# Design and FPGA Implementation of High Throughput and Low Latency Machine Learning based Approximate Multiplier for Image Processing Applications

### ANIL KUMAR D.

ECE department, BMS Institute of Technology and Management, Bengaluru, INDIA

Abstract:- One of the uses of approximate circuits is machine learning (ML) and with the help of inexact logic minimization as well as through probabilistic pruning, these approximate computing circuits can be implemented. Nowadays, these approximate circuits have been widely explored due to their essential factors such as compact silicon areas well as low power consumption in movable devices. This research work shows how a 4:2 compressor can be designed using inexact logic minimization and thereby reversing a few bits of the output to ensure efficiency as well as accuracy. The average area, propagation delay as well as the average power of the proposed 4:2 compressor is been calculated and are employed in the  $8 \times 8$  and  $16 \times 16$  Dadda multiplier and truncation and rounding-based scalable approximate multiplier (TOSAM). Using Vivado Design Software Systems in 45nm technology, all the simulations were carried out and the MATLAB tool make use of error analysis to distinguish between precise as well as approximate proposed circuits. This work is mainly concentrated on the design of exact and approximate multipliers and measures the error between them and minimization of this error using the Machine Learning approach and finally validated the results on the Artix-7 FPGA development board of part XCA7CSG324\_110t, the partial products which are generated by multipliers are added using 4:2 compressor adder. In the case of digital processing at nano-metric scales, approximate or inexact computing is considered one of the important examples. For computer arithmetic designs, inexact computation plays a significant role, and the new approximate 4:2 compressors are used in a multiplier that is based on TOSAM. These architectures mainly depend on various compression aspects to enable inaccuracy in computing which is described as error rate and is also referred to as normalized error distance which is used to satisfy circuit-based figures of merit, the number of transistors, delay as well as consumption of power. For a Dadda multiplier, four distinct approaches for exploiting the suggested approximation compressors are designed as well as evaluated. The usage of approximate multipliers for image processing, as well as a wide range of simulation results, are presented in this work. When contrasted to an accurate design, the proposed designs achieve a substantial reduction in the number of transistors, power dissipation as well as delay. Furthermore, the presented multiplier models exhibit outstanding image multiplication capabilities in terms of average normalized error distance as well as peak signal-to-noise ratio that is more than 50dB for the analyzed image samples. The proposed ML-based digital system has been developed in Vivado Design Suite and synthesized which is designed using Verilog HDL. Based on obtained results, 17% reduction in power, 21% reduction in latency, and 33% improvement in throughput.

Key-words: Machine learning, 4:2 Compressor based adder, Dadda Multiplier, TOSAM and FPGA

Received: June 21, 2021. Revised: April 22, 2022. Accepted: May 21, 2022. Published: July 5, 2022.

## **1** Introduction

In multiplier designs, compressors have been considered the major elements for partial product reduction. Previously, various adders were utilized for partial product reduction and one of such adders is carry-save adders. However, these adders were substituted by different order compressors like 3:2, 4:2, as well as 5:2 because of their low power requirement as well as their compact structure. Compact size, as well as reduced power consumption, are the essential factors that are considered for the development of portable devices application. As a result, approximate computation must be used in digital systems to achieve the required power. Approximate computing has received considerable attention whenever precision is

not important in applications such as signal as well as video processing hardware, accelerators that are used in a specific application, machine learning, and so on. 4:2 compressors that consist of XOR-XNOR circuits, as well as transmission gate multiplexers with complemented as well as un-complemented outputs, were presented by Veeramachaneni et al [1]. The research work in [2] illustrates various XOR-XNOR circuits with varying numbers of transistors and each of these transistors overcomes the drawback of another. The suggested XOR-XNOR circuit is being used in compressors such as 4:2 as well as 5:2. The work in [3] utilized a transmission gate multiplexer along with CMOS logic as well as a buffer to increase the current-carrying capacity that is to be used in a 4:2 compressor. The idea of approximate computing has been developed in VLSI architecture because of the increase in the application of portable devices as well as the reduction in power consumption and supply voltage. Many of the applications make use of approximate computing which is illustrated in [4]. Approximate compressors were introduced in [5] and were used in the Discrete Cosine Transform. These compressors used spintronic devices which include magnetic domain wall motion stripe as well as magnetic channel junction. Probabilistic pruning, as well as inexact logic minimization, are the two approximate functional equivalence relaxation approaches introduced by Avinash et al., and further works are presented in [6]. The work proposed in [7] produced only 50% of the output with approximate compressors when contrasted to the inputs. Whereas, the output weights produced were the same as the input weights with the proposed approximate compressor. The research work suggested in [8] illustrates two 4:2 compressors with design1 and design2 showing 20 as well as 12 correct outputs out of 32 outputs and 16 outputs respectively. Dadda multipliers employed this type of compressor and error analysis was carried out by using the applications of image processing.

Error distance (ED), mean error distance (MED), as well as normalized error distances (NED), are some of the error metrics that are presented in [9-10] and these metrics can be employed in the compressor as well as multiplier error analysis. An inaccurate multiplier with reduced performance measures such as area, time as well as power can achieve desired precision by making use of a tool which is presented in [11] wherein a user may specify the maximum

error rate that the circuit can tolerate. The work presented in [12] describes the use of machine learning on existing approaches of CAD to develop unique devices. In [13], an approach for automatically synthesizing approximate circuits using Verilog code was presented, which was later turned into a tool called ABACUS. A mechanism for automatically generating approximate circuits was introduced in [14] and was accomplished through the use of a unit in the ASIC design process, which resulted in significant power, area, as well as latency reductions. The majority of applications of computer arithmetic are accomplished using digital logic circuits which offer high-level accuracy as well as reliability. Multimedia, as well as image processing, are some of the applications that can tolerate computation errors and inaccuracies and thereby generating significant as well as effective output. Algorithms and techniques that are accurate as well as reliable are often not desirable or effective for using these applications. Consider an instance that whenever developing an energy-efficient system, the inexact computation approach mainly concentrates on reducing accurate as well as fully predictable structures. This enables approximate computation to reconfigure the current design process of digital circuits, systems and thereby resulting in reduced complexity, and cost, as well as a probable increase in power efficiency and performance. When contrasted to accurate (precise) logic circuits, approximate or imprecise computing mainly depends on the fact to develop approximate circuits that operate at high performance and/or reduced power consumption [1]. In computer arithmetic, operations such as addition as well as multiplication are commonly employed. In the case of approximate computation, full-adder cells have been extensively examined for addition operation [2-4]. The research work in [1] examined various adders and also introduced several new models for measuring approximate as well as probabilistic adders in terms of integrated figures of merit for inexact computing applications. By taking into account the averaging impact of several inputs and the normalization of multiple-bit adders, the mean error distance (MED, as well as normalized error distance (NED), were presented. Since this normalized error distance is generally constant with size, it can be used to examine the reliability of a certain model. In the work presented in [1] the difference between accuracy as well as power has been critically

examined. The architecture of approximate multipliers on the other hand has received minimal consideration. Multiplication can be performed as the repetitive addition of partial products. Furthermore, the usage of approximate adders to develop an approximate multiplier is not feasible due to inefficiency in terms of accuracy, computational complexity, as well as other performance measures. In the research work of [4] [5] [6] [7], various approximate multipliers have been presented. The majority of these approaches employ the truncated multiplication technique in which the least significant columns of partial products are approximated as a constant. In the case of neural network applications, an approximate array multiplier is being used and is presented in [4] where some of the least significant bits in partial products are excluded and thereby eliminate some of the adders in an array. The work in [5] introduces a truncated multiplier along with a correction constant. To minimize error distance, the correction constant n+k bits are chosen which is found to be as close as possible to the estimated sum of these errors. Whenever the partial products are all 1's or all 0's in the n-k least significant columns then a truncated multiplier with constant correction will be having a maximum error. In the work presented in [6,] a variable correction truncated multiplier is been introduced which is based on column n-k-1 and this technique modifies the corrective term. The correction term is increased as well as decreased only when all the partial products in column-k-1 are one and zero respectively.

To develop large multiplier arrays, a reduced and imprecise 2x2 multiplier block is been introduced in [7]. Compressors are extensively employed in the design of fast multipliers which are presented in [8-10] to enable faster partial product reduction tree and power consumption reduction. In [8, 11 - 16], 4-2 accurate compressors design that is optimized were suggested. Approximate multiplication compression is been addressed in [17] [18]. An approximate multiplier for arithmetic signed data value speculation (AVDS) was presented in [17] and the Baugh-Wooley algorithm is used for multiplication. Therefore, for approximate computation, no unique design for compressors is been suggested. In [18], estimated compressor models were designed and these models do not aim at multiplication. It is important to remember that the work presented in [7] is more efficient than that proposed in [17] [18] by implementing a reduced multiplier block that can handle imprecise multiplication.

In DSP algorithms, addition operations are most widely employed and this algorithm includes filters, transforms, as well as predictions. This type of digital signal processing algorithm is commonly used in audio as well as video processing mobile devices that are powered by batteries. Four additions can be carried out at the same by using an effective 4-2 adder compressor. Generally, the critical path, as well as internal glitching, is been reduced by this higher order of parallelism which further minimizes the dynamic power dissipation. Topologies that are based on two CMOS+ gates to minimize power, area, as well as the delay of the 4-2 adder compressor are proposed in this research work. Using the Cadence Virtuoso tool, the suggested CMOS+ 4-2 adder compressor circuit topologies are implemented and simulated at electric as well as layout levels by using 45 nm technology. When contrasted to the research work in [19], the proposed 4-2 adder compressor realization minimizes power by 22.41%, delay by 32.45% as well as area by 7.4% respectively [19].

Designers have been compelled to explore alternative sources of calculating efficiency because the advantages of technological scalability have been reduced. The architectures that are based on multicores, as well as heterogeneous accelerators, are the outcomes of this work which further enhances the computing efficiency by reducing the power budgets. We have also examined approximate computing which is considered one of the major areas of interest over the previous few years. The basic concept behind approximate computing is to produce efficient outcomes in terms of quality and these concepts must be accepted in several areas like designing an algorithm for networks as well as distributed systems The methodologies of precise computing have been developed from ad hoc as well as applications that are specific and are extensively validated by systematic design approaches. At last, the development of applications like identification, processing, discovery, data analytics, inference, as well as vision is significantly extending the possibilities for imprecise computing. A unified cross-layer architecture for approximate computing [20] is been presented in this research work and also describes the concepts as well as important ideas that have influenced our work in this field.

Approximate computing has been widely analyzed for digital signal processing applications to maximize

accuracy as well as to improve other circuit characteristics like power, area as well as efficiency. With the help of upcoming nanoscale spintronic devices, we propose an approximate arithmetic circuit in this work. Primarily we describe a hybrid spin-CMOS majority gate model that is based on a hybrid spintronic device structure that comprises of magnetic domain wall motion stripe as well as a magnetic tunnel junction and thereby exploits the advantage of the spintronic device's intrinsic currentmode thresholding function. We also present a majority gate-based compact as well as energyefficient accuracy-configurable adder architecture. However, when compared to earlier approximate circuit designs, the proposed architecture adapts to the intrinsic robustness in a variety of applications to various degrees of precision which in turn were fixed to a constant degree of approximation in existing circuit designs. Two unique approximate compressors were presented in this work and fast multiplier systems employ these types of compressors. When contrasted to the recently developed domain wall motion that is based on full adder design, the simulation results of the proposed device-circuit SPICE indicate 34.58 percent and 66 percent reduction in power for precise as well as approximate modes of the accuracy-configurable adder. Furthermore, the suggested accuracyconfigurable adder, as well as approximate compressors, can be extensively used in digital image processing algorithms and the discrete cosine transform (DCT). The final results demonstrate that inverse discrete cosine transform and discrete cosine transform that employs approximate multiplier accomplish approximately 2x energy as well as speed up 3x by maintaining comparable output quality when compared to that of the precisely designed circuit [21].

One of the latest trending concepts in digital design is approximate computing which reduces the necessity for exact computation to enhance efficiency as well as speed. This research work presents a unique approximate compressor as well as an approach for these compressors to design effective approximate multipliers. With the help of the 40-nm library, approximate multipliers for multiple operand lengths have been synthesized using this approach. The suggested circuits offer enhanced speed and power for a target precision when compared to that of the existing approximate multipliers. This research work also includes image filtering as well as adaptive least

mean squares filtering applications [22]. For digital processing at nanometric scales, inexact computing is considered one of the desirable techniques. In the case of computer arithmetic designs, inexact computing plays a major role in this work. Design, as well as analysis of two unique approximate 4-2 compressors that are employed in a multiplier, are the main objectives of this research work. These architectures mainly depend on various compression aspects to enable inaccuracy in computing which is described as error rate and is also referred to as normalized error distance which is used to satisfy circuit-based figures of merit, the number of transistors, delay as well as consumption of power. For a Dadda multiplier, four distinct approaches for exploiting the suggested approximation compressors are designed as well as evaluated. The usage of approximate multipliers for image processing, as well as a wide range of simulation results, are presented in this research work. When contrasted to an accurate design, the proposed designs achieve a substantial reduction in the number of transistors, power dissipation as well as delay. Furthermore, the presented multiplier models exhibit outstanding image multiplication capabilities in terms of average normalized error distance as well as peak signal-tonoise ratio that is more than 50dB for the analyzed image samples [23].

To minimize the consumption of power of errortolerant intrinsic applications, approximate computing is been used which has received considerable attention. An approach for partial product perforation is described in this research work for developing approximate multiplication which mainly concentrates on hardware-level approximation. We demonstrate that in partial product perforation, the induced errors are finite as well as predictable and are dependent on the input distribution in a precise mathematical approach. A partial product perforation approach is implemented for various multiplier designs as well as used for identifying effective architecture-perforation configuration combinations for varied error conditions by detailed experimental analysis. It is demonstrated that the suggested partial product perforation reduces power, area as well as critical delay by 50%, 45%, and 35% respectively when compared to the precise model. Furthermore, the product perforation technique outperforms the conventional approximation techniques like truncation, the voltage over scaling, as well as a logic

approximation concerning error and power dissipation [24].

# 2 Proposed ML-Based Low Power and High Throughput 4:2 Compressors Adder for Tosam Multiplier

Probabilistic logic minimization on a precise 4:2 compressor is utilized in this work. Here the minterms bit flipping of Boolean functions of Sum, Carry as well as *Cout* is computed to reduce operands and thereby minimize delay, area, and power consumption of the circuit. Various combinations have been examined and are found to be proportional to the number of bits that are flipped to determine the desirable bit flips. The complexity of the circuit should be reduced with a reduced error rate only after the implementation of the above procedure on any precise circuit. Since the most significant bit is very much important in achieving the outcome than the least significant bit, the lower order bits of the 4:2 compressor are flipped from one's to zero's and thus resulting in a 25% error rate (i.e., the number of inaccurate outputs to correct outputs) with no reduction in the number of inputs as well as outputs. The bits can be reversed from 'one to zero' or even from 'zero to one,' but 'one to zero' is carried out in this work to minimize the circuitry size. Carry is flipped by 4, Cout and SUM are given in Equation 1 and Equation 2 illustrates the k-maps of Sum, Carry, as well as Cout.

Two unique approximate 4-2 compressors are suggested as well as examined in this research work. When compared to the existing improved (accurate) 4-2 compressor designs illustrated in [8], the proposed simplified compressors show improved results in terms of delay, as well as power, consumed [8]. Further, the restoration unit of the Dadda multiplier makes use of the proposed approximate compressors. The approximate multiplication makes use of four distinct techniques. In the case of CMOS feature sizes like 32, 22 as well as 16 nm, the detailed simulation results are presented at the circuit level for figures of merit like delay, power dissipation, transistor count, and error rate, as well as normalized error distance. Further, the use of these multipliers in image processing is been discussed. The outcomes of two instances of image multiplication are presented. Furthermore, the presented multiplier models exhibit outstanding image multiplication capabilities in terms

of average normalized error distance as well as peak signal-to-noise ratio that is more than 50dB for the analyzed image samples. The suggested approximate architectures for the compressor as well as multiplier are feasible alternatives for inexact computing as per the analysis and simulation findings. Minimizing n numbers to two numbers is the primary objective of multi-operand carry-save addition or parallel multiplication. As a result, n-2 compressors or n-2 counters are frequently employed in computer arithmetic. An n-2 compressor shown in Fig.1 is a circuit slice that reduces n integers to two numbers when correctly reproduced. The n-2 compressor accepts n bits in position as well as one or more carry bits from the places to the right i.e., i - 1 or i - 2, in slice i of the circuit. Two output bits in i and i + 1places, as well as one or more carry bits in higher positions like i + 1, are generated. When compared to the precise compressor in [1] as well as approximated 4:2 compressors, the proposed architecture has more elements in the form of AND and OR gates. However, the precise compressor's XOR, XNOR, as well as MUX circuits use more transistors, which improves the structure of the multiplier and power consumption of this circuit as shown in Fig.2.



Fig. 1: Proposed design flow diagram of TOSAM for both signed and unsigned operations

In Fig.1,  $P_{L_A}$  and  $P_{L_B}$  are one detection-based position leadings, and  $|A_{appx}|$  and  $|B_{appx}|$  are truncated bits in both input operands. The  $F_{A_{partial}}$ and  $F_{B_{partial}}$  are fractional of both operands after performing truncation. X is the output of arithmetic operation i.e to perform  $1 + F_{A_{partial}t} + F_{B_{partial}t} +$ 

$$F_{A_{appx}}xF_{B_{appx}}$$

$$AxB \cong AxB_{appx} = 2^{P_{L_A}}x2^{P_{L_B}}(1 + F_{A_{partial_t}} + F_{B_{partial_t}} + F_{A_{appx}}xF_{B_{appx}})$$
------(1)

The remainder of Eq.(1) is stored as fractional part (F) of both operands A and B derived from Eq.(1). From the Eq.(1), the proposed 4:2 compressor for proposed TOSAM multiplier is shown in Fig.3.



Fig. 2: Gate level schematic of one detector leading module for 8-bit input data.

The Fig.2, the sign is the output bit which will be set in line with the sign of the proposed multiplier operands and the zero-detector module set the output bits to zero when any bit of input is zero. In unsigned multipliers, the TOSAM should be neglected the sign and it replaced with a Zero detector module to optimize the delay and area. The suggested 4:2 compressor's power consumption, as well as the size of the circuit, have been lowered because of the application of modules such as AND, 2T MUX, OR, and 6T XOR & XNOR that are based on pass transistor logic, as well as the transistor count has been reduced to 34 from 52 [2] and 50 [1]. 25% is the error rate for the proposed design because the total number of bit flips is approximately 8 as shown in Fig.3. The error rate of design1 is 37.5 percent and the error rate of design 2 is 37.5 when compared to the work presented in [8]. In the suggested design error rate as well as design 2 [8] error rate, there are only four inputs due to the absence of carrying Cin. The suggested architecture, as well as the precise compressor, consists of the same number of inputs, but slightly higher transistor counts, as well as power, are consumed than the work in [8]. Dadda multiplier is the quickest multiplier among all other parallel multipliers. As a result, this approach is used in this research work to execute multiplication operations. In this paper, simple AND gates were utilized to generate the partial products for the 3 stages of multiplication, whereas Carry Save is utilized to generate the partial products for the second stage of multiplication as well as Carry Propagate Adders were used for the third stage of multiplication. Among all the three stages, the second stage of multiplication is considered the most crucial stage in the design of a multiplier because it significantly reduces the operation and efficiency. As a result, to minimize power consumption and to enable faster functioning of the entire multiplier circuit, compressors are been employed in the second stage. Fig 6 depicts the simulated results of multiplier architecture with precise compressors. In this paper, AND gates are used to design an n=8 Dadda multiplier, with approximated 4:2 compressor, as well as an accurate carry, propagate adder in the second and third phases respectively.

Five Dadda multipliers are simulated in this research work.

All the precise 4:2 compressors of [2] make use of the initial multiplier.

- All the precise 4:2 compressors of [1] make use of the next multiplier.
- During the initial phase of the Dadda multiplier, the precise multipliers of [2] are employed by the 3rd multiplier and in the second phase, the suggested approximate 4:2 compressor is been used. In the initial phase, precise compressors are used to allow only estimates in the second phase.
- During the initial and second phases of the Dadda multiplier, precise multipliers of [1] are employed by the 4th multiplier, and suggested approximate 4:2 compressors are employed in the second phase.
- During the first as well as the second phase of the Dadda multiplier, all approximate 4:2 compressors are employed by the 5th multiplier.



Fig. 3: Proposed 4:2 compressor adder for TOSAM multiplier

The proposed 4:2 compressor is the main block adder for adding the partial products generated by the TODAM multiplier and Fig.3 shown only XOR and AND gates to minimize the number of transistors and power consumption optimization. The MUX blocks are to select the proper selection of carrying bits generated by XOR gates to optimize latency and increase throughput. The main advantages of this circuit are the reduction of the number of transistors from 22 to 34.

The initial two multipliers are precise multipliers with zero error distance, but with greater delay and power consumption. Since the approximate compressor is employed by only one phase of the multiplier, the third multiplier, as well as the fourth

multiplier, are partially approximate multipliers. The fifth multiplier increases the error distance and thereby reduces the area as well as power consumed by incorporating all approximate multipliers. Therefore, it is a completely exact multiplier. The designs of the 3rd and 4th multipliers are not depicted because the design is similar to that of Fig. 3, however, the precise compressors in phase 1 have been substituted with an approximate 4:2 compressor and the 5<sup>th</sup> multiplier consists of all suggested compressors. Since the number of inputs as well as outputs does not change, the multiplier structure also does not change. The significance of employing the suggested compressors for multiplication is been examined in this section. In most cases, a precise (exact) multiplier is made up of three components (or modules) [8].

- Generation of Partial product.
- A Carry Save Adder (CSA) tree is used to minimize the matrix of partial products to two addition operands.
- For the final estimation of the binary output, a Carry Propagation Adder (CPA) is used.

The second unit is very important for designing the multiplier in terms of circuit complexity, power consumption, and delay. To enable faster operation of the CSA tree, compressors have been frequently employed to reduce the power consumed and thereby resulting in quick and low-power operation [9, 10]. The approximate multiplier is obtained with the application of approximate compressors in a multiplier's CSA tree. To evaluate the effect of implementing the suggested compressors in approximate multipliers, an 8x8 unsigned Dadda tree multiplier is being used. In the initial phase, all the partial products using AND gates are generated by the suggested multiplier. The reduction architecture of a precise multiplier for n=8 is shown in Figure 9(a). Half-adders, full-adders, as well as 4-2 compressors, are used in the reduction phase of this design, and every partial product bit is indicated by a dot. Two half-adders, full-adders, as well as eight compressors, are used in the first phase to minimize partial products into four rows. The final two partial product rows are calculated using one half-adder, full-adder, as well as ten compressors in the second or final phase. In the reduction circuit of an 8x8 Dadda multiplier, two phases of reduction as well as three half-adders, full-adders, and 18 compressors are

required. When contrasted to an accurate multiplier, the first two estimated models aim to minimize delay and power usage. A large error distance, on the other hand, is predicted. To reduce the error distance, multiplier 3 and multiplier 4 are presented. The precise compressors in the critical path influence the delay in these systems. As a result, when compared to an accurate multiplier, there is no change in latency for these imprecise systems. The use of imprecise compressors in the least significant columns is intended to reduce power consumption as well as the number of transistors. The initially suggested two multipliers have improved delay as well as power consumption, whereas, the 3rd and 4th designs are predicted to have reduced error distances. By using CMOS logic in NOR and XNOR gates, as well as by using pass transistor logic, the approximate 4:2 compressor design 1 has been employed in distinct ways and the combinations are illustrated in Table I. The area, as well as power consumed by the multipliers, vary depending on the number of transistors of exactly 4:2 compressors and 4:2 approximate compressors. In all of the preceding multipliers, the first and 3rd phases are identical and precise; whereas in the second phase, the 4:2 compressors have been substituted as per the multiplier implementation type. The performance measures of multipliers are shown in Fig 4 (a & b).

Table 1. Comparison between existing and proposed TOSAM multiplier

Parameters	Existing	Proposed
Slice Registers	852	466
Slice LUT's	520	290
Flip-Flops	752	465
Delay in ns	5.72	3.875
Power in mW	1.4	0.088

The complete TOSAM multiplier is synthesized in Vivado Design suite software and its design summary is shown in Table.1, the number of slice registers and LUTs is reduced by 45% and power is minimized by 41%.

Table 2. Comparison between existing and proposed Dadda multiplier with 4:2 compressor adder

Duduu manipher with 112 compressor udder		
Parameters	Existing	Proposed
Slice Registers	852	48
Slice LUT's	520	103
Flip-Flops	752	103

Delay in ns	5.72	11.705
Power in mW	14	0 4 4 8

The Dadda multiplier with a 4:2 compressor adder consumed more power and produced more delay in partial products. So it concluded that the TOSAM multiplier with a 4:2 compressor adder has optimized power, area, delay, and hardware utilization. Therefore, the TOSAM with adder is applied for image processing applications as validations of results. To minimize an error between exact and approximate multipliers, the Machine Learning (ML) is applied, based on the number of iterations, ML has greatly reduced the error and it is nearly 0.02%, therefore, proposed TOSAM with ML and 4:2 compressor adder is best suitable for any complex applications like image and video processing applications with minimal hardware utilization and high throughput.



Fig. 4: Performance analysis of proposed TOSAM with 4:2 compressor adder.

The proposed design is compared with a recently published article in [24] about Resilient backpropagation Neural networks (RBPNN) as Machine learning and it has been applied for reduction of error produced between exact and approximate multipliers. The proposed multiplier is validated for real-time data which is published and available at Rajasthan technical university as a standard dataset and discussed in [24], this dataset is about forecast weather. The neural network has compared two data sets errors after RBPNN is trained and accordingly generates an error to minimize this error. The RBPNN is the latest biological network and its operation is based on interconnected neutrals with biases and weights. This network is mainly to solve regression-related issues like forecasts related to weather.

*Error Analysis:* Measuring accuracy is been conducted for the specified binary inputs A[0-7] as well as B[0-7] along with 100ns stop time. A multiplier's error distance is defined as the absolute difference between the actual product (P) as well as the approximate product (P'), which is determined for every 5ns time ranging between 0 to 100ns. MATLAB is used to carry out all of these tasks. P, P' are depicted in Fig. 9 as an instance of ED, MED, as well as NED computations for the period of 10ns to 15ns.

Error Distance: To evaluate the error distance, four new approximate multipliers are computed. For n=8, the multiplier presented in [7] i.e., multiplier 5 is approximated. For n=8 as well as for k=1, the reduced multiplier with constant correction [5] (Multiplier 6) as well as the reduced multiplier with variable correction [6] (Multiplier 7) is approximated. Furthermore, to evaluate the effect of implementing the suggested approximate compressors with other approximate compressors, an approximate multiplier i.e., Multiplier 8 is simulated. The 4-2 compressors are designed using two precise full adders and are used in an 8x8 Dadda multiplier (Figure 3). The approximate multiplier utilizes the first full-adder design that is presented in [2]. 8 approximate multipliers are evaluated in this work, namely the four proposed architectures as well as the remaining four are the approximate multipliers which is been described in Table IX along with their significant characteristics. To evaluate these approximate multipliers, a normalized error distance (NED) is employed. The average error distance throughout all inputs and is further reduced by the maximum possible error is termed normalized error distance in [1]. As a result, the average NED and the normalized error distance described in [1] are all same. The normalized error distance is specified for

every input in this work. The highest absolute value of NED is achieved when the incorrect output is more (less) than the precise outcome and is also known as the maximum high (low) NED. The average normalized error distance, maximum high and low normalized error distances, as well as the accurate number of approximate multiplier outcomes for n=8, is illustrated in Table X. The possibility of accuracy for any design is represented by the number of correct outputs out of the overall outputs. According to Table X, the possibility of accuracy in Multiplier 1 and Multiplier 4 is 0.16 percent (103 out of 65025), as well as 14.3 percent (9320 out of 65025) respectively. The suggested approximate multipliers will yield an incorrect output only if one of the inputs is 0, and the suggested approximate compressors will produce output when all the inputs are zero (row 1 in Tables II and III). Therefore, the multiplier can yield an accurate outcome by introducing a circuit to identify zero-valued inputs for n=8. As a result, the zero-valued input patterns are not taken into account anymore in the simulation for a valid comparison.

When compared to all estimated multipliers, a low average normalized error distance is achieved by multiplier 4. Multiplier 4's average NED is improved 18 times that of Multiplier 5, 2 times that of Multiplier 6, as well as 1.5 times that of Multiplier 7. Maximum correct outcomes are achieved by multiplier five and also possess reduced max high NED. The maximum high NED for this architecture is Zero because the estimated result is always less than the actual output. As a result, when compared to all the architectures, multiplier 5 possesses the worst maximum low NED. To evaluate the results of approximate multipliers, a graph of the NED distribution is made (Figure 10). The product in an 8x8 multiplier ranges between 0 to 65025 and is unsigned values. 127 intervals are used to classify all probable outcomes. The output varies between 0 to 512 during the first interval and ranges between 513 to 1024 in the second interval. The outcome varies from 64513 to 65025 during the last interval. The approximate multiplier is then calculated using the average NED for every interval. Figures 10a and 10b demonstrate that the average NED for Multipliers 1 and 2 increases only at extremely high or very small product values, indicating that these approximate multipliers result in a minimal output error on average when compared to the precise estimation.

*Approximate Multipliers:* For n=8, the suggested four approximate multipliers are been evaluated and these approximate strategies, as well as precise multipliers, are examined in terms of delay, power consumed as well transistor count. The comparison between other approximate multipliers as well as with the suggested multipliers' error distance is explored in this work.

*Approximate Compressors:* At a frequency of 1 GHz, the two estimated compressors, as well as the least precise compressor of [8] that are constructed with XOR-XNOR gates, are simulated in this work. In all simulations, a fan out of four is used. Table IV shows the simulation results for power consumption, delay, as well as power-delay product (PDP) employing PTMs at 16 nm, 32 nm as well as 22 nm.

Delay: Based on the reduction phases as well as the delay of every phase, the delay of the reduction circuitry i.e., the second module of a Dadda multiplier is computed. Approximate compressors are employed in all columns in the case of multipliers 1 and multiplier 2. As a result, approximate compressors' delay, as well as the delay of each phase, are the same. Therefore, approximate compressors delay and multiplier three, as well as multiplier 4 delay, are all same. On the other hand, there is no change in delay when using an approximate compressor in n/2 LSBs than using a precise multiplier. Table VI represents the delayed improvement in the reduction circuitry of every multiplier at 32 nm CMOS technology when contrasted with a precise adder.



Fig. 5: Simulated results of Dadda multiplier with 4:2 Compressor for 16 bits.



Fig. 6: Simulated results of TOSAM multiplier with 4:2 Compressor for 16 bits of inputs A = 4321 and B = 7643 and product is 32964608.

The simulated results of both Dadda and TOSAM multipliers are validated for different test vectors and based on products results as shown in Fig.5 and Fig.6, the error between exact i.e Dadda multiplier and approximate multiplier i.e TOSAM is 33025403 -32964608 =60795 so error rate is 0.6 and this error is acceptable for image and video processing applications.

# **3** Application of Proposed Digital Multipliers Using ML

Nowadays, ML has become the most important topic in almost all areas. In general, there are two ways to apply machine learning to VLSI. One way is by utilizing hardcoded coding where the program's exact constraints, as well as necessary outputs, are described and the instructions for the entire process are specified. Therefore, by employing ML algorithms the VLSI processes are made more precise as well as simple. In contrast, an architecture for a neural network is developed to understand how the task can be carried out by itself. Designing effective VLSI hardware designs that are based on neural networks enables the applications of Machine Learning. ML is a process that is carried out continuously by evaluating different alternatives until an appropriate solution is found. Minimizing the number of errors that are present between precise as well as approximate 4:2 compressors is the objective of this research work. By using inaccurate logic minimization, several combinations of imprecise compressors are developed until a lower error rate is achieved. Tensor Flow is used to integrate VLSI approximate circuits with Machine Learning. An algorithm with a reduced error rate is demonstrated by choosing a 4:2 compressor.

## 4 Manual Calculation of TOSAM Multiplier

Let 16 bit of X operand and 16 bit Y operand Input Parameters (h, t), where h: height and t: fraction part For Ex: 1011 1000 001 can be represented as 1.011 1000 001 x 2<sup>-10</sup> Green – fraction part 0000 0010 0001 0 101 x 0001 1010 001 1 1100  $2^{15}, 2^{14}, 2^{13}, \dots, 2^1, 2^0$ Check first 1 from MSB and find its binary location Here for X first '1' comes at  $2^9$  KA=9 Here for Y first '1' comes at  $2^{12}$  KB=12 Next to KA fractional parts and t = 7(XA)t = 0000101(YA)t = 1010001APX first 3 bits from (XA)t and pad '1' at LSB side (XA)APX = 0001(YA)APX = 1011**Final computation** =  $((XA)APX) \times (YA)APX + 1$ = 8 bit output  $= 0000 \ 1011 \ = 0000 \ 1011$  $((XA)APX) \times (YA)APX + 1 + (XA)t + (YA)t =$ 0000\_1011 0000 1010 (pad 0 at LSB side) 1010 0010 (pad 0 at LSB side ) 1011 0111 **POST SHIFT** >> KA+KB-(t+1)= (9+12)=21 1011 0111) << 21 (already 8 bit fractional part) (21-8) = 13 times << FINAL OUTPUT= 01 1011 0111  $0000 \ 0000 \ 0000 \ 0 = 35,96,288$ **EXACT OUTPUT= 1101101001111011101** = 35,79,628

## **5** Conclusion

This research work provides the presentation of approximate 4:2 compressors using approximate logic minimization with error analysis. The 8x8 Dadda multiplier employs a modern 4:2 compressor. The analysis of error and simulations of these designs were carried out using MATLAB and Vivado Design Suite 2018.1 simulator. The circuit developed in this work achieves a lower error rate and average power by considering the transistor count. At last, this research work described the concept of integrating VLSI approximate circuits as well as machine learning. One of the unique approaches for computing nanoscale is approximate computing. For computing, approximate computer arithmetic provides huge operational benefits, and there exist many works on inexact adders. Furthermore, this research work primarily concentrates on compression in the context of a multiplier till now no work on this area has been presented as per the author's knowledge. The unique architectures of two approximate 4-2 compressors were proposed in this research. The compression unit comprising four approximate multipliers uses these approximate compressors. When compared to a precise design, proposed approximate compressors exhibit a reduced number of transistors, delay as well as power consumption.

- The number of transistors used in design 1 and design 2 is reduced by 46% and 50% respectively.
- For CMOS implementation at technology scaling of 32, 22 as well as 16 nm, the first design and second design offer a power reduction of 57% as well as 60 % respectively.
- Overall, for varied CMOS feature sizes of 32, 22 as well as 16 nm, the second design as well as the first design show 44% and 35% improvement in delay respectively.

This research work proposes four distinct approximate approaches to examine the performance of inexact compressors for the specified imprecise multiplication measures. The reduction unit of a Dadda multiplier employs inexact compressors. The simulation outcomes reported in this work are as follows:

- When contrasted to a precise multiplier, the first multiplier and second multiplier suggested in this work exhibit better results in terms of the number of transistors as well as power usage.
- The second multiplier and first multiplier exhibit significantly large normalized error distances as well as large PSNRs. The second multiplier also exhibits maximum delay by using a second inexact compressor for all bits.
- The 3<sup>rd</sup> multiplier and 4<sup>th</sup> multiplier exhibit reduced normalized error distance values because of the reduced number of transistors and power consumed and thereby indicating optimum tradeoff for energy as well as precision

Furthermore, the presented imprecise multipliers are employed in many of the outstanding image processing applications that achieve a peak signal-tonoise ratio that is approximately 50dB. The comparison of the suggested four approximate

models against four other approximate models is been evaluated in Table XIII. In terms of all performance criteria for imprecise multiplication as well as the for two PSNR instances, the multiplier 4 is considered one of the best solutions. Concerning Max High NED as well as the number of accurate outputs, the multiplier 5 exhibits better results. Whenever the PSNR instances are taken into account, the other performance criteria that are low enable the PSNR to rank in the middle. Among all the strategies examined in this research work, multiplier 3 is the second most efficient design. Existing, as well as upcoming work, mainly concentrates on the trade-offs between various performance criteria. Physical architectures of approximate multipliers are explored to facilitate the analysis described in this work. At last, this report illustrates the multipliers that are used for approximate computing and can be designed using an appropriate approximation compressor architecture. These multipliers provide huge benefits in terms of error measurements as well as circuit-level. Analysis of error indicators is the present concept that is examined in this work.

#### References:

- [1]. Veeramachaneni *et.al.*, Novel architectures for high-speed and low power 3-2, 4-2 and 5-2 compressors, Proc. Int. Conf. on VLSI Design (VLSID), 2007, pp. 324-329.
- [2]. Chang *et.al*, Ultra- voltage, low power CMOS 4-2 and 5-2 compressors for fast arithmetic circuits, IEEE Trans. Circuits Syst. I, Fundam. Theory Appl., 2004, 51, (10), pp. 19851997.
- [3]. Raphael, D *et.al*, A Power-Efficient 4-2 Adder Compressor Topology, 15th IEEE (NEWCAS), Strasbourg, France, 2017, pp. 281-284.
- [4]. Swagath, V *et.al*, Approximate Computing and the Quest for Computing Efficiency, 52nd (DAC), 2015, San Francisco, CA, USA.
- [5]. Shaahin, A *et.al*, Majority-Based Spin-CMOS Primitives for Approximate Computing, IEEE Trans. on Nanotech., 17(4), 2018, pp. 795-806.
- [6]. Avinash, L *et.al*, Parsimonious Circuits for Error-Tolerant Applications through Probabilistic Logic Minimization, Int. Workshop on PATMOS 2011, pp.204-213.
- [7]. Darjnlow, E *et.al*, Approximate Multipliers Based on New Approximate Compressors,

Anil Kumar D.

IEEE Trans. on CAS-I: Reg. Pap, PP(99), 2018, pp. 1-14.

- [8]. Momeni, A *et.al*, Design, and analysis of approximate compressors for multiplication, IEEE Trans. on Comp., 64 (4), 2015, pp. 984994.
- [9]. Liang, J *et.al*, New Metrics for the Reliability of Approximate and Probabilistic Adders, IEEE Trans. on Comp., 63(9), 2013, p. 1760-1771.
- [10]. Zervakis, G *et.al*, Design-Efficient Approximate Multiplication Circuits Through Partial Product Perforation, IEEE Trans. on VLSI Systems, 24(10), 2016, pp. 3105-3117.
- [11]. A. Betti, M. Gori, and G. Marra, "A Constrained-Based Approach to Machine Learning," 2018 14th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), 2018, pp. 737-746, DOI: 10.1109/SITIS.2018.00118.
- [12]. Y. Cho and M. Lu, "A Reconfigurable Approximate Floating-Point Multiplier with CNN," 2020 International SoC Design Conference (ISOCC), 2020, pp. 117-118, DOI: 10.1109/ISOCC50952.2020.9332978.
- [13]. M. Hajizadegan and P. Chen, "Harmonics-Based RFID Sensor Based on Graphene Frequency Multiplier and Machine Learning," 2018 IEEE International Symposium on Antennas and Propagation & USNC/URSI National Radio Science Meeting, 2018, pp. 1621-1622, DOI: 10.1100/APUSNCUPSINPSM 2018 8608604
  - 10.1109/APUSNCURSINRSM.2018.8608604.
- [14]. D. G. Mahmoud, B. Shokry, A. ElRefaey, H. H. Amer and I. Adly, "Runtime Replacement of Machine Learning Modules in FPGA-Based Systems," 2021 10th Mediterranean Conference Embedded Computing on 2021. (MECO). pp. 1-4, DOI: 10.1109/MECO52532.2021.9460192.
- [15]. Y. Ishiguchi, D. Isogai, T. Osawa and S. Nakatake, "A Perceptron Circuit with DAC-Based Multiplier for Sensor Analog Front-Ends," 2017 New Generation of CAS (NGCAS), 2017, pp. 93-96, DOI: 10.1109/NGCAS.2017.23.
- [16]. Xiang-Jun Ji and Na Han, "Evaluation model on the building of real estate brand based on income multiplier," 2009 International Conference on Machine Learning and Cybernetics, 2009, pp. 2549-2554, DOI: 10.1109/ICMLC.2009.5212098.

- [17]. R. Zhang and Q. Zhu, "Consensus-based transfer linear support vector machines for decentralized multi-task multi-agent learning," 2018 52nd Annual Conference on Information Sciences and Systems (CISS), 2018, pp. 1-6, DOI: 10.1109/CISS.2018.8362195.
- [18]. H. Wang, Y. Gao, Y. Shi, and R. Wang, "Group-Based Alternating Direction Method of Multipliers for Distributed Linear Classification," in IEEE Transactions on Cybernetics, vol. 47, no. 11, pp. 3568-3582, Nov. 2017, DOI: 10.1109/TCYB.2016.2570808.
- [19]. R. Dornelles, G. Paim, B. Silveira, M. Fonseca, E. Costa, and S. Bampi, "A power-efficient 4-2 Adder Compressor topology," 2017 15th IEEE International New Circuits and Systems Conference (NEWCAS), 2017, pp. 281-284, DOI: 10.1109/NEWCAS.2017.8010160
- [20]. S. Venkataramani, S. T. Chakradhar, K. Roy and A. Raghunathan, "Approximate computing and the quest for computing efficiency," 2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC), 2015, pp. 1-6, DOI: 10.1145/2744769.2744904.
- [21]. S. Angizi, H. Jiang, R. F. DeMara, J. Han and D. Fan, "Majority-Based Spin-CMOS Primitives for Approximate Computing," in IEEE Transactions on Nanotechnology, vol. 17, no. 4, pp. 795-806, July 2018, DOI: 10.1109/TNANO.2018.2836918
- [22]. D. Esposito, A. G. M. Strollo, E. Napoli, D. De Caro, and N. Petra, "Approximate Multipliers Based on New Approximate Compressors," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 65, no. 12, pp. 4169-4182, Dec. 2018, DOI: 10.1109/TCSI.2018.2839266.
- [23]. A. Momeni, J. Han, P. Montuschi and F. Lombardi, "Design and Analysis of Approximate Compressors for Multiplication," in IEEE Transactions on Computers, vol. 64, no. 4, pp. 984-994, April 2015, DOI: 10.1109/TC.2014.2308214.
- [24]. G. Zervakis, K. Tsoumanis, S. Xydis, D. Souris and K. Pekmestzi, "Design-Efficient Approximate Multiplication Circuits Through Partial Product Perforation," in IEEE Transactions on Very Large-Scale Integration (VLSI) Systems, vol. 24, no. 10, pp. 3105-3117, Oct. 2016. DOI: 10.1109/TVLSI.2016.2535398.

### Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0 https://creativecommons.org/licenses/by/4.0/deed.en

US