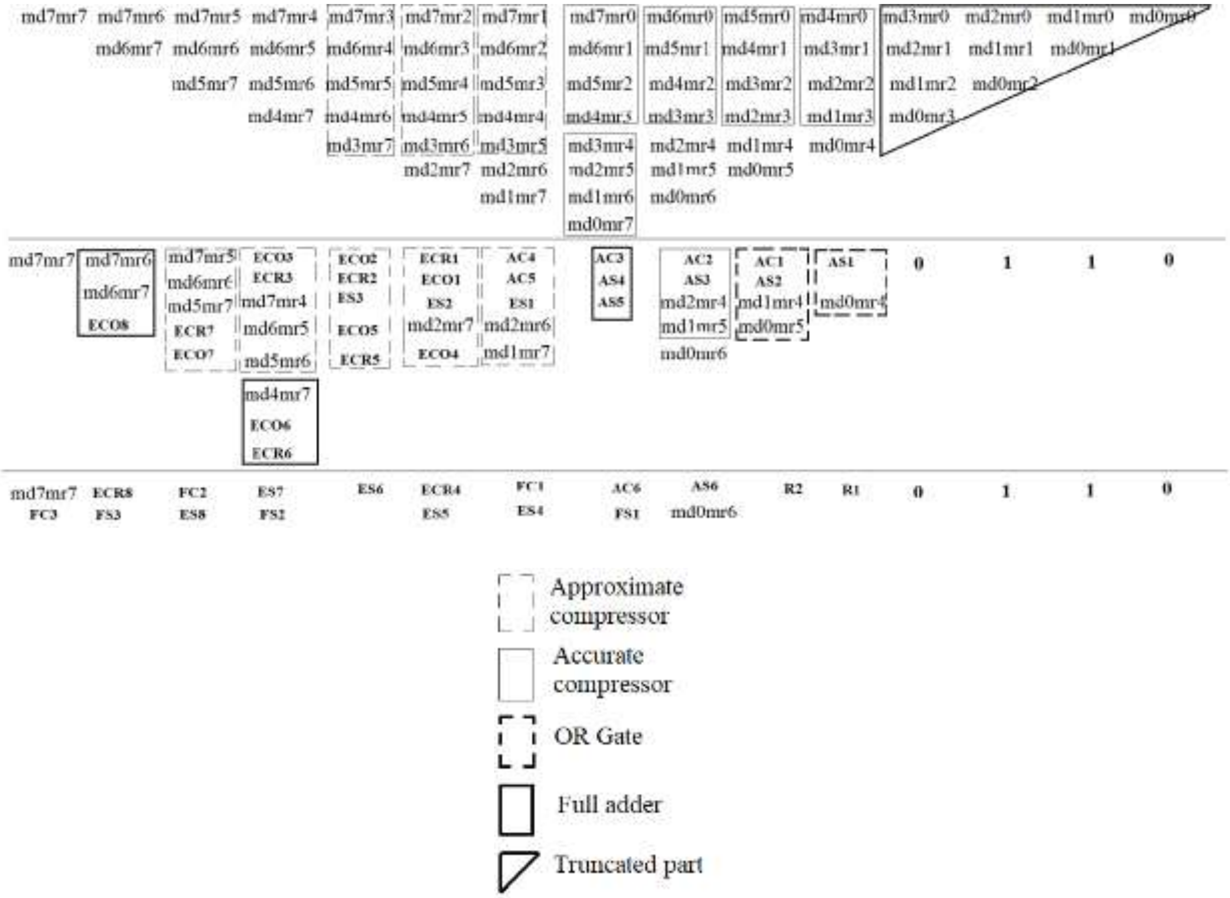

md7mr7 md7mr6 md7mr5 md7mr4 md7mr3 md7mr2 md7mr1 md7mr0 md6mr0 md5mr0 md4mr0 md3mr0 md2mr0 md1mr0 md0mr0
md6mr7 md6mr6 md6mr5 md6mr4 md6mr3 md6mr2 md6mr1 md5mr1 md4mr1 md3mr1 md2mr1 md1mr1 md0mr1
md5mr7 md5mr6 md5mr5 md5mr4 md5mr3 md5mr2 md4mr2 md3mr2 md2mr2 md1mr2 md0mr2
md4mr7 md4mr6 md4mr5 md4mr4 md4mr3 md3mr3 md2mr3 md1mr3 md0mr3
md3mr7 md3mr6 md3mr5 md3mr4 md2mr4 md1mr4 md0mr4
md2mr7 md2mr6 md2mr5 md1mr5 md0mr5
md1mr7 md1mr6 md0mr6
md0mr7
md7mr7
md7mr6 md6mr7 ECO8
md7mr5 md6mr6 md5mr7 ECR7 ECO7
ECO3 ECR3 md7mr4 md6mr5 md5mr6
md4mr7 ECO6 ECR6
ECO2 ECR2 ES3 ECO5 ECR5
ECR1 ECO1 ES2 md2mr7 ECO4
AC4 AC5 ES1 md2mr6 md1mr7
AC3 AS4 AS5
AC2 AS3 md2mr4 md1mr5 md0mr6
AC1 AS2 md1mr4 md0mr5
AS1 md0mr4
0 1 1 0
md7mr7 FC3
ECR8 FS3
FC2 ES8
ES7 FS2
ES6
ECR4 ES5
FC1 ES4
AC6 FS1
AS6 md0mr6
R2
R1
0 1 1 0
Approximate compressor
Accurate compressor
OR Gate
Full adder
Truncated part

Fig 6.proposed design

5.RESULTS

RTL SCHEMATIC:- RTL stands for register transfer level, which is an abbreviation for the register transfer level schematic. This schematic represents the blueprint of the architecture and is utilised to test that the designed architecture is comparable to the ideal architecture that we are yet to construct.The coding language known as verilog.vhdl is utilised in order to transform the description or summary of the architecture into the functioning summary. This is accomplished through the utilisation of the hdl language. The RTL schematic even includes a description of the internal connection blocks, which allows for more accurate analysis.A schematic representation of the RTL architecture that was created may be seen in the figure that is shown below.

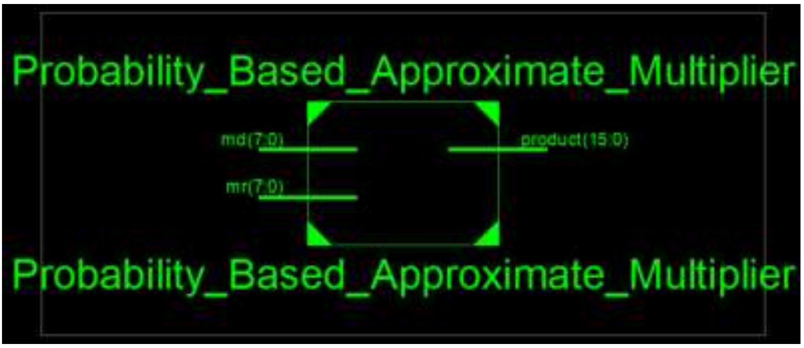

Probability_Based_Approximate_Multiplier
md(7:0)
mr(7:0)
product(15:0)
Probability_Based_Approximate_Multiplier

Fig 7.RTL Schematic Of The Proposed Design

TECHNOLOGY SCHEMATIC:- The representation of the architecture is made in the LUT format by the technology schematic. The LUT is considered to be the parameter o area that is utilised in VLSI for the purpose of estimating the architecture design.the LUT is considered to be a square, and the memory allocation of the code is reflected in the LUTs that are present in the FPGA.

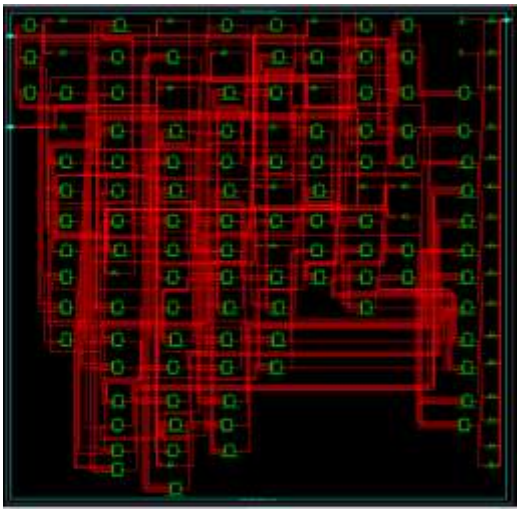


Fig 8.Technology Schematic Of The Prposed Design

SIMULATION:-As opposed to the schematic, which is the verification of the connections and blocks, the simulation is the procedure that is referred to as the final verification in terms of its functioning. During the process of transitioning from the implantation to the simulation on the main page of the tool, the simulation window is activated. The simulation window is responsible for containing the output in the form of wave shapes. For this purpose, it is able to provide a variety of radix number systems, which is a flexible feature.

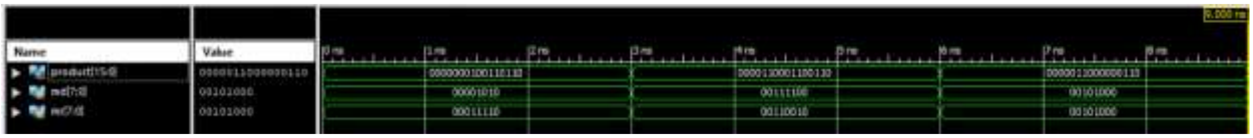


Fig 9.Simaulation wave forms of proposed approximate multiplier

PARAMETERS:-When it comes to very large scale integration (VLSI), the parameters that are considered include area, delay, and power. Based on these criteria, one may evaluate one vs another design. Verilog is the HDL language that is used here, and the tool XILINX 14.7 is used to acquire the parameter. Additionally, the consideration of area power and latency is also taken into account.

PARAMETERS	Existed probability based Approximate multiplier	Proposed probability based Approximate multiplier
No of LUT's	101	98
Delay(ns)	12.548	11.578

TABLE 3. PARAMETERS TABLE

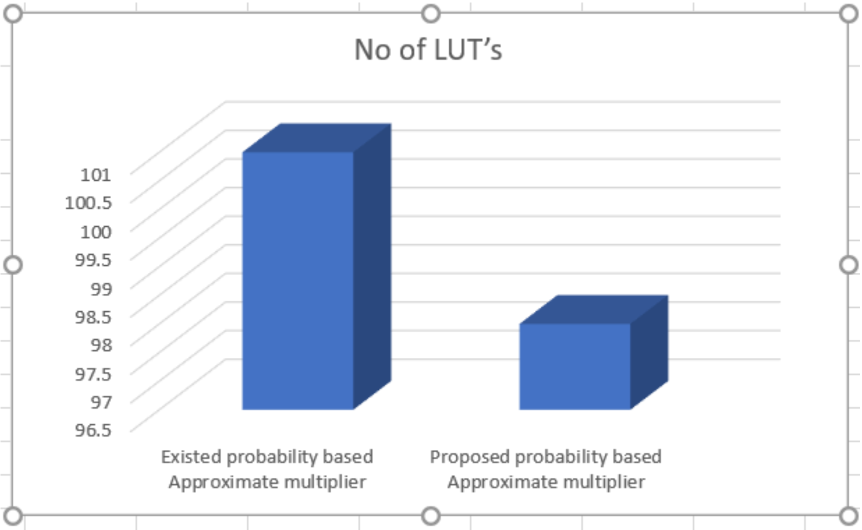


Fig 10. LUT comparison bar graph

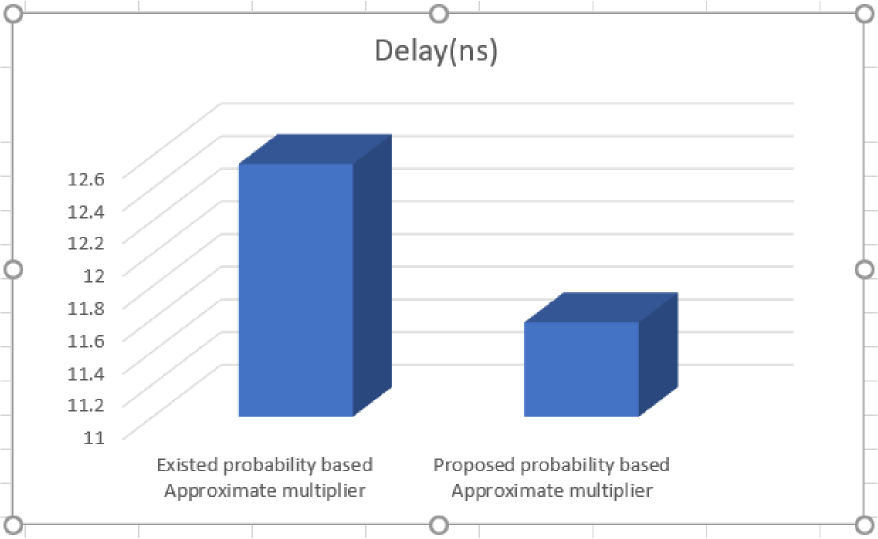


Fig 11. delay comparison bar graph

CONCLUSION

A novel method to approximation 4:2 compressor structures is shown in this research, which also includes an approximate multiplier application. In the first place, a high speed area efficient compressor architecture is suggested. This architecture was able to accomplish a significant decrease in area, delay, and power when compared to previous compressor designs that are considered to be state-of-the-art. There is a comparable level of precision in the suggested design.Consequently, the design that has been presented decreases both the area power and the latency. In order to facilitate image processing applications such as image multiplication and smoothing, the architecture was built to incorporate an 8×8 Dadda multiplier function. When compared to other multipliers that are already in use.In general, it is rather difficult to create an approximation multiplier that brings about absolute advantage, and the solution that is considered to be optimum is often the one that is most suitable for the application that is being targeted. The approximation multiplier architecture that we have developed provides a contender that has a competitive error-electrical performance tradeoff.

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.

[11] M. Ahmadinejad, M. H. Moaiyeri, and F. Sabetzadeh, “Energy and area efficient imprecise compressors for approximate multiplication at nanoscale,” AEU Int. J. Electron. Commun., vol. 110, Oct. 2019, Art. no. 152859.