Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.7922871

MAC DESIGN USING APPROXIMATE MULTI-LEVEL COMPRESSORS WITH BALANCED ERROR ACCUMULATION

K Sridevi[1],Dr M Santhanalakshimi[2,] Dr S Bhavani[3],R Priya[4]
Assistant Professor[1], Department of ECE, Sri Shakthi Institute of Engineering and Technology, Coimbatore, Tamil Nadu

Associate Professor[2], Department of ECE, PSG College of Technology, Coimbatore, Tamil Nadu

Professor [3], Department of ECE, Sri Shakthi Institute of Engineering and Technology, Coimbatore, Tamil Nadu

PG Student[4], Department of ECE, Sri Shakthi Institute of Engineering and Technology, Coimbatore,

Tamil Nadu

srideviece@siet.ac.in , ms.ece@psgtech.ac.in, hodece@siet.ac.in, priyarajendran8552@gmail.com

ABSTRACT: *The Multiply-Accumulate (MAC) processing can be implemented using an approximative computing technique. Unlike earlier research, where the approximate range was constrained by an error accumulation issue. In contrast, a number of approximation multipliers are interleaved in this study to account for mistakes that occur when accumulate operations are performed in the opposite direction. Create the roughly 4-2 compressors that produce mistakes in the opposite direction while first reducing the computational expenses for the balanced error accumulation. Then, to provide a comparable error distance, positive and negative multipliers are painstakingly developed based on the probabilistic analysis. According to simulation results on various real-world applications, the suggested MAC processing offers the energy-efficient computing scenario by widening the range of approximatively portions. The core-level energy is decreased by the proposed interleaving strategy.*

Keywords: *Convolutional Neural Networks, Multiplier, Image Processing, Approximate Computing,Multiplier.*

1.INTRODUCTION

(a)Approximate Computing:

The energy usage of various signal processing techniques, including machine learning and multimedia digital signal processing, which are widely recognised to have error-tolerable properties, has been identified as a viable area for energy savings [1]. It is necessary to design the cost-effective approximation Multiply-Accumulate (MAC) operator that minimises the energy consumption for the given amount of computational mistakes because these multimedia-related algorithms mostly involve intensive matrix multiplications [7]. Although studies of approximate adders have been put forth [8], prior works have concentrated on relaxing multiplier costs, which consume a great deal more energy than addition operations due to the complexity of realising each arithmetic unit.

Energy efficiency is one of the primary design criteria for virtually all electronic systems, including portable ones like tablets, smartphones, and other devices. It is highly desirable to achieve

this minimization with the least possible performance (speed) penalty. Digital signal processing (DSP) blocks are the essential elements of these portable devices for implementing diverse multimedia applications. The computational heart of these blocks is the arithmetic logic unit. The majority of all executed mathematical operations—multiplications—occur here. Therefore, increasing the multipliers' speed and power/energy efficiency characteristics is crucial for raising the efficiency of processors. Many DSP cores are used to implement image and video processing algorithms, the results of which are either videos or images that are fit for human consumption. This explains why approximations can be used to increase speed or energy efficiency.

(b)Multiplier in Error Tolerant Applications:

Some error-tolerant applications, like data mining, deep learning, and multimedia signal processing, have advanced quickly in recent years.These programmes use a lot of electricity and have high computational demands. Approximate computation is a viable technique for creating energy-efficient systems for these applications because even with a little amount of computation error, they can still produce meaningful and effective results. Multiplications require a lot of space and processing power in hardware since they are one of the most frequently utilised fundamental arithmetic operations. As a result, approximation multipliers that could lower hardware cost by compromising some precision have drawn an increasing amount of interest. Approximate multipliers have been the subject of extensive research. The dynamic segment approach, which is more accurate than simple truncation, multiplies by the leading consecutive significant bits of the input operand. Due to the truncation of more than half of the partial products, the multiplication operation is changed to a smaller digit multiplication plus add and shift operations.

Because full adders (3/2 compressor) are used so frequently, it is noteworthy that partial product reduction uses a lot of space and energy. To effectively decrease the critical path and lower hardware costs, great effort has been devoted into improving approximation compressors.In approximate compressors, the partial products are compressed using only OR gates. There are both precise and approximate operating modes in the four dual-quality reconfigurable approximate 4/2 compressors proposed in [9]. Additionally, some large scale multipliers have also used high-order compressors.The approximation multipliers offered exhibit superior performance than many previous designs as a result of the advancement of compressors. In order to develop approximate multipliers, this study proposes an optimised compression technique. This strategy not only greatly improves accuracy but also marginally lowers hardware cost.

(c)Partial Product Reduction:

There are not many fundamental building blocks in a multiplier. These three processes are partial product production, partial product reduction, and carry-propagate addition. Among them, approximations may be used. Truncation of the partial products is one such example and a well-known approximation method. Some of the partial product formations are not found in this method. There is a reduction in truncation error. It is accomplished with the aid of the appropriate correction functions. Other methods that use approximately 2 2 or 4 4 sub-multipliers to simplify the partial-product matrix. The partial product reduction step's approximations rely on. Compressors are used to convert a multi-operand sum into a two-operand addition utilising Wallace-style logarithmic schemes or the Three-Dimensional Method TDM. A logic circuit is a compressor that counts the number of "ones" in the input. The full-adder, which functions as a 3:2 compressor, is most frequently utilised. There are also

higher-order compressors (4:2 or 5:3). Compressors with XOR-rich circuits have a complicated multiplier block for partial products reduction. In terms of area, power, and speed.

(d)Approximate Compressors:

The suggested circuits, which are employed in two different 16-bit multiplier variations, include approximate half adders, full adders, and 4:2 compressors. This essay suggests a compressor with a ratio of roughly 15:4. This is constructed using a basic module known as the 5:3 compressor. It is suggested to use a lossy compression method based on the partial-product rows' bit importance. This strategy makes advantage of approximate half-adders (realised with straightforward OR gates) to provide a smaller set of product terms. A similar strategy is utilised, using basic OR gates as approximative counters. It is investigated how to approximate arithmetic units in architectural space. where four approximative multipliers for books have been suggested.

(e)Approximate MAC Design:

However, the earlier techniques typically only approximate in one direction, leading to imbalanced errors in either the positive or negative directions. Aggressive approximation from these cost-aware structures significantly worsens algorithm-level performance as approximate MAC operations collect the one-directional mistakes from the multiplier. The number of approximation bits is therefore strictly constrained. minimising the effects of approximation computation for large-scale MAC processing, such as deep learning applications. While some methods have balanced distributions at the multiplier level, these accuracy-aware compressor designs are complex and only slightly reduce energy consumption [19]. Therefore, by accumulating errors from multiplier-level approximation in a balanced way, it is necessary to extend the concept of approximate computing to the MAC level for the energy-optimized processing. Each of the suggested approximate multipliers significantly reduces the computational complexity, which allows it to support acceptable algorithm-level performance while radically reducing the energy used for MAC processing. Simulation results demonstrate that the suggested interleaving strategy reduces energy consumption for the deep neural network (DNN) processing and the image sharpening algorithm by 34.1% and 15.3%, respectively, compared to the state-of-the-art approximate multiplier.

2.PROPOSED METHODOLOGY

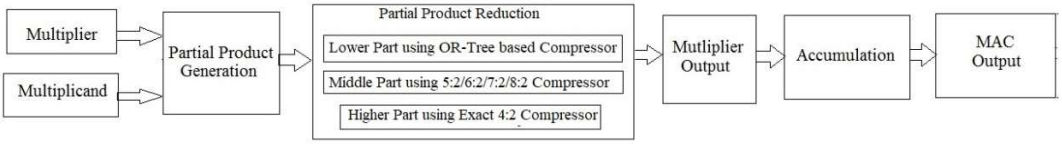
(a)Approximate 4-2 Compressor:

In this project, we start by developing the reasonably priced approximate 4:2 compressors. By taking into account the erroneous directions, it is done.The quantity of errors is not considered. The positive compressor (PC) and the negative compressor (NC) are two types of approximation 4:2 compressors that are designed to provide erroneous results in the opposite direction. For all inputs, the carry-free compressor approximation yields an error. The truth table can be changed to lower the cost of the hardware. By producing a value of 1 at y i independent of the input cases, the device complexity and power consumption are significantly reduced. Large relative mistakes then follow. Particularly for operands with zero values. In order to guarantee the accuracy for zero-valued inputs with the lower hardware costs, it is necessary to minimise the performance deterioration in algorithm level for real applications. With the exception of all zero inputs, the suggested approximation compressors always yield a value of 1, as shown in the following.

$$ai + bi + ci + di = yPi = yNi,$$

(b) Multi-level Compressors:

The maximum height of the PPM (partial product matrix) is frequently related to the critical path of a multiplier. As a result, the PPM must be compressed. When properly reproduced, a n:2 compressor is a portion of a multiplier that reduces n numbers (i.e., product terms) to two numbers. The n:2 compressor creates two output bits in places i and i+1 and one or more carry bits in the higher positions in slice i of the multiplier after receiving n bits in position i and one or more carry bits from the lower positions (such as i-1).Processor inputs include multiplier and multiplier. A process generator, the Partial Product Generator instructs you on what to do next and how to produce the output.



(c)The Approximation of Carry :

In this case, the Carry output logic approximation is being researched. Ch, the carry bit As follows is what a conventional half adder is defined as: X0. X1 = Ch (X0,X1).(7)In a typical full adder, the carry bit Cf is defined as follows:The equation (7) can be implemented as a modified half adder, and the equation can be implemented as a modified full adder. Cf (X0,X1,X2) = X0.X1 + X1.X2 + X0.X2 (8). Figures 1(a) and 1(b) show the logic of modified half adders and modified full adders, respectively. For the Carry output of a high-order approximate compressor, we can build the approximate logic based on the modified half adder and modified full adder.

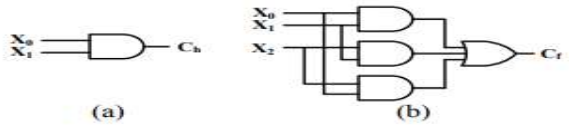


Fig 1. (a) Half adder modified (b)Full adder modified .

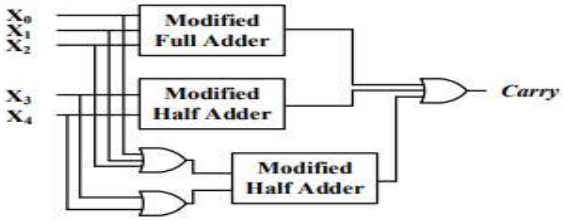


Fig 2. The approximate 5:2 compressor carry output logic.

Fig.2 It shows the carry output logic of a roughly 5:2 compressor. When there are 8 input bits (i.e., n = 8), we can divide them into 3 groups: one group contains X0, X1, and X2, another group contains X3, X4, and X5, and a third group contains X6 and X7. This would result in the following output from our roughly 8:2 compressor: Cf (X0,X1,X2) + Cf (X3,X4,X5) + Ch(X6,X7) + Cf(X0+X1+X2, X3+X4+X5, X6+X7).

(d)The Approximation of Sum:

Typically, the output Sum is generated using a tree of XOR gates. The XOR gate has more design overheads when compared to other logic gates. All design overheads, such as power, area, and latency, can be decreased if we can swap out XOR gates for other logic gates.

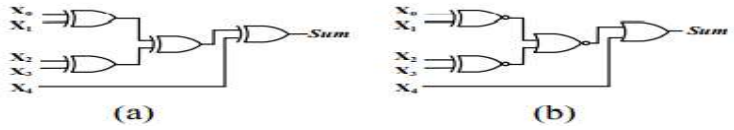


Fig 3. Sum of 5:2 compressor. (a) Accurate (b) Our approximate

For the Sum output of a high-order approximate compressor, we build the approximation logic (a tree of logic gates) as shown below. Instead of an XOR gate at the first level, we employ an XNOR gate. Keep in mind that the XNOR gate's output is the opposite of the XOR gate's output. We employ NOR gates at the second level and OR gates at the third level to account for the error rate. The design overheads are significantly reduced because all XOR gates are swapped out for other logic gates. The Sum output of the 5:2 compressor (Fig. 3) is used as an example in the sections that follow. The logic of the Sum output of a precise 5:2 compressor is shown in Fig. 3(a). The rationale of the Sum output is shown in Fig. 3(b) offers the reasoning for our approximate 5:2 compressor's Sum output.

(e) Multiplier using Multi-level Compressors:

Three components make up a multiplier. AND gates are used in the first phase. These are used to produce incomplete products. The height of the partial product matrix is greatest in the second component. Utilising a carry save adder tree reduces it. A carry propagation adder makes up the third element. It is employed to create the finished product. The PPM reduction circuitry, which is the second portion of the multiplier, and the partial product matrix reduction circuitry, which is related to the multiplier, are related in terms of design complexity. As a result, the research of multiplier design is centred on optimising the PPM reduction circuitry. In this section, we suggest a roughly 8 × 8 multiplier design.

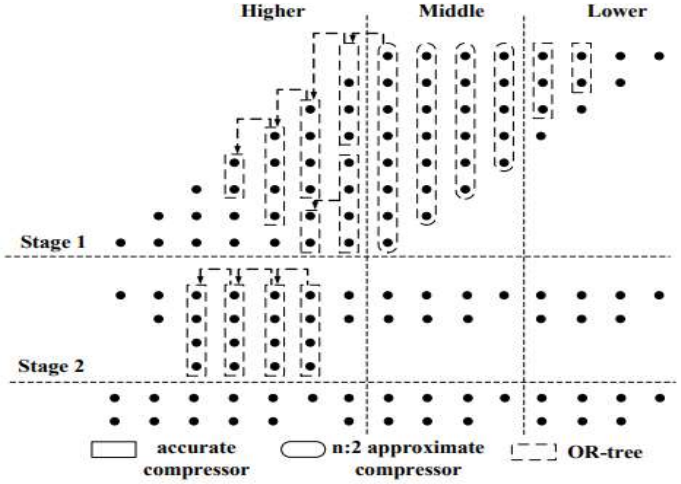


Fig : 4, The PPM reduction in the proposed approximate multiplier.

The general layout of our PPM reduction circuitry is shown in Fig. 4. The weights are divided into three groups based on their significance: the weights with a higher significance, the weights with a middle significance, and the weights with a lower relevance.Our PPM reduction circuitry uses the considerably driven logic compression technique to lower the power consumption with a little error: accurate The higher importance weights employ 4:2 compressors, the medium weights employ approximate high-order compressors, and the lower weights employ incorrect compressors. Two stages make up our PPM reduction circuitry. All of the weights are in the first phase. For the weights with a higher significance level, the second

step is applied. The maximum number of product terms for each weight after the second stage is two. Thus, the final result can be obtained by a carry propagation adder. We go into further depth about these two stages in the sections that follow.

First Stage:

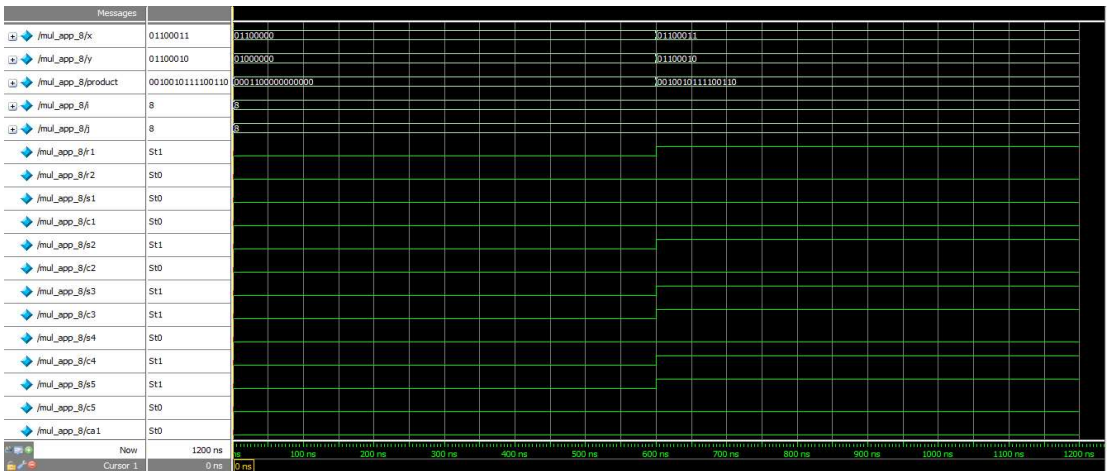
If there are n inputs, a straightforward OR tree-based approximation is utilised for each smaller significance weight to save power. if for n 2 no action is taken. To approximate the accumulation result of these n-1 inputs, on the other hand, if n > 2, we utilise an OR tree for the n-1 inputs. Consequently, each lower significance weight has a maximum of two product terms once the first stage is complete. For power savings, an estimated n:2 compressor is utilised for each middle significance weight, where n is the number of product terms in weight. The designers can select between the two implementations listed below, as explained in Section II: The first method uses an accurate Sum and an approximative Carry, while the second uses an approximative Sum and an approximative Carry. Each middle significance weight has a maximum of two product phrases after the first stage is complete. We employ accurate (i.e., exact) 4:2 compressors to produce results with great precision for each increasing significance weight. If the number of product terms is fewer than 4, the values of the other inputs to each accurate 4:2 compressor are set to 0. The Carry output of the leftmost middle significance weight is used to create the carry bit Cin of one accurate 4:2 compressor in the rightmost higher significance weight. Additionally, the other accurate 4:2 compressor's carry bit Cin is set to be zero.

Second Stage :

Note that only the higher significance weights are eligible for the second stage. In order to obtain great accuracy and lower the PPM's maximum height, we use precise 4:2 compressors. The carry bit Cin of the rightmost accurate 4:2 compressor is set to 0. Each higher importance weight has two product words once the second stage is finished.

3.RESULTS AND DISCUSSION

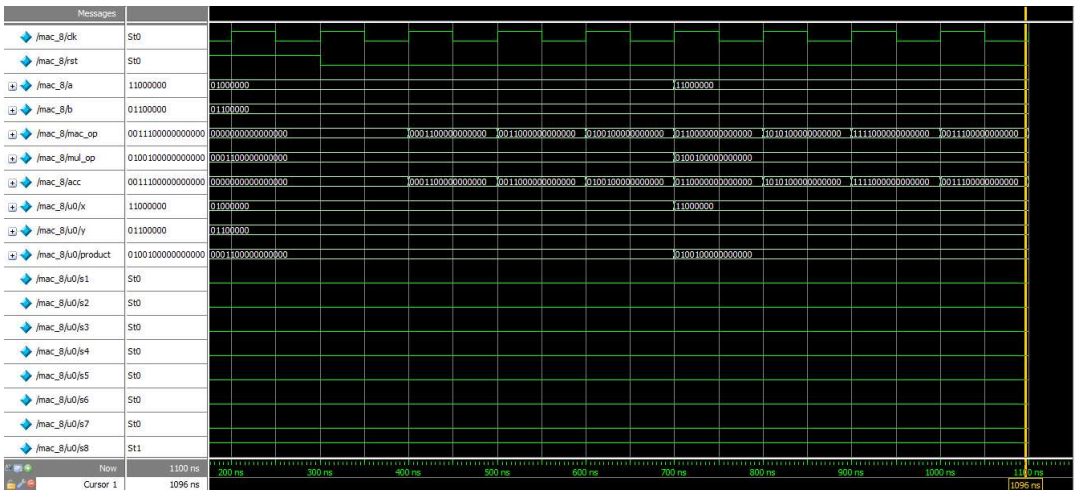
(a)Simulation result of multiplier:



The multiplier's input is made up of X and Y. The result of the approximating multiplier is PRODUCT. The intermediate sum and carry bits are S1–S8 and C1–C8. The proposed approximate multi-level compressors are used to implement approximate multipliers. By using

the precise importance driven logic compression technique, our PPM reduction circuitry reduces power consumption with little error. The higher significant weights employ 4:2 compressors, the intermediate weights our approximative high-order compressors, and the lower weights inaccurate compressors (OR-tree based approximation).

(b)Simulation result of MAC unit:



RST and CLK are the input signals. The MAC unit's inputs are A and B. The output of the approximative multiplier is MUL_OP. The MAC unit's output is called MAC_OP. The accumulator register's code is ACC. The intermediate sum and carry bits are S1–S8 and C1–C8. Approximate multipliers are implemented by using the suggested approximate positive and negative compressors. Using an adder and an accumulator register, the multiplier output is accumulated. To reduce hardware costs while producing errors in opposing directions, two compressor types are first designed. Then, the associated approximation multipliers are thoroughly examined to forecast the amount of errors in a probabilistic manner. In contrast to earlier studies that suffered from errors that had accumulated over time, the suggested interleaving method introduces multipliers in a different way depending on the blending ratio, resulting in an error distribution that is balanced and narrow. The proposed method is better suitable for implementing approximate computing in various case studies, according to simulation results, successfully saving processing energy with little perceptible performance loss.

Approximate multiplier Comparison:

Parameters	Area(Gate count)	Delay(ns)	Power(mW)
APP MUL using 4:2	600	26.001	622.03
APP MUL using Multi-level Compressor	336	18.493	489.46

4.CONCLUSION

It is suggested to apply new approximate multipliers based on multi-level approximation compressors to lower the energy requirements of MAC-oriented signal

processing algorithms. To reduce hardware costs while producing errors in opposing directions, two compressor types are first designed. Then, the associated approximation multipliers are thoroughly examined to forecast the amount of errors in a probabilistic manner. The complexity of the existing design is then decreased by using approximate 5:2 and 6:2 compressors**.** The proposed interleaving method instead provides two simple multipliers according to the blending ratio, resulting in a narrow and balanced error distribution, as opposed to the earlier works, which suffered from the accumulation of errors. The proposed method is better suitable for implementing approximate computing in various case studies, according to simulation results, successfully saving processing energy with little perceptible performance loss.

REFERENCES

[1] P. N. Whatmough, S. K. Lee, H. Lee, S. Rama, D. Brooks, and G.-Y. Wei, "14.3 A 28 nm SoC with a 1.2 GHz 568 nJ/prediction sparse deep-neural-network engine with >0.1 timing error rate tolerance for IoT applications," in IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers, Feb. 2017, pp. 242–243.

[2] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-power digital signal processing using approximate adders," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 32, no. 1, pp. 124–137, Jan. 2013.

[3] M. Kazhdan, "An approximate and efficient method for optimal rotation alignment of 3D models," IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 7, pp. 1221–1229, Jul. 2007.

[4] M. J. Wainwright and M. I. Jordan, "Log-determinant relaxation for approximate inference in discrete Markov random fields," IEEE Trans. Signal Process., vol. 54, no. 6, pp. 2099–2109, Jun. 2006.

[5] J. Jo, J. Kung, and Y. Lee, "Approximate LSTM computing for energy efficient speech recognition," Electronics, vol. 9, no. 12, p. 2004, Nov. 2020.

[6] B. K. Mohanty, "Parallel VLSI architecture for approximate computation of discrete Hadamard transform," IEEE Trans. Circuits Syst. VideoTechnol., vol. 30, no. 12, pp. 4944–4952, Dec. 2020.

[7] X. Si et al., "15.5 A 28 nm 64Kb 6T SRAM computing-in-memory macro with 8b MAC operation for AI edge chips," in IEEE Int. Solid- State Circuits Conf. (ISSCC) Dig. Tech. Papers, Feb. 2020, pp. 246–248.

[8] H. Jiang, J. Han, and F. Lombardi, "A comparative review and evaluation of approximate adders," in Proc. 25th Ed. Great Lakes Symp. (VLSI), May 2015, pp. 343–348.

[9] C. Chen, S. Yang, W. Qian, M. Imani, X. Yin, and C. Zhuo, "Optimally approximated and unbiased floating-point multiplier with runtime configurability," in Proc. 39th Int. Conf. Comput.-Aided Design (ICCAD), Nov. 2020, pp. 1–9.

[10] V. Leon, K. Asimakopoulos, S. Xydis, D. Soudris, and K. Pekmestzi, "Cooperative arithmetic-aware approximation techniques for energy efficient multipliers," in Proc. 56th Annu. Design Autom. Conf. (DAC), Jun. 2019, pp. 1–6.