International Journal for Modern Trends in Science and Technology, 9(05): 809-815, 2023

ISSN: 2455-3778 online
DOI: https://doi.org/10.46501/IJMTST0905138
Available online at: http://www.ijmtst.com/vol9issue05.html

Approximate Adder Design Using Error Reduced Carry Prediction and Constan Truncation

G. Sindhura Bhargavi, G.V.Bala Subramanyam, G.Abhinay, A.V.Sathish, G.Surya Narayana, G.Johnson Arnold

Department of Electronics and Communication Engineering, Narayana Engineering College, Nellore, Andhra Pradesh, India

To Cite this Article
G. Sindhura Bhargavi, G.V.Bala Subramanyam, G.Abhinay, A.V.Sathish, G.Surya Narayana, G.Johnson Arnold. Approximate Adder Design Using Error Reduced Carry Prediction and Constan Truncation. International Journal for Modern Trends in Science and Technology 2023, 9(05), pp. 809-815. https://doi.org/10.46501/IJMTST0905138

Article Info
Received: 21 April 2023; Accepted: 20 May 2023; Published: 24 May 2023.

ABSTRACT

Now a days with the prevalence of battery-operated mobile and portable devices, power and energy consumption become the key constraint in system design because applications on these devices process a vast amount of computationally intensive information, such as multimedia (i.e., image, video, and audio) processing, deep learning, data mining, and recognition, under a limited power and energy budget. Many applications do not always require perfect computation accuracy. For example, multimedia processing that involves human senses is error-tolerant. In other words, humans usually do not perceive the output quality degradation caused by computation errors on these applications, and a certain level of errors can be acceptable. The limitation of human perception offers an opportunity for a new computing paradigm, approximate computing, trading computation accuracy for power and energy. The proposed approximate adder that exploits an error-reduced carry prediction and constant truncation with error reduction schemes. The proposed adder design techniques significantly improve overall computation accuracy while providing excellent hardware efficiency. Particularly, the proposed carry prediction technique can reduce a prediction error rate by up to 75% compared to existing approximate adders considered in this project. Furthermore, the error reduction technique also enhances the overall computation accuracy by decreasing the error distance (ED). Approximation errors caused by the proposed adder have very little impact on output quality when adopted in practical applications, such as digital image processing and machine learning.

KEYWORDS: approximate computing, error reduction, carry prediction.

1. INTRODUCTION

With the prevalence of battery-operated mobile and portable devices, power and energy consumption become the key constraint in system design because applications on these devices process a vast amount of computationally intensive information, such as multimedia (i.e., image, video, and audio) processing, deep learning, data mining, and recognition, under a limited power and energy budget. Many applications do not always require perfect computation accuracy. For example, multimedia processing that involves human senses is error-tolerant. In other words, humans usually do not perceive the output quality degradation caused by computation errors on these applications, and a certain level of errors can be acceptable. The limitation of human

perception offers an opportunity for a new computing paradigm, approximate computing, trading computation accuracy for power and energy. Because adders are fundamental arithmetic components in computing systems, the design of efficient approximate adders is a practical way to enable approximate computing. Therefore, it has gained remarkable attention from researchers and a significant number of approximate adder designs have been presented in the technical literature. We will review some existing approximate adders in Section II. Approximate adders can be classified as block-based and full adder (FA)-based designs. Block-based approximate adders split an entire adder into smaller multiple sub-adders that perform partial additions concurrently. The main idea of this approach is to cut a long carry propagation chain to achieve faster additions. However, it requires more area and power than FA-based approximate adders. FA-based adders use approximate 1-bit FAs to add some lower-order input bits approximately by replacing accurate FAs with approximate ones in the corresponding bit positions. This improves the area and power performance at the expense of the computation accuracy degradation.

In this paper, we propose a new approximate adder design based on new approximate FA cells, enhanced carry prediction, and a constant truncation with error reduction. The proposed carry prediction scheme significantly reduces the prediction error rate by up to 75% compared to existing approximate adders considered here. Also, the truncation with error reduction logic enhances the overall computation accuracy while reducing energy and power consumption.

In summary, this paper makes the following key contributions in designing approximate adders:

- We present a novel efficient approximate adder design that effectively trades off between hardware cost and computation accuracy through systematic analysis, and prove that our design outperforms the others by extensively comparing it with 12 approximate adders.
- We propose 1) a new carry prediction scheme that reduces the prediction error rate by up to 75% compared to the others, 2) approximate FA cells that improves accuracy, and 3) a constant truncation with an error reduction scheme that reduces hardware cost while offering good accuracy performance.

This way we are using different techniques at thesame time and also having high efficiency.

The remainder of this paper is organized as follows. Section II provides a brief review of existing approximate adders. In Section III, we present the proposed adder, which consists of our proposed approximate FAs, novel carry pre diction, and constant truncation with error reduction.

Illustrated examples of the adder operation and mathematical analysis of the error rate is also provided. Then, Section IV explains the experimental results and systematic analysis of the proposed adder as well as extensive comparison with the existing approximate adder.

2. LITERATURE REVIEW

The lower-part OR adder (LOA) and error tolerant adder I (ETAI) are two representative approximate adders implemented using an approximate FA for the least significant bits (LSBs) of a multibit adder [1], [7], and many of their variants were presented so far [2]– [6], [8], [9]. The proposed method is solely based on the LOA type approximate adders. The LOA consists of two parts: an accurate part and an inaccurate part [1]. The former part uses a traditional precise adder, such as the ripple carry adder (RCA) and carry-lookahead adder (CLA), to calculate the most significant bits (MSBs) with no computation error. Whereas, the latter part only uses an OR operation to approximately obtain LSB summations. Furthermore, the output of an AND operation for the MSB input pair of the inaccurate part is utilized as a carry input to the accurate part to improve overall computation accuracy. Design variants based on the LOA have been proposed to further optimize the LOA, such as LOA without the AND-based carry prediction (LOAWA), optimized lower-part constant OR

adder (OLOCA), hardware optimized and error reduced approximate adder (HOERAA), hardware optimized adder having a near-normal error distribution (HOAANED), and hybrid error reduction lower-part OR adder (HERLOA) [2]– [4]. The LOAWA is identical to the LOA, except for the AND-based carry prediction [2]. In other words, the carry input to the accurate partis fixed to a constant ‘‘0,’’ which degrades accuracy but improves the computation speed. The OLOCA is also similar to the LOA in that the OR operation is utilized for the inaccurate part approximation, but it outputs a constant ‘‘1’’ to a few LSBs

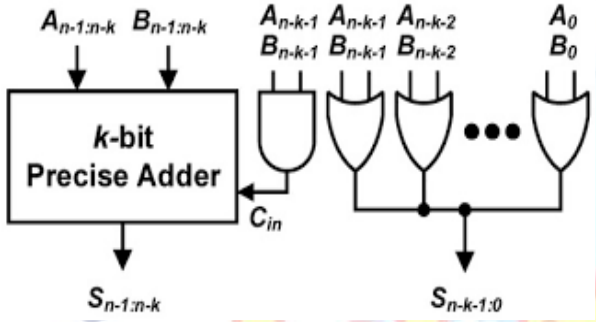

Figure 1 Lower part OR adder (LOA)

regardless of the corresponding bit inputs [3]. This also degrades accuracy a bit while reducing hardware cost. In addition to the OLOCA, the HOERAA uses the OR operation for two MSB input pairs of the inaccurate part and sets the remaining LSB outputs to a constant ‘‘1’’ regardless of the inputs [4]. For the MSB output of the inaccurate part, it uses a 2-to-1 multiplexer to select ‘‘0’’ or an OR operation output of the corresponding input pairs. The multiplexer output is then used in anOR operation with the AND gate output of thesecond MSB input pair of the inaccurate part. Also, it includes an AND-based carry prediction for the accurate part, which also serves as the selection input of the multiplexer. The HOAANED is derived from the HOERAA by including one additional OR gate at the MSB of the inaccurate part [5]. This OR gate contributes to the improvement of an error metric, and thus, the HOAANED produces outputswith almost normal error distributions. To enhance overall computation accuracy, the HERLOAcombines the basic LOA structure with the hybrid error reduction scheme [6]. figure 2 shows the architecture of the inaccurate part of the HERLOA.

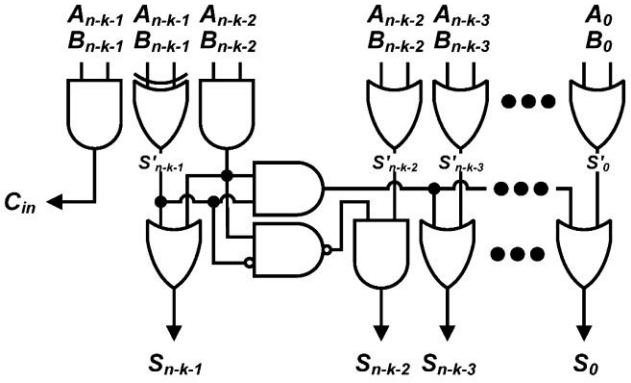

Figure 2 Hybrid error reduced lower part or adder(HERLOA)

Note that the accurate part is the same as the LOA (figure 1). When the second MSB input pair of the inaccurate part is both ‘‘1,’’ error reduction logic decreases the error distance (ED) by investigating the MSB input pair. The grayed gates in Figure 1 are the hybrid error reduction logic, while the others are the LOA logic. The error rate is reduced by replacing an OR gate at the MSB in the LOA with an XOR gate in the HERLOA.

3. PROPOSED METHOD

This section presents our proposed FA cell-based approximate adder, which exploits a novel carry prediction scheme and a constant truncation technique to reduce the ED and improve overall computation accuracy. We call our adder the error reduced carry prediction approximate adder (ERCPAA). We denote two n-bit input operands and one n-bit output of the adder as An−1:0, Bn−1:0, and Sn−1:0, respectively. Also, Ai , Bi , and Si represent the (i) th LSBs of An−1:0, Bn−1:0, and Sn−1:0, respectively.

OVERALL ADDER ARCHITECTURE

Figure 3 shows the overall hardware architecture of the proposed approximate adder with n-bit inputs. An n-bit adder is divided into two parts: a k-bit accurate part and an (n − k)-bit inaccurate part, where k < n. The accurate part simply consists of a k-bit precise adder that

produces an accurate output (i.e., Sn−1:n−k) from k MSB inputs (i.e., An−1:n−k and Bn−k:n−k) and a carry input (i.e.,Cin). The inaccurate part uses some of the remaining LSB inputs to generate an approximate output and a carry input to the precise adder. Note that the sizes of the accurate part and inaccurate part do not have to be equal, and the precise adder can be implemented in any type of traditional adders, such as RCA and CLA. The inaccurate part is further divided into three parts: an array of the proposed approximate FA cells, a carry prediction logic, and a constant truncation with error reduction logic. The proposed FA cell (see blue-highlighted box) simplifies the conventional single-bit FA cell to produce an approximate summation and an approximate carry, and is placed in some higher-order bit positions of the inaccurate part. The carry prediction logic, which is highlighted in green, generates the carry input to the precise adder. While most FA-based approximate adders employ an AND operation with the MSB inputs of the inaccurate part to produce the carry input, our prediction logic leverages the two MSB inputs to improve carry prediction accuracy at the cost of

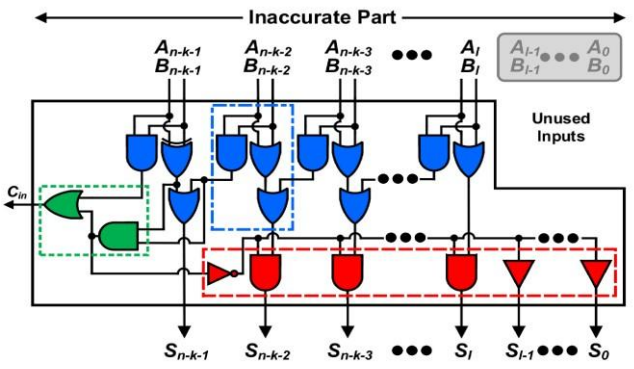

Figure 3 Inaccurate part of the ERCPAA

two additional logic gates. The constant truncation with error reduction logic highlighted in red sets l LSB outputs (i.e., Sl−1:0) to either a constant ''0'' or ''1'' to reduce hardware costs depending on input conditions. In other words, the l LSB inputs are not used to generate approximate summations. It also assigns the other output bits except for the MSB of the inaccurate part to a constant ''0'' to reduce the ED under certain input conditions. We will describe the condition to determine to fix the output bits to ''0'' or ''1'' with illustrative examples.

PROPOSED APPROXIMATE FULL ADDER

An FA is the key building block for carry propagate adders (e.g., RCA). The traditional 1-bit FA adds two inputs, Ai and Bi , as well as a carry from the previous bit position Ci−1 and produces asum Si and a carry output Ci using

$$Si = Ai \oplus Bi \oplus Ci-1 \quad (1)$$

$$Ci = AiBi + AiCi-1 + BiCi-1 \quad (2)$$

Although the FA requires two XOR gates to generate a sum, we replace the XOR gates with OR counterparts to do the same approximately to reduce hardware overhead in our approximate FA. In addition, the FA generates a carry output Ci from not only the two inputs, Ai and Bi, but also the carry from the previous bit position Ci−1. In other words, the carry of the previous bit position can be propagated to the next bit position through the current FA, resulting in a long critical path delay and degraded hardware performance in the carry propagate adders. To reduce the critical path delay and hardware overhead, we remove the dependency of the carry from the previous bit position to generate the carry output in our FA. Thus, the Boolean equations of our approximate FA are given by

$$Si,ERCPAA = Ai + Bi + Ci-1 \quad (3)$$

$$Ci,ERCPAA = AiBi \quad (4)$$

Consequently, the approximate part using the proposed FA cell does not form the carry propagationchain from the lower to the higher-order bit positions and thus the delay of the approximate part is consistent, although the size of the approximate part is larger (i.e., k decreases under a given n). Note that the MSB position of the inaccurate part has a different configuration of the FA, which uses an XOR gate instead of the OR gateto generate the sum of the two input operands Aiand Bi. This improves the overall computation TABLE 1. Truth table for traditional FA and

proposed approximate FAs.

accuracy since the XOR-based FA gives a more accurate sum, and it also allows the carry prediction logic to produce a more accurate carry input to the precise adder than the OR based FA. Table 1 depicts the truth table of the traditional and proposed FAs. The proposed FAs introduce errors if either of the operands is ''1'' and the carry of the previous bit position is ''1.''

The OR-based FA causes an additional error at TABLE 1. Truth table for traditional FA and proposed approximate FAs. sum Si when both operands are ''1'' and the carry of the previous bit position is ''1''.

CARRY PREDICTION

The accurate part can take a carry input generated from the inaccurate part to improve overall computation accuracy at the expense of a few logic gates. The AND-based carry prediction scheme, which has an error of approximately 25%, is widely adopted since it is easily implemented by performing an AND operation with the inaccurate part's MSB inputs (i.e., An−k−1 AND Bn−k−1) to produce the carry input to the precise adder. In the proposed prediction, only two additional gates (i.e., an AND gate and an OR gate in the green highlighted box in Figure 4) are utilized to produce the carry input with twice the prediction accuracy of the conventional AND-based one. Also, the inputs of the MSB and its previous bit position of the inaccurate part (i.e., An−k−1: n−k−2 and Bn−k−1: n−k−2) are exploited to predict the carry input. Let Pi denotes the propagate signal of the (i) th bit position, and the carry from the previous bit position Ci−1 is propagated to the carry output Ci if the propagate signal is ''1,'' defined as

$Pi = Ai \oplus Bi$ *(5)*

$Ci = Ci-1$ if $Pi = 1$ *(6)*

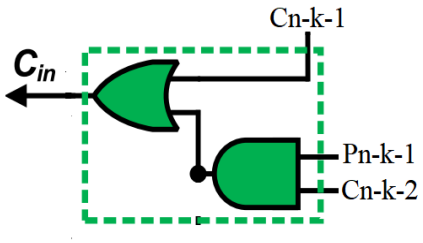

Figure 4 carry prediction logic

Since carry prediction scheme leverages the inputs of two-bit positions, a carry can be generated from either the (n−k −1) th or (n−k −2) th bit position. If a carry is produced in the (n − k − 1) th bit position, the carry input Cin is simply Cn−k−1. On the other hand, if a carry is generated in the (n − k − 2) th bit position, the carry Cn−k−2 should be propagated through (n − k − 1) th bit position to pass it to the accurate part.

Therefore, the carry input Cin is derived by

$Cin = Cn-k-1 + Pn-k-1Cn-k-2$ *(7)*

According to the above equation, one XOR, three AND, and one OR gates are required to generate the carry input Cin. Cn−k−1 and Cn−k−2 can be obtained from the proposed FAs in the corresponding bit positions and Pn−k−1 can also be calculated using the XOR gate of the FA in the MSB position of the inaccurate part. It is worth noting that one of the reasons to replace the OR with an XOR in the FA at the MSB is to generate a Pn−k−1 signal. Therefore, it only needs two additional gates (see green box in Figure 3.4) to implement the proposed carry prediction logic.

CONSTANT TRUNCATION

The proposed adder outputs a constant to a few LSBs to reduce hardware overhead by sacrificing overall accuracy slightly since the lower-order outputs have relatively less impact on the accuracy than higher-order outputs. Figures 5 and 6 exhibits an example of constant truncation operations with error reduction using the adder design parameters n = 16, k = 8, and l = 4. As shown in Figure 5, AFA sets the l LSB outputs to ''1'' regardless of the inputs of the corresponding bit positions. When a carry is generated from (n − k − 2) th bit position and then propagated through (n

– k – 1) th bit position, proposed adder performs error reduction. In short, the reduction is performed when Pn–k–1Cn–k–2 = 1. (n – k – 1) th bit position, proposed adder performs error

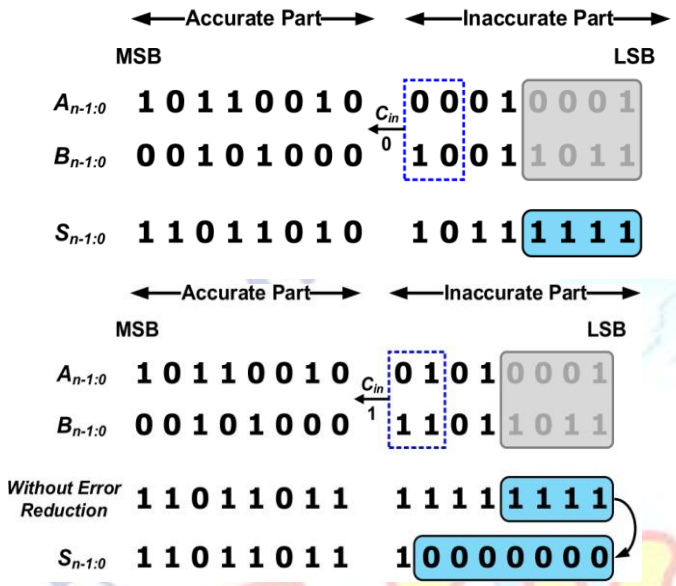

Figure 5 operations of ERCPAA: constant "1"

Figure 6 operations of ERCPAA: constant "0"

reduction. (n – k – 1) th bit position, proposed adder performs error reduction. In short, the reduction is performed when Pn–k–1Cn–k–2 = 1. Under this given input condition, the correct output of (n – k – 1) th bit is ''0,'' however, AFA produces ''1'' as the output at this bit position as shown in Figure 6.

4. EXPERIMENT RESULTS

EXPERIMENT SETUP

The proposed method for designing an approximate adder involves the use of Verilog HDL and synthesis with the Xilinx ISE, using the Spartan 3E family and XC3S250E device to examine the hardware characteristics of the adder in terms of area, delay, and power consumption. The method involves the implementation of a 16-bit adder using an 8-bit RCA-based precise adder, with n = 16 and k = 8. Prior studies have suggested that a size of 7 to 9 bits for the inaccurate part of the adder is appropriate for obtaining a good tradeoff between output quality and power and energy savings for practical applications, such as video and image processing. Since 16-bit adders are widely adopted in these applications, the design parameters of n = 16 and k = 8 were selected for the implementation.

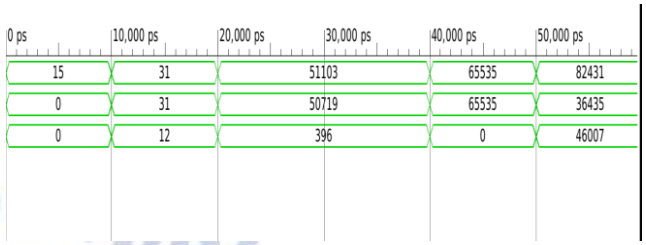

Figure 7 Simulation results of ERCPAA

PERFORMANCE COMPARISION

To determine whether the newly designed ERCPAA is superior to the previous version of LOA, namely HERLOA and ripple carry adder (RCA), comparison is necessary. Now comparison should be fair so the design HERLOA and RCA are implemented using the same process as ERCPAA. This will allow us to accurately assess the performance and characteristics of each design and determine which one is the most suitable for specific needs. The same family, same device, same parameters i.e., N=16, k=8 are used for the adders ERCPAA and HERLOA. 8-bit RCA is used as precise adder for both of them. Finally, they are compared with zero error precise adder RCA to understand about them more clearly.

Data on the area, delay, and power consumption for various approximate adders was collected fromfigures 6.1 to 6.17 and compiled into table 2.

Table 2 comparison between the adders

Sno	Area (LUT's)	Delay (ns)	Power(w)	Error (%)
ERCPAA	25	14.422	0.039	97
HERLOA	24	13.239	0.048	84
RCA	32	21.690	0.060	100

Based on the available information, it can be observed that ERCPAA has a smaller area (25 LUTs) and lower delay (14.422 ns) compared to HERLOA (24 LUTs and 13.239 ns, respectively).

However, ERCPAA has a higher error rate (97%) compared to HERLOA (84%). Both designs have relatively low power consumption, with ERCPAA consuming slightly less power (0.039 W) than HERLOA (0.048 W).

If the intended application does not require high accuracy, ERCPAA's advantages in area, delay, and power consumption may already make it the preferable option. However, the optimal design ultimately depends on the specific requirements and constraints of the application.

5. CONCLUSION

In conclusion, the ERCPAA (Error Reduced Carry Prediction and Constant Truncation) represents a significant step forward in the design of approximate adders. By leveraging the concept of error reduced carry prediction logic and constant truncation, the ERCPAA adder is able to achieve a remarkable level of error resilience while still maintaining a higher level of accuracy. This makes it an attractive option for applications where high accuracy is a critical requirement, but where a certain degree of reliability is still necessary. Compared to other approximate adders, the ERCPAA adder stands out for its superior error resilience mechanism, which is able to detect and correct errors at multiple levels of the computation. This not only improves its accuracy but also increases its robustness and reliability, making it suitable for a wide range of applications.

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

REFERENCES

[1] H. R. Mahdiani, A. Ahmadi, S. M. Fakhraie, and C. Lucas, ''Bio-inspired imprecise computational blocks for efficient

[2] VLSI implementation of softcomputing applications,'' IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 4, pp. 850–862, Apr. 2010.

[3] P. Albicocco, G. C. Cardarilli, A. Nannarelli, M. Petricca, and M. Re, ''Imprecise arithmetic for low power image processing,'' in Proc. Conf. Rec. 46th Asilomar Conf. Signals, Syst. Comput. (ASILOMAR), Nov. 2012, pp. 983–987.

[4] A. Dalloo, A. Najafi, and A. Garcia-Ortiz, ''Systematic design of an approximate adder: The optimized lower part constant-OR adder,'' IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 26, no. 8, pp. 1595–1599, Aug. 2018.

[5] P. Balasubramanian and D. L. Maskell, ''Hardware optimized and error reduced approximate adder,'' Electronics, vol. 8, no. 11, p. 1212, Oct. 2019.

[6] P. Balasubramanian, R. Nayar, D. L. Maskell, and N. E. Mastorakis, ''An approximate adder with a near-normal error distribution: Design, error analysis and practical application,'' IEEE Access, vol. 9, pp. 4518–4530, 2021.

[7] H. Seo, Y. S. Yang, and Y. Kim, ''Design and analysis of an approximate adder with hybrid error reduction,'' Electronics, vol. 9, no. 3, p. 471, Mar. 2020.

[8] N. Zhu, W. L. Goh, W. Zhang, K. S. Yeo, and Z. H. Kong, ''Design of low-power high-speed truncation-error-tolerant adder and its application in digital signal processing,'' IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 18, no. 8, pp. 1225–1229, Aug. 2010.

[9] Y. Kim, ''An accuracy enhanced error tolerant adder with carry prediction for approximate computing,'' IEIE Trans. Smart Process. Comput., vol. 8, no. 4, pp. 324–330, Aug. 2019.

[10] J. Lee, H. Seo, Y. Kim, and Y. Kim, ''Approximate adder design with simplified lower-part approximation,'' IEICE Electron. Exp., vol. 17, no. 15, pp. 1–3, Aug. 2020.

[11] A Critical Analysis of Approximate Adders: Correctness and Analysis of Performance by Ajay Kumar Gottem, Arunmetha S, Aravindhan Alagarsamy, Murali Krishna B

[12] Design and Analysis of an Approximate Adder with Hybrid Error Reduction Hyoju Seo 1, Yoon Seok Yang 2 and Yongtae Kim 1.