International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

ENERGY EFFICIENT COMPACT APPROXIMATE MULTIPLIER FOR ERROR RESILIENT APPLICATION

Saba Fatima[1], Samreen Unnisa[2]

[1]PG Scholar, Department of VLSI, Shadan Women's College of Engineering and Technology, Hyderabad, sabamirza121472@gmail.com

[2]Asst. Professor, Department of ECE, Shadan Women's College of Engineering and Technology samreenunnisa23@gmail.com

Received: 03-08-2025 Accepted: 05-09-2025 Published: 12-09-2025

ABSTRACT

Improving system performance, including speed, form factor, and energy economy, is the main objective of approximation computing. Even while approximate multipliers are becoming more and more popular, designing effective approximation compressors, a basic multiplier block, is still quite difficult. Eight-transistor and fourteen-transistor 4:2 compressors are suggested in this short. With fewer negative errors, both compressors take use of CMOS technology and a constant and conditional approximation of specific inputs. Consequently, a resource-intensive error recovery module is removed, resulting in better performance than previous work. By sacrificing less size for more precision, the 14-transistor architecture produces a lower error rate than the 8-transistor architecture. Using image multiplication, the compressor's customized circuit design is also suggested and assessed. In comparison to the precise multiplier, the suggested multiplier shows superior accuracy, 38% PDP improvement, 50% area savings, and a 93% reduction in power-delay-product.

1. INTRODUCTION

One interesting technique for increasing the speed and effectiveness of arithmetic circuits is APPROXIMATE computing [1], [2]. To balance accuracy and circuit performance in error-tolerant applications, such image processing. This novel concept has been applied in several error-resilient domains. By ignoring some inputs and outputs at a fair cost to output accuracy, approximate computing seeks to improve circuit performance. A compressor circuit is an essential part of more intricate circuits such as multipliers [3, 4, 5, 6, 7, 8, 9]. Numerous designs tailored to certain compressor applications have been put forth, each with unique benefits and drawbacks [7], [9]. In contrast to precise compressors, which take into account every input and output, approximate compressors ignore some inputs and outputs [10]. These circuits often perform better than perfect circuits because of significant reductions in circuit complexity, input/output counts, and load capacitance [11], [12]. Two essential components of a multiplier are compressors and full-adders (FAs). Therefore, as part of the partial product reduction tree (PPRT) of multiplier circuits, compact approximation multipliers are usually built employing different compressors and FAs with fewer transistors. Keep in mind that increased power and delays are caused by the wide region of current approximate solutions. Thus, preserving the circuit characteristics and error rate in compressors continues to be a significant difficulty. We suggest a new circuit based on majority logic (ML) to address this urgent problem [7]. The approximation compressor we proposed employs eight transistors within the approximate PPRT subcircuit of an 8-bit multiplier, as opposed to the 12-transistor cell in [7]. We suggest a more precise 14-transistor compressor to lower the error rate of our first compressor. Based on the system-wide performance of an approximation multiplier during picture multiplication, we examine the effectiveness of our suggested circuits. Our circuits eliminate 15 AND gates from the PPRT stage by drastically reducing the number of partial products (PPs) and precise compressors. When compared to an exact multiplier, the size and energy consumption are reduced by 49% and 93%, respectively, because only one exact compressor, comprising 786 transistors, is utilized. According to a thorough examination of current approximate compressors, each of the suggested and cutting-edge compressors has a unique circuit design and Boolean function that trades off accuracy, power, and area in a special way.Enhancing digital circuits' energy efficiency is a key need for modern systems-on-a-chip. Data analysis, multisensory processing of signals, neural networks, and data classification are examples of applications that are error robust [1]. For these types of programs, approximate computing is a helpful method to achieve increases in power, speed, and area efficiency [2]–[4]. The majority of research on hardware-level

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

approximating has focused on numerical units, which are often among the most important energy-intensive digital blocks [11], such as multipliers and adders [5]–[10].To generate extremely efficient approximation multipliers, a number of methods have been presented [12]. The basic In approximation recursion multipliers, 2×2 approximation multiply modules are combined to form n × n multipliers [13]–[15]. The impact of n × n multipliers is approximated in the works [16]–[18] using a small m × m multiplier (where m < n). Ref. [19] uses approximations to produce the partial products by omitting the numerical calculation of some part outcomes of the multiplier. To approximate the outcome, the calculation An approximation to the operands' exponential is obtained by adding the logarithmic multipliers [20], [21], and the antilogarithm. In reduced multipliers, some partial outcomes are not generated [22], [23], and the resulting truncation error is minimized by using the proper correction functions. Runtime approximation configuration [24], approximate Booth encoding [25], approximate redundant binary multipliers [26], and approximation technique collaboration [27] are other approaches.It is now possible to generate approximated multipliers with approximation compressors. The multi-operand sum of the partial products is transformed into a two-operand combination using compressors, sometimes known as "one counters," using tree-based proportional reduction techniques such as Wallace [28], Dadda [29], or the Three-Dimensional Method TDM [30].The most popular compressor, the full-adder, is often referred to as a (3,2) compressor since it converts every input into a count that is encoded in two outputs. The half-adder and more complex compressors (such 4-2 or 5-3) are also commonly utilized in multipliers [31], [32]. Based on the stacking circuit approach, hardware-efficient 6-3 and 7-3 precise compressors are developed in [33].

Approximate multipliers can be created by substituting components of the accurate compressors alongside simpler circuits that result in some errors but offer efficiency improvements in terms of power, speed, and space. Although [35], [36], and [34] use approximated multipliers with just two outputs, [34] provides approximate processors that are made by truncating the outputs of accurate compressors. In the article [37], a 5-3 compressors is used as the basic module to create an approximately 15-4 compressor. The approach in [38] performs lossy compression of the partial product rows using approximation half-adders to get a reduced set of product terms. Simple OR gates are employed as approximations counters in [39]. Ref. [40] presents a novel family of approximation compressors where the outputs of the multipliers have the exact identical weight than their inputs although there are no carry outputs. Two 4 × 4 Using encrypted part products and approximated compressors, approximate multipliers with different accuracies are produced; the proposed 4x4 multipliers are used as building blocks to scale up to larger multipliers. 4-2 compressors are often used to build exact multipliers [42] due to their efficient transistor-level implementation [43]–[45] and simpler wiring. We focus on them in this study because of this. Recently, many approaches have been proposed for the construction of approximating 4-2 exchangers and their application as approximating multipliers [46]–[54]. The number of recommended topologies makes it difficult for the designer who want to employ around 4–2 compressors in a multiplier to select the ideal one. In this paper, we present a detailed examination and comparison of previously proposed approximation 4-2 air compressors, with a focus on the designs meant for use in traditional tree-based multipliers. Using the pyramidal circuit technique, we show how to construct approximate compressors [33], note that several of the already suggested approximated 4-2 compressors may be generated in this way, and provide a new approximate 4-2 compressor. In all, we analyze a total of twelve different approximation 4-2 compressors. 16 × and 8 × The circuits under consideration are used to create 16 multipliers, which are then implemented in 28nm CMOS technology. A hybrid technique is also used, which employs two different approximated 4-2 compressors to lower different parts of the partial product matrix. We consider two signed and unsigned multiplier setups with different approximation levels for all operand sizes. The study highlights that the error outcome for multipliers using approximate 4-2 compressors depends on the exact connection between each partial product and each approximated compressor input. This argument challenges the design of the partial product reduction tree, which has been disregarded in previous publications. Our analysis shows that approximated compressors are effective for constructing unregistered multiplier effects, but they can drastically degrade precision when used in signed multiplier effects, especially in the left-most column of the part product matrix. When the circuits in this investigation are compared, it can

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

be shown that several 4-2 compressors significantly improve electrical performance. (more than 50% decrease in power and delay), while other designs result in a significantly smaller power saving. Our investigation shows that exists no one winning topology when taking the power-accuracy tradeoff into account because the best option depends on the necessary precision, on the taken into inaccuracy metric as well as the multiplier's signedness. We present power vs. accuracy compromise curves that the could possibly help determine the best design.

PROPOSED METHOD

On the basis of majority logic (ML), We recommend anew circuit [7The approximate compressors that we proposed employs eight transistors within the approximate PPRT subcircuit of an 8-bit multiplier, as opposed to the 12-transistor cell in [7]. We suggest a more precise 14-transistor compressor to reduce the rate of errors of our first compressor. Based on the system-wide performance of an approximation multiplier during picture multiplication, we examine the effectiveness of our suggested circuits. Our circuits eliminate 15 AND gates from the PPRT stage by drastically reducing the quantity of incomplete goods (PPs) and precise compressors.

2. LITERATURE SURVEY

Two roughly 4:2 compressors are suggested in [3], [4], with a lower delay and a focus on circuit efficiency rather than precision. In [13], the multiplier shows increasing area and higher mistake rates. In order to achieve high accuracy and area consumption, an approximation compressor in [5] uses a hash table to calculate error distances (ED) and an error recovery module (ERM) to reduce errors. A programmable multiplier with both accurate and approximate modes is presented in [6], displaying a large area and output errors of up to 50%. Circuits based on CMOS [8] and majority logic (ML) [7] drastically cut down on power and area at the cost of a greater error rate. Techniques such as algorithm-based optimization with error control modules [11], probability-based error correction [12], and circuit-stacking-based approximation compressors [9] demonstrate lower error despite needing a greater area.

"**Security in approximate computing and approximate computing for security: Challenges and opportunities**" by W. Liu, C. Gu, M. ÓNeill, G. Qu, F. Lombardi and P. Montuschi

An sophisticated computational method called approximate computing compromises calculation precision in favor of more efficient use of system resources. In many situations where erroneous results are acceptable, it has become the new paradigm that is preferred over conventional computer systems. Nonetheless, There are dangers to security connected to approximate computing as well, mostly because the unpredictable and unknown inherent faults that occur during approximation execution might be mistaken for intentional alteration of the input data, the execution procedure, and the output. However, it's noteworthy to note that approximation computing offers new ways to protect the computation and the system. Although threat models, defenses, and assessments are covered in the work that has already been done on the security of approximation computing, there isn't a framework for comparison and analysis. This page offers a taxonomy of the most recent research in this area, including threat models in approximation computing and prospective approximate computing-based security techniques. There is also discussion of open questions and possible future research avenues.

In their paper "Design and analysis of approximating compression for Multiplication," A. Momeni, J. Han, P. Montuschi, and F. Lombardi

An appealing algorithm for nanometric computer processing is computation that is approximate or imprecise. Inexact computation is particularly interesting for computer mathematical concepts. This project focuses on the design and analysis of two novel, roughly 4-2 compressors for use in a multiplier. Calculation inaccuracies (as shown by the error rate and the so-called standardized error distance) can be satisfied in respect to a design's circuit-based figures of beauty (transistor count, latency in addition to and power consumption) since these designs depend on different compression properties.Four different methods of using the proposed assumption compression are shown and analyzed for a Dadda multiplier. Along with comprehensive simulation results, the application of the approximated multipliers to image processing is demonstrated. The results show that the proposed designs significantly reduce power dissipation, delay, and transistor count compared to an exact design; two of the proposed multiplier designs also provide outstanding image multiplication performance in terms of average standardized error separate and peak ratio of signal to

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

noise (greater than 50 dB for the considered photograph examples).

"Design of power and area efficient approximate multipliers" by S. Venkatachalam and S.-B. Ko

For error-resistant applications, approximate computation can improve performance and power efficiency while reducing design complexity. This brief discusses a novel design method for multiplier approximation. To add other probability terms, the multiplier's partial Devices are changed. The logic complexity of approximate for the accumulation of modified partial products varies according to their probability. The proposed approximation is applied to two variants of 16-bit multipliers. Synthesis results show that two proposed multipliers provide 72% and 38% power reductions, respectively, better than an exact multiplier.They are more accurate than existing approximation multipliers. In comparison to the previous study, the proposed extrapolation multipliers have mean relative error values as low as 7.6% and 0.02%, respectively. In an image processing application used to evaluate the performance of the proposed multipliers, one of the proposed models achieves the greatest peak signal to noise ratio.

"Multipliers with roughly four to two compressors and error recovery modules," by M. Ha and S. Lee

A typical operation in approximation High-speed computer methods and low power computing is approximate multiplication. Using an approximate 4-2 air compressor, one may produce power-efficient circuits for approximate multiplication. This letter introduces a new design that incorporates an error recovery module and is based on a modification of an earlier roughly 4-2 compressor concept. Compared to earlier suggested 4-2 compressor-based approximate multiplier solutions, the suggested design is more accurate, uses less hardware, and uses less power—even with the added error recovery module.

M. Kamal, A. Afzali-Kusha, M. Pedram, and O. Akbari, "Dual-quality 4:2 compressors for use in dynamic accuracy configurable multipliers"

The four 4:2 compressors that we provide in this study can be switched between precise and approximate working modes. These dual-quality compressors offer less precision at the expense of faster speeds and lower power consumption in the approximate mode. In addition to varying delays and power dissipations in the approximate and accurate modes, each of these compressors has a unique degree of precision in the approximate mode. By incorporating these compressors into parallel multiplier structures, it is possible to create programmable multipliers with dynamically changing accuracies, powers, and speeds throughout the runtime. In a 45-nm standard CMOS technology, the efficiency of these compressors in a 32-bit Dadda multiplier is assessed by contrasting their properties with those of the most advanced approximate multipliers. Comparison results show that the approximation mode has an average reduction in delay and power usage of 46% and 68%, respectively. Additionally, certain image processing apps evaluate these compressors' efficacy.

3. PROPOSED METHODOLOGY

3.1 Approach

A strong and fault-tolerant digital logic design paradigm, majority logic (ML) uses the strength of group decision-making to improve electronic circuit efficiency, simplicity, and dependability. When more than half of its inputs are True (False), an ML gate produces a True (False) output using the odd input logic concept. More configurability means that ML-based circuits usually need fewer gates. Therefore, a possible method for reducing circuit area without compromising precision is the creation of ML-based compressors. Our design path for an 8-bit multiplier that generates a mix of fixed, approximate, and accurate outputs is depicted in Fig. 1. The estimated bits are created by the ML-based approximate components (see Sections III-A and III-B) and sent to the simplified circuits that include half adders (HAs) and full adders (FAs) with a minimum of one fixed input.

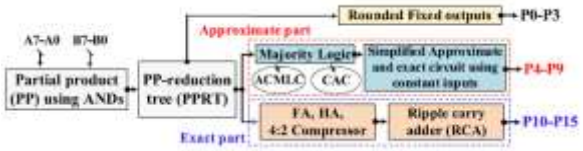


Fig. 1. Proposed design flow of 8-bit multiplier

A. Condition-Based Approximate ML Compressor at 4:2

A two-stage approximation method is used to enhance and use the ML circuit design that was detailed in [7]. The circuit schematics are displayed in Fig. 2, which also includes the cell view of the exact compressor circuit (c.f., Fig. 2(a)), the ML-based compressor (c.f., Fig. 2(b), [7]), and the compensation approximate compressor (CAC) and suggested approximate condition-based ML (ACMLC) compressor for the ACMLC (c.f., Fig. 2(c)). A predetermined output (Sum = VDD) and only three of the four inputs are taken into account by the ML-based compressor [7]

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

(input X2 is omitted). If at least two of the inputs are "1" (X1 + X3 + X4 = 2 or 3), the carry output is True, resulting in four negative and four positive mistakes overall. The effects of positive and negative faults in approximation circuits have been extensively studied recently [14]. Each approximation compressor creates a mistake in the approximate area of a multiplier, resulting in a significant error distance in the columns. For individual approximation compressors, using an error recovery module (ERM) based on particular input patterns is essential for lowering negative errors and the total error distance. As a result, it has been highlighted how crucial ERM circuits are for reducing negative mistakes [5, 11, 12]. The need for ERM is removed by using a multiplier that wisely employs approximation compressors because of their constant error production and low negative error rates.

Therefore, in order to simultaneously limit negative error and reduce the circuit size, we suggest the ACMLC circuit architecture, as seen in Fig. 2(c). In addition to the usual majority-based approximation, which ignores one of the three inputs, X4, and sets Sum to VDD, the Carry output is "1" in the two situations listed below: (i) as indicated in blue in Fig. 2(c), two or three of the inputs are "1" (like [7]); or (ii) as indicated in red in Fig. 2(c), just X2 is "1" (and the other inputs are "0"). All other input combinations result in SumACMLC = VDD and CarryACMLC = (X1•X3)+X2, as the Carry output is "0". The Gathering Place of Truths for the ACMLC 4:2 compressor is displayed in Table I, and it yields a total of 9 mistakes (only one extra compared to [7]), of which only three mistakes are negative (as opposed to four negative errors of [7]) and only one output shows an error distance of two (ED = 2).Additionally, our layout reduces the circuit footprint and significantly simplifies the Boolean functionality. A compensator circuit is required to allow data to flow between the approximate and accurate sub-circuits of a multiplier in compressor topologies with a greater number of negative faults (see [5], [11], [12]). We suggest the CAC circuit architecture (c.f., Fig. 2(c)) for ACMLC compensation in order to overcome this difficulty. The design is based on a standard 4:2 compressor architecture, with an estimated sum of Sum = (X1 • X2) (X3 + X4) + (X3 • X4) and a fixed carry output of Carry = VDD. With this method, there is only one negative error and seven errors (all with an error distance of ED = 1). As an alternative, there are 14 transistors in CAC instead of the 8 in ACMLC. Lower area is a natural result of the proposed ACMLC compressor's simplicity when compared to the ML-based compressor [7] and CAC. Despite having two more transistors than [7], CAC gains a lot by having fewer negative faults. We suggest an effective multiplier circuit with excellent accuracy and performance, based on the special characteristics of the ACMLC and CAC.

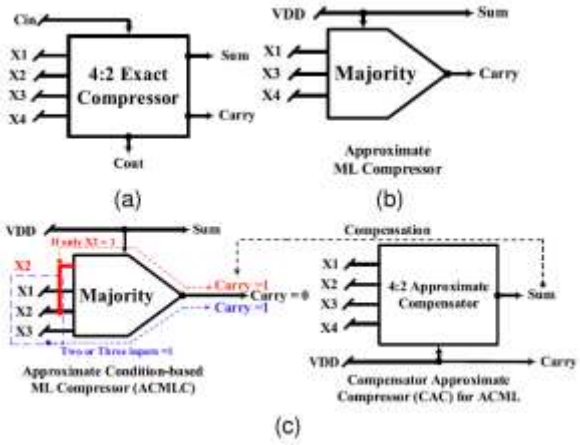


Figure 2: Compressor circuit schematics (a): Specifically, (b): ML-based, and (c): CAC and ACMLC, in that order.

B. An approximate multiplication of 8 bits

Using the suggested ACMLC-based compressor and CAC, we display the multiplier's architecture in Figure 3. Each of the three parts of our suggested approximate multiplier—truncation, constant truncation, approximation, and precise compute—contributes to the partial product (PP) computation in Stage 1 (c.f., Fig. 3). In these circuits, truncating at least four LSB columns of PPs (for an 8-bit multiplier) is a typical technique. This method works well with extremely precise approximations. Based on [13], an alternate approach is favored in this brief in order to reduce a greater number of mistakes using the suggested compressors. A number of LSB PP bits are given fixed values in [13]. By applying various inputs, the chances of producing 0 or 1 for each of the circuits producing p0, p1, p2, p3, and p4 were examined. The four LSB bits on the right are fixed at p0p1p2p3=0110, according to observations. There are fewer gates needed when the LSB bits are fixed to a fixed value. In this instance, as indicated by the gray triangular shaded area in Fig. 3, 10 AND gates are eliminated from the first four columns of the suggested multiplier circuit.

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

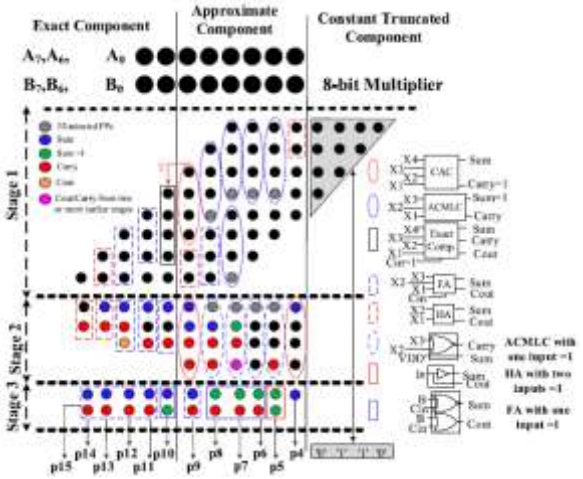


Figure 3: Suggested multiplier schematic utilizing the suggested CAC and ACMLC-based compressor.

A CAC, two precise HAs, an exact FA, and five ACMLC-based compressors make up the approximate component of the suggested multiplier. Five AND gates are removed inside a single PP stage with the suggested 3-input ACMLC-based compressor since one of the four PPs is essentially abolished. The second stage reduces the overall size by reusing the eight unused (solid black circles) AND gates from that stage to create a compressor-based chain.

The precise compressor circuit is made simpler by transferring Carry = 1 to the exact component using the suggested CAC in the final column of the approximation component. Lastly, the entire multiplier just makes use of the simplified exact compressor [15] (with constant input, Cin = 1). It should be noted that the precise HAs and FA improve multiplier precision, which balances the compact size of the suggested ACMLC-based compressor and CAC. The function of the circuit is re-examined below.CAC makes up the approximate component's first and last columns in Stage 2. Higher compression accuracy is the effect of using more AND gates.In Stage 2, when input X4, shown by gray circles in Figure 3, is disregarded, an extra four ACMLC-based compressors can be used thanks to the suggested ACMLC-based Stage 1 design. Consequently, 15 of the 64 AND gates—which are necessary for the 8-bit multiplier—are eliminated, leaving 49 gates needed for the suggested design. Furthermore, Stage 2 employs three exact FAs and two exact HAs. The last addition level receives the carry output from the Stage 2 CAC.A significant number of HAs and FAs are needed for Stage 3, which usually employs ripple carry adders (RCAs) with variable bit lengths. This number can be considerably decreased by using an ACMLC-based compressor and a CAC with Sum = 1 and Carry = 1, respectively. As a result, the approximation component in the last stage uses a HA that acts as an inverter and has two constant inputs of "1." Moreover, an XOR and an OR may be used to build the three FAs with a constant input of "1" (as seen in Fig. 3). It seems sense that our architecture's overall simplicity, as compared to an accurate multiplier or the ML-based multiplier [7], would result in notable power, area, and delay reductions. The suggested multiplier, which consists of 786 transistors, is 12.6% smaller than the 900-transistor ML-based multiplier [7] and 49% smaller than the 1530-transistor exact multiplier [15].

3.2 4-2 COMPRESSORS

A. Accurate Compressor

Since an exact 4-2 compressor has three outputs (S, C, and Tout) and five inputs (x1, x2, x3, x4, Tin) of equal weight, it is more appropriate to refer to it as a (5,3) counter. Whereas C and Tout have double the weight of the inputs, the S output has the same weight. In tree multipliers, the Tout generated by a 4-2 compressor in the i-th column is connected to the Tin of a compressor in column i+1, without affecting delay, because the compressor is built to not rely on Tin. A typical 4-2 compressor implementation with two complete adders is seen in Fig. 1; more effective designs are suggested in [43]–[45].

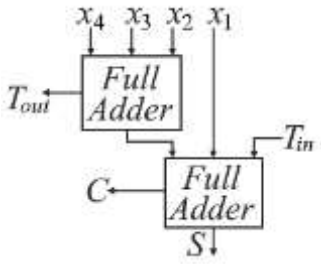


Figure 1 depicts the application of an identical 4-2 compressor using two full-adders.

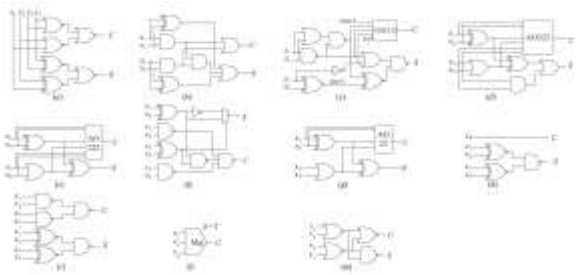
B. Earlier Approximate 4-2 Compressor Proposals

To make wiring and circuit implementation simpler, the Tin and Tout pins are not utilized in the majority of previously suggested approximate 4-2 compressors (with a few outliers, [55]). Three is the highest value that may encoded using just the S and C outputs. With four inputs (x1,..x4), it is evident that at least one inaccuracy is inevitable when every input are "1." The schematic of the approximate 4-2 compressor suggested in [46] is displayed in Fig. 2(a), while Fig.

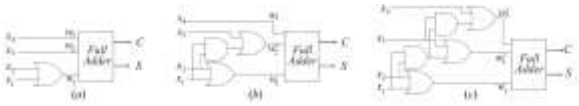
International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

2(b) displays the approximate compressor suggested in [47]. In order to achieve a simpler logic implementation, these circuits—referred to as "Momeni" and "Venka" in the following—introduce faults into the truth-table of the precise 4-2 compressor.

In [48], three roughly 4-2 compressors are described; they will be referred to as "Yang1," "Yang2," and "Yang3" in the rest of this article. Figure 2(c)–(e) displays the compressor schematics. With a single inaccuracy in the truth table, the Yang1 compressor in Fig. 2(c) is the most accurate form; when all inputs are "1," the output is C = 1, S = 1. This behavior is consistent with the "saturating counter" theory put forward in [49].



In order to simplify the S output circuitry, the Yang2 and Yang3 circuits in Fig. 2(d-e) introduce further faults into the compressor's truth table. The design of the roughly 4-2 compressor suggested in [50] is seen in Fig. 2(f); it will be referred to as the "Lin" compressor from now on. When all inputs are "1," as in the Yang1 compressor, an incorrect result is obtained. In this instance, the outputs are C = 1, S = 0, which causes a two-fold discrepancy between the number of inputs that are "1" and the estimated count value that the compressor calculates. The Lin compressor's behavior is consistent with the "reflecting 4:2 counter" that was suggested in [49]. The estimated compressor's C output from Fig. 2(e) is changed in [51]; As shown in Fig. 2(g), the suggested design—referred to as the "Ha" compressor from here on—can be put into practice. Like the Yang3 compressor, the Ha compressor's truth table has four imprecise entries, but it always underestimates the precise outcome. [52] presents dual-quality 4-2 compressors that can be switched between precise and approximate operation modes. The approximate portion of the compressors in [52] that will be referred to as "Akbari1" and "Akbari2" is seen in Fig. 2(h-i). In [53], a basic approximation compressor is suggested. The circuit, referred to as "Sabetz" in the following, is seen in Fig. 2(l), which is a majority gate, assuming that S output is constant and equal to "1" and does not employ the x2 input. As seen in Fig. 2(m), the compressor in [54], known as "Ahma," is likewise incredibly hardware efficient, using just three NOR and one NAND gate.



C. Suggested About a 4-to-2 compressor

The stacking circuit approach, first presented in [33] to create hardware-efficient 6-3 and 7-3 exact compressors, is modified in the proposed approximation 4-2 compressor to produce approximate compressors. A 4-bit stacker circuit with four inputs (x1, x2, x3, x4) has four outputs (y1, y2, y3, y4). If any one of the inputs is "1," y1 will be high; if any two of the inputs are "1," y2 will be high; and so on. The following Boolean equations describe a four-bit stacker:

$y1 = x1 + x2 + x3 + x4$ (1)

$y2 = x1x2 + x1x3 + x1x4 + x2x3 + x2x4 + x3x4$ (2)

$y3 = x1x2x3 + x1x2x4 + x1x3x4 + x2x3x4$ (3)

$y4 = x1x2x3x4$ (4)

Instead of counting the xi while building a compressor, we may count the number of yi that are "1" since it is obvious that:

$$\sum_{i=1}^{4} x_i = \sum_{i=1}^{4} y_i$$

The counting of the Boolean terms that are "1" is indicated by the summation here.

Assuming the inputs x1..x4 are independent of one another, we may express their probability of being "1" as p. with example, with p = 0.5, the probabilities of the yi terms are P(y1) = 0.9375, P(y2) = 0.6875, P(y3) = 0.3125, and P(y1) = 0.0625. The analysis of (1)–(4) shows that y1 has the highest likelihood of being "1," followed by y2, y3, and y4, which has the lowest probability. We must ignore one of the yi terms in (1)–(4) as the maximum count value of an approximation 4-2 compressor is three. According to the discussion above, ignoring y4 is the option that reduces the mistake likelihood. Now, we want to get three new Boolean functions (w1, w2, w3) that are easier to understand than (1)–(3). By making consecutive approximations, we go forward. The first phase involves building the wi as simply as feasible while covering all situations when y1 is "1" (keep in mind that y1 is the word with the highest probability in (1)–(3)). One potential remedy is

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

$$w_1 = x_1 + x_2$$

$$w_2 = x_3$$

$$w_3 = x_4$$

All cases in which y1 = "1_" and all cases in which y2 = "1_" are covered by counting the number of wi that are "1" in (5)–(7), Apart From the situation in which x1x2 = "1_" (keep in mind that y2 is the term in (1)–(3) with the greater probability, after y1). Furthermore, (5)–(7) also cover two of the four situations in which y3 is '1'. Thus, as seen in Fig. 3(a), an approximate 4-2 compressor may be created by counting the number wi that are "1" in (5)-(7) with the use of a full-adder; the truth table of this circuit corresponds to the Ha compressor, [51]. By adding the term x1x2 required to cover all the situations in which y2 is high to (5)–(7), a more accurate compressor may be produced. This may be accomplished by making the following changes to W2:

$$w_2' = x_3 + x_1x_2$$

With the exception of the situation where x1x2x3 = "1_," the approximate 4-2 compressor derived from (5), (7), and (8) has never been Shown before and covers all scenarios in which y3 is high. In the following, the analogous circuit, depicted in Fig. 3(b), shall be referred to as the "Proposed" compressor.

Adding the phrase x1x2x3 to w3 will ultimately result in an even more precise circuit:

$$w_3' = x_4 + x_1x_2x_3$$

The condition is satisfied by equations (5), (8), and (9).

$$\sum_{i=1}^{3} w_i = \sum_{i=1}^{3} y_i$$

Fig. 3(c) displays the analogous circuit, which shares the Yang1 compressor's truth table.

D. Estimated Compressor Properties

The truth tables for the roughly 4-2 compressors shown in Figures 2 and 3(b) are reported in Table I. The truth tables' incorrect values are indicated in bold. The error, which is the difference between the number of inputs "1" and the estimated count value determined by compressors, is reported in the columns marked with the letter E. The truth table of the Yang1 (Fig. 2(c)) and Lin (Fig. 2(f)) compressors has a single inaccuracy. While the other approximate compressors have a greater number of mistakes, the Yang2 compressor (Fig. Both the suggested circuit (Fig. 3(b)) and 2(d)) show two erroneous entries.The error situation emerges for the Yang3, Ha, and Proposed compressors when x3x4 = '1_,' although the error occurs for x1x2x3 = '1_ in the Designed circuit. In other compressors that produce air the errors are dispersed more equally over the table.

3.3 FUNCTION AND DESIGN OF MULTIPLIER

Using the approximately 4-2 compressors, we construct n × n multipliers with n = 8 and n = 16, accounting for two different arrangements per multiplier: i) The partial-product C-N only uses around 4–2 crushers. matrix's (PPM) n less important columns. tries to reduce the mistakes. ii) C-FULL: employs around 4–2 compressors throughout the PPM. The goal of this more forceful strategy is to reduce power dissipation.

A. Reduction of Products in Parts

By employing a technique similar to the Dadda multiplier, which uses several compressor stages to lower the PPM's maximum height to two rows, the PPM is decreased. The last stage is worked back to determine the maximum height. Because 4-2 compressors are used, the maximum heights for the different stages are 2, 4, 8, 16, 32, …. Only once they're done completely necessary to achieve the desired PPM reduction are Employed are full & half addition.

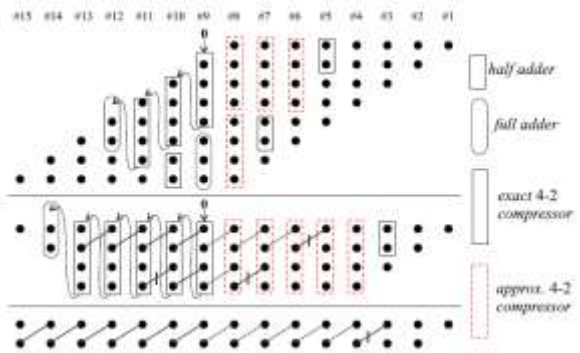
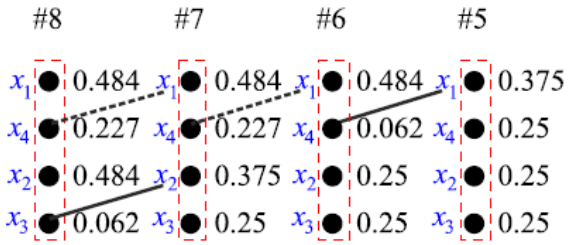


Figure 4: Reduction plan for an 8 x 8 unsigned multiplier using a C-N arrangement

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper



For example, Fig. 5 shows columns 5-8 of the PPM of Fig. 4 using the Ha compressor, where the component products of the subsequent stage are identified with their possibility. As can be observed in the same Fig. 5, the two partial results with a reduced likelihood should drive x3 and x4 in order to lessen the mistake probability. This is due to the fact that x3x4 = '_1' causes an error in the Ha circuit. The scenario is substantially different when Yang2 or Yang3 proximity filters are utilized. The error circumstances for Yang2 compressors are a subset of the Ha compressor's, whereas the incorrect requirements for Yang3 valves are identical to the Ha compressor's, as seen in Fig. 5.

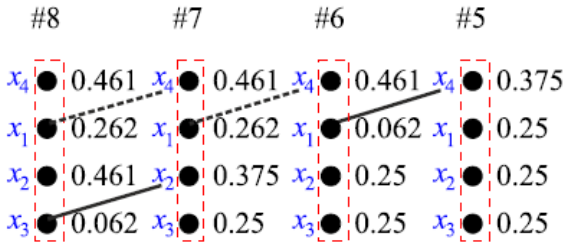
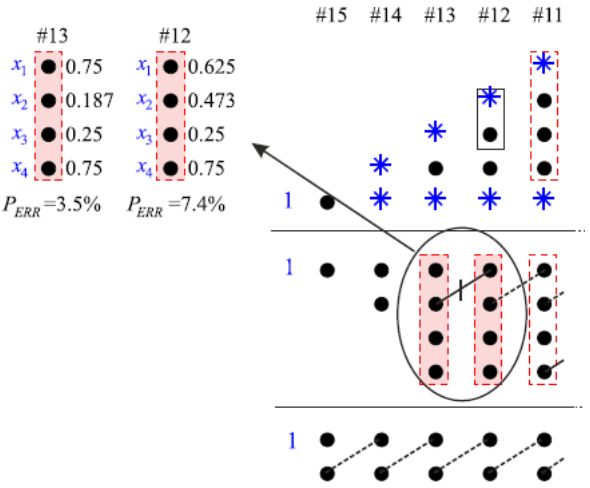


Figure 6. The portion of the product for the second step in the multiplier in Figure 4 is displayed using the hypothetical 4-2 compressor proposed in this study.

The rough compressor proposed in this work has an inaccurate constraint x1x2 x3 = '1_,' therefore to lower the erroneous probability, x4 should be the value driven by the portion of the product with a greater probability of occurring. This leads to a clear layout, as seen in Fig. 6. A possible input arrangement that reduces the error probability for the roughly 4-2 compressors is shown in the picture. Finding the optimal correlation between closeness compressed inputs & the partial results for the other 4–2 turbines that lowers the error probability because the truth table (see Table I) has mistakes dispersed across it. Fortunately, this error condition dispersal also reduces the impact of each signal's unique relationship to each approximation compressor input on the multiplier's overall performance. After experimenting with several setups, we discovered that the same strategy used for the Ha, Yang2, and Yang3 compressors also yields the best results for the other circuits under investigation. The suggested method may also be used to reduce partial products in signed multipliers. However, in this instance, The architect ought to consider that supplemented partial products, which show up in the PPM of signed multipliers, have a 3/4 chance of being "1." Because there is a greater chance of incomplete products, mistakes are more likely to occur. To better illustrate this phenomenon, Fig. 7 shows a part (left-most columns) of the PPM of an 8 × 8 signed multiplier, C-FULL configuration. The star icons stand for the supplemented partial goods. The graphic shows the probability of the inputs from the two approximately 4-2 compressors in columns #12 and #13. These two pumps have an error probability of about 11% when using the recommended circuit. As can be seen in the following (see Tab. II), this value is larger than the error probability of the full 8×8 unsigned multiplier C-N setup and near to the wrong possibility of the 8×8 C-FULL unsigned multiplier. Since the likelihood of component products across every topologies causes the error rate to rise noticeably, all other approach compressors exhibit comparable behavior.

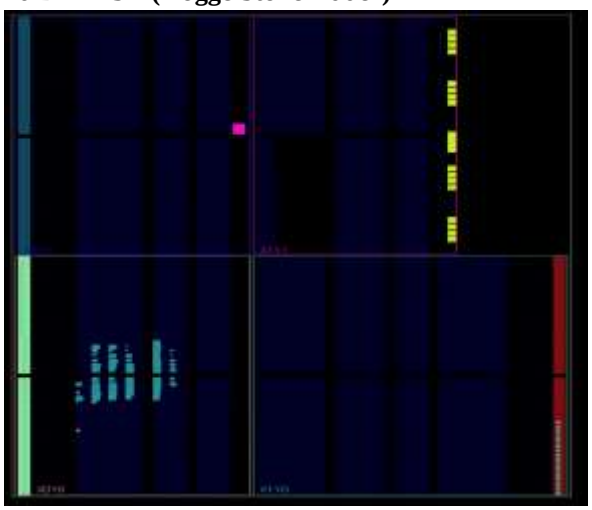


The left-most columns of the PPM for an 8x8 unsigned multiplied in the C-FULL configuration are seen in Figure 7. The star symbols represent the augmented partial products. The input likelihood of the two approx 4-2 converters is shown in columns #12 and #13 (the recommended approx 4-2 compressor is intended to be utilized).

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

5. SIMULATION RESULTS:

16 BIT KSA (Kogge Stone Adder)



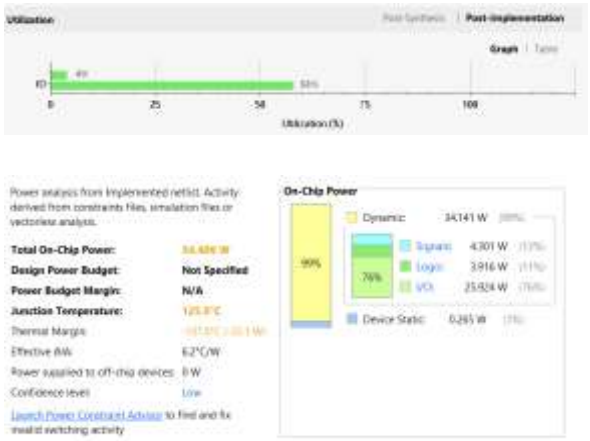


FIGURE: RUN SYSTHESIS

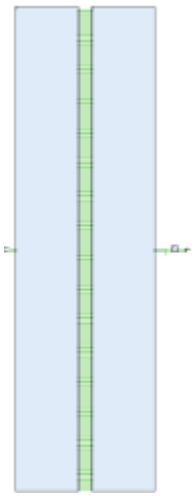


FIGURE: RTL ANALYSIS SCHEMATIC

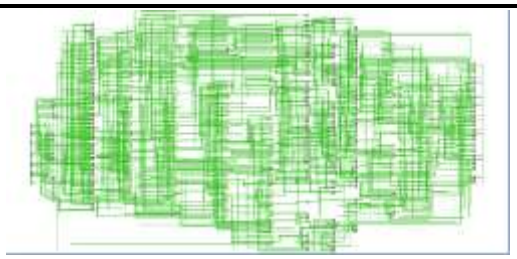


FIGURE: IMPLEMENTATION SCHEMATIC

FIGURE: FINAL OUTPUT FOR 16-BIT KSA

16 BIT RCA (Ripple Carry Adder)

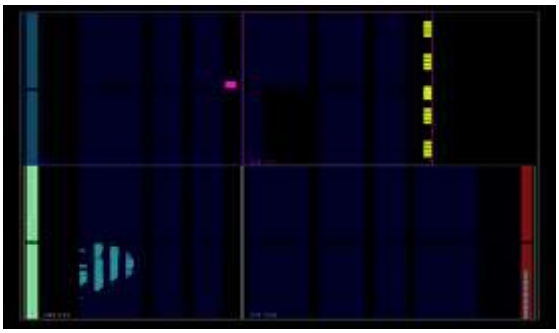


FIGURE: RUN SYSTHESIS

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

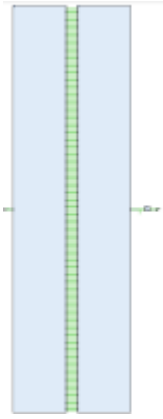


FIGURE: RTL ANALYSIS SCHEMATIC

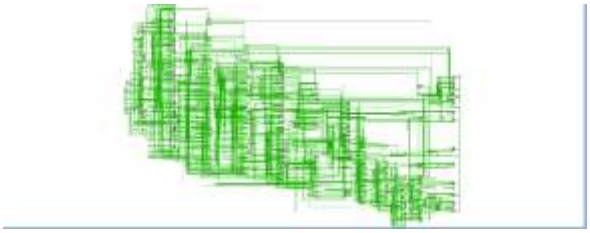


FIGURE: IMPLEMENTATION SCHEMATIC

FIGURE: FINAL OUTPUT FOR 16-BIT RCA

6. CONCLUSION

In this short, we suggested a 14-transistor CAC, an approximate 8-bit multiplier, and an 8-transistor ACMLC compressor for precise and effective image multiplication. The compressor's low power consumption and compact size come at the cost of a comparatively high mistake rate. We suggest CAC, which shows seven mistakes—only one of which is negative—to offset the negative errors. To take use of the special qualities of the suggested compressors, we suggest an approximate multiplier based on ACMLC/CAC. The suggested multiplier shows 93% power savings and 50% area reduction compared to an exact multiplication. When compared to the most advanced approximation multipliers, the suggested multiplier performs better on the majority of assessed parameters. The Pareto findings show that ML-based suggested circuits show promise for low-power and energy dissipation applications, even though their accuracy is lower.

REFERENCES

[1] W. Liu, C. Gu, M. ÓNeill, G. Qu, P. Montuschi, and F. Lombardi, “Security in approximate computing and approximate computing for security: Challenges and opportunities,” Proc. IEEE, vol. 108, no. 12, pp. 2214–2231, Dec. 2020.

[2] W. Liu, F. Lombardi, and M. Schulte, “Approximate computing: From circuits to applications [scanning the issue],” Proc. IEEE, vol. 108, no. 12, pp. 2103–2107, Dec. 2020

[3] 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.

[4] S. Venkatachalam and S.-B. Ko, “Design of power and area efficient approximate multipliers,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 25, no. 5, pp. 1782–1786,
May 2017.

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

[6] O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, “Dualquality 4:2 compressors for utilizing in dynamic accuracy configurable multipliers,” IEEE Trans. Very Large-Scale Integr. Syst. (TVLSI), vol. 25, no. 4, pp. 1352–1361, Apr. 2017.

[7] 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.

[8] 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.

[9] 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.

[10] N. Shiri, A. Sadeghi, and M. Rafiee, “High-efficient and error-resilient gate diffusion input-based approximate full adders for complex multistage rapid structures,” Comput. Electr. Eng., vol. 109, Jul. 2023, Art. no. 108776.

[11] U. Anil Kumar, S. V. Bharadwaj, A. B. Pattaje, S. Nambi, and S. E. Ahmed, “CAAM: Compressor-based adaptive approximate multiplier for neural network applications,” IEEE Embed. Syst. Lett., vol. 15, no. 3, pp. 117–120, Sep. 2023.

[12] M. Zhang, S. Nishizawa, and S. Kimura, “Area efficient approximate 4-2 compressor and probability-

based error adjustment for approximate multiplier," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 70, no. 5, pp. 1714–1718, May 2023.

[13] F. Sabetzadeh, M. H. Moaiyeri, and M. Ahmadinejad, "An ultra-efficient approximate multiplier with error compensation for error-resilient applications," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 70, no. 2, pp. 776–780, Feb. 2023.

[14] L. Sayadi, S. Timarchi, and A. Sheikh-Akbari, "Two efficient approximate unsigned multipliers by developing new configuration for approximate 4:2 compressors," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 70, no. 4, pp. 1649–1659, Apr. 2023.

[15] A. Arasteh, M.H. Moaiyeri, M. Taheri, K. Navi, and N. Bagherzadeh, "An energy and area efficient 4:2 compressor based on FinFETs," Integration, vol. 60, pp. 224–231, Jan. 2018.

[16] E. Zacharelos, I. Nunziata, G. Saggese, A. G. M. Strollo, and E. Napoli, "Approximate recursive multipliers using low power building blocks," IEEE Trans. Emerg. Topics Comput., vol. 10, no. 3, pp. 1315–1330, Jul.–Sep. 2022.

[17] Y. Zhu, Z. Jia, J. Yang, and N. K. Kasabov, "Change detection in multitemporal monitoring images under low illumination," IEEE Access, vol. 8, pp. 126700–126712, 2020.