

Article

# COREA: Delay- and Energy-Efficient Approximate Adder Using Effective Carry Speculation

Hyelin Seok , Hyoju Seo, Jungwon Lee and Yongtae Kim \* 

School of Computer Science and Engineering, Kyungpook National University, Daegu 41566, Korea; tmzkd18518@knu.ac.kr (H.S.); hyoju@knu.ac.kr (H.S.); knuc17@knu.ac.kr (J.L.)

\* Correspondence: yongtae@knu.ac.kr

**Abstract:** This paper presents a delay- and energy-efficient approximate adder design exploiting an effective carry speculation scheme with error reduction. The proposed scheme reduces the delay and improves the energy efficiency without any significant accuracy degradation by effectively adding the predicted carry input using the OR operation. Additionally, the error reduction technique improves the overall computation accuracy at the expense of a few logic gates. As a result, the proposed adder achieves 3.84- and 7.79-times greater energy and energy-delay product (EDP) efficiencies than the traditional adder when implemented in 65-nm CMOS technology. In particular, when jointly analyzed with hardware accuracy, our design attains 69% and 70% reductions of the energy- and EDP-normalized mean error distance (NMED) products, respectively, compared to the other approximate adders under consideration. Furthermore, the proposed adder's efficacy over the existing adders is demonstrated by adopting it in a machine learning application.

**Keywords:** approximate adder; approximate circuit; approximate computing; arithmetic circuit; energy-efficiency; low-power; carry speculation; error reduction



**Citation:** Seok, H.; Seo, H.; Lee, J.; Kim, Y. COREA: Delay- and Energy-Efficient Approximate Adder Using Effective Carry Speculation. *Electronics* **2021**, *10*, 2234. <https://doi.org/10.3390/electronics10182234>

Academic Editors: Gaetano Palumbo and Akash Kumar

Received: 6 August 2021  
Accepted: 9 September 2021  
Published: 12 September 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

To date, energy-efficiency has been the primary growing concern for designing modern computing systems, especially battery-operated electronic devices. This is because the increasing density and complexity of state-of-the-art VLSI systems require tremendous power and energy to perform demanding tasks, such as digital signal processing (DSP) and machine learning [1–5]. One key observation is that many of these tasks do not require stringent accuracy in their computations. For example, an image with some noise and loss processed by an image compression algorithm can still be recognized by human vision. Therefore, to tackle this exceptional energy-efficiency challenge, approximate computing has emerged as an alternative design paradigm [6]. The main objective of this approximation is to reduce hardware resource consumption with acceptable output quality for achieving overall energy-efficiency. The approximate computing technique can be found at both hardware and software layers. As the arithmetic units, particularly adder, are the primary and power-hungry building blocks at the hardware layer, the design of an efficient approximate adder has attracted significant attention from researchers [7]. In this regard, we focus on the energy-efficient approximate adder design.

A significant number of approximate adders has been presented in the literature [8–25]. One of the major techniques in designing approximate adders is to split an adder into two parts: accurate and inaccurate parts. The accurate part includes a precise adder, such as a ripple carry adder (RCA) and carry lookahead adder (CLA), to correctly add the higher-order input bits. The inaccurate part leverages its own approximation logic, such as OR and XOR, to produce approximate outputs for lower-order bits. This adder architecture makes approximation errors concentrate on the lower-order output bits (i.e., less significant bits), resulting in limited error distances. The lower-part OR adder (LOA) is one of the most representative adders based on this split architecture [8]. Its approximate part adopts

the OR gate to imprecisely add the lower-order input bits and the most significant bit (MSB) input pairs of the part are exploited to generate a carry input signal by an AND operation with the pair for the accurate part where the correct addition with the carry occurs. The error tolerant adder I (ETAI) presented in [9] also adopts the same architecture and so does the approximate mirror adder 5 (AMA5), which is the only one implemented at gate-level for five AMAs proposed in [10]. The ETAI and AMA5 leverage the modified XOR and mirror operations, respectively, for their inaccurate parts. Another main difference arises from the carry prediction scheme where the ETAI excludes the prediction, but the AMA5 utilizes the one from the inaccurate part's MSB input pair as the carry for the accurate part.

Additionally, the design variants based on the LOA and ETAI have been proposed to optimize their original designs further [11–13]. For example, the optimized lower-part constant OR adder (OLOCA), hybrid error reduction LOA (HERLOA), and simplified ETA (SETA) are presented. The OLOCA and HERLOA are based on the LOA architecture; however, they have different approximation schemes [11,12]. The former sets some output bits of its inaccurate part to “1” regardless of the corresponding input bits to reduce the hardware resource consumption by sacrificing accuracy. However, the latter employs a hybrid error reduction scheme to enhance the error characteristics with little increased hardware cost. The SETA simplifies the ETAI's approximation to improve the hardware efficiency without a significant accuracy loss [13]. In addition, the hardware optimized and error reduced approximate adder (HOERAA) and hardware optimized adder having a near-normal distribution (HOANED) also employ a constant truncation scheme in which some outputs of the LSBs are set to “1” [14,15]. They employ only two input pairs of their inaccurate part to produce the approximation outputs, and their differences can be observed in the OR gate of the HOANED's inaccurate part. This OR gate enhances an error characteristic that makes the adder outputs follow almost near-normal distribution. Moreover, the lower-part zero truncation adder (LZTA) also employs the constant truncation scheme, with the key difference from the other constant scheme-based adders being that the entire output bits of its inaccurate part are set to all constant “0” instead of “1” and an OR-based carry prediction is used for its precise adder [16].

In this paper, we present an energy-efficient approximate adder leveraging an effective carry speculation scheme with error reduction. The proposed carry speculation scheme does not increase the critical path delay to add the predicted carry input without any significant computation accuracy loss. This offers a remarkably enhanced energy-efficiency of the proposed adder compared to other approximate adders. The proposed adder outperforms other existing adders for energy and energy-delay product (EDP) while offering excellent error characteristics. Specifically, the proposed adder is 3.84 and  $7.79 \times$  more energy- and EDP-efficient than a traditional adder when implemented in 65-nm CMOS technology. The main contributions of this paper are as follows:

- We propose a novel approximate adder that offers excellent energy-efficiency with high accuracy.
- We systematically analyze the proposed adder for error characteristics and hardware performance.
- We extensively compare the proposed adder with other adders using various aspects, including hardware-accuracy joint metrics.
- We present the efficacy of the proposed adder over existing approximate adders in a machine learning application.

The remainder of this paper is organized as follows. Section 2 presents the proposed adder architecture consisting of effective carry prediction with error reduction, and provides illustrative examples for the operation and mathematical error analysis. Section 3 explains the experimental results and comparison with the existing adders using various hardware, accuracy, and joint metrics. In Section 4, we present a case study, such as  $k$ -means clustering using various adders, to demonstrate the efficacy of the proposed adder. Finally, Section 5 presents the conclusion.

## 2. Proposed Approximate Adder Design

This section presents the proposed approximate adder that effectively adds the speculated carry using an OR operation and performs error reduction under a certain input condition, termed a carry OR error reduced adder (COREA). Let  $A_{n-1:0}$ ,  $B_{n-1:0}$ ,  $S'_{n-1:0}$ , and  $S_{n-1:0}$  denote  $n$ -bit two input operands, intermediate, and final outputs of the adder, respectively, and  $A_i$ ,  $B_i$ ,  $S'_i$ , and  $S_i$  denote their  $(i)^{th}$  LSBs.

### 2.1. Proposed Adder Architecture

Figure 1 shows the overall hardware architecture of the proposed adder. The  $n$ -bit adder comprises a  $k$ -bit accurate part and a  $(n - k)$ -bit inaccurate part, where  $k < n$ . The accurate part adds the high-order  $k$ -bit inputs accurately using a  $k$ -bit precise adder and produces the upper sum (i.e.,  $S_{n-1:n-k}$ ) and carry output (i.e.,  $C_{out}$ ). Note that the precise adder can be implemented using any traditional accurate adder, such as RCA and CLA. The latter part adds the rest of the inputs to produce the approximate sum (i.e.,  $S_{n-k-1:0}$ ) and carry input for the accurate part (i.e.,  $C_{in}$ ).

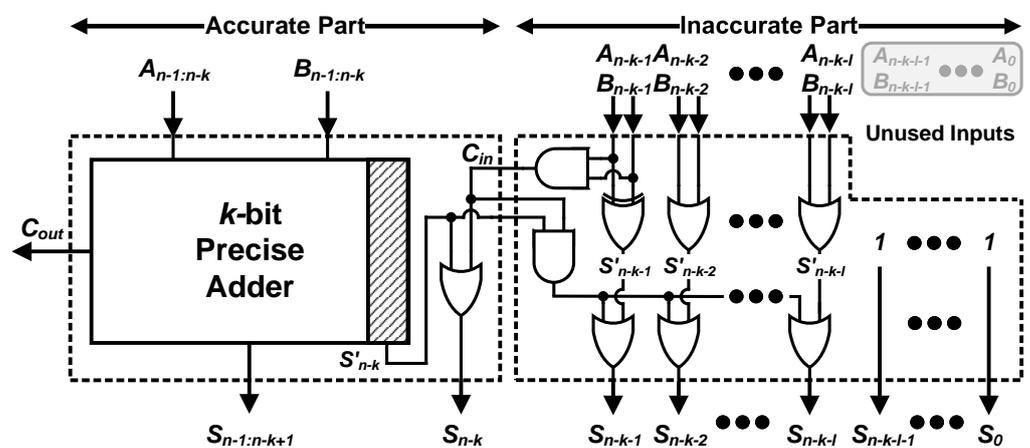


Figure 1. Overall architecture of the proposed adder, carry OR error reduced adder (COREA).

The carry input is generated by an AND operation of the inaccurate part’s MSB input pair. While the LOA and its variants fed the carry into the precise adder directly, the proposed adder uses only an OR operation of the carry and precise adder’s LSB output to add the carry and produce the final LSB output (i.e.,  $S_{n-k} = C_{in} \text{ OR } S'_{n-k}$ ). Therefore, the LOA and its variants require an additional delay to add the carry. However, the proposed scheme reduces the critical path delay, resulting in improved energy-efficiency while degrading the accuracy slightly. Furthermore, this OR-based carry handling scheme also reduces the area and power since the precise adder does not require any logic to add the carry at its LSB position. For example, the RCA-based precise adder requires a full adder (FA) at its LSB to take the carry, whereas this scheme allows the precise adder to necessitate only a half adder (HA) at the LSB due to no carry being fed into the adder.

The inaccurate part is based on the OR operation and constant truncation. This part adds the upper  $l$ -bit inputs by OR gates, except for its MSB where the XOR gate that forms a HA is used to improve overall computation accuracy. The remaining  $(n - k - l)$ -bit inputs are not used, and the corresponding output bits are set to “1” to reduce hardware resource without any significant accuracy degradation. Because the proposed OR-based carry handling causes an incorrect LSB output of the accurate part under a certain input condition, the adder performs error reduction using additional OR gates. It is worth noting that these OR gates do not affect the output results when the LSB output is correct. We will describe the input condition that requires the error reduction by providing illustrative examples in the following section.

### 2.2. Operation of the Proposed Adder

Figure 2 shows operation examples of the proposed adder with the design parameters of  $n = 16, k = 8,$  and  $l = 4$ . As shown in Figure 2a, the precise adder of the accurate part adds  $k$  MSB inputs without any carry input and produces the intermediate output  $S'_{n-1:n-k}$ . Then, the precise adder's LSB output is OR-ed with the predicted carry from the inaccurate part to produce the final output  $S_{n-1:n-k}$ , which is the correct result in which the carry is properly added. Thus, the carry and no error reduction is required. This result shows that the OR operation effectively adds the carry at the LSB without any delay increase. The inaccurate part performs XOR and OR operations for its upper four output bits with the constant truncation to "1" for its lower counterparts as described in Section 2.1.

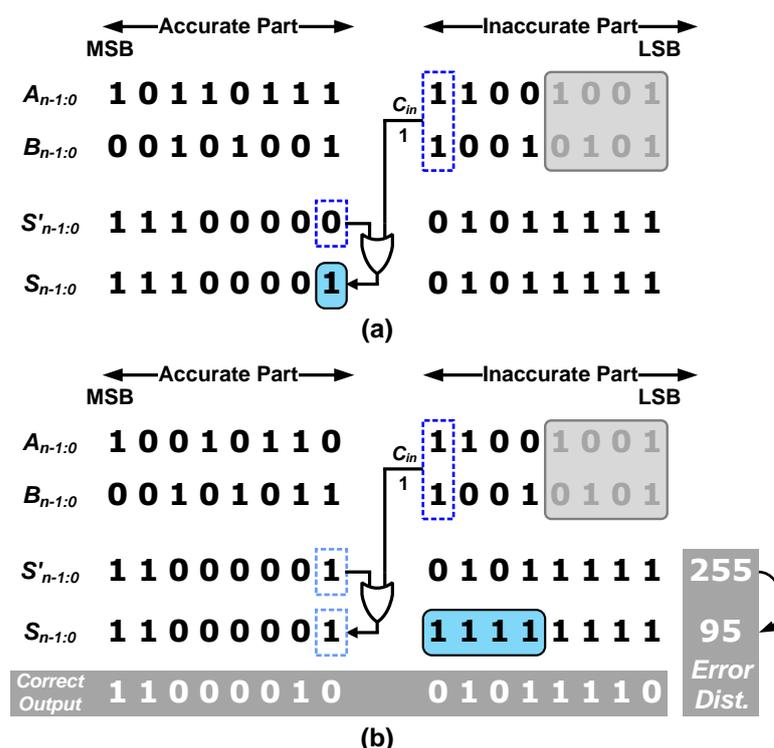


Figure 2. Operations of the proposed adder when (a)  $C_{in} = 1$  and  $S'_{n-k} = 0$  and (b)  $C_{in} = 1$  and  $S'_{n-k} = 1$ .

Unlike the above example with  $C_{in} = 1$  and  $S'_{n-k} = 0$ , the error reduction needs to perform to reduce the error distance further when  $C_{in} = 1$  and  $S'_{n-k} = 1$ . As shown in Figure 2b, if the intermediate LSB output is "1", the OR-based carry handling does not affect the final output at all, resulting in the incorrect LSB value. To make the approximation output closer to the correct output, the error reduction logic forces the inaccurate part's upper output bits to all "1" using the OR gates described in Figure 1. Under the given input in Figure 2b, the error distance, defined by the value difference between the approximate and correct outputs in absolute, is reduced from 255 to 95. This error reduction scheme leads to up to a  $2^{n-k} - 2^{n-k-l}$  decrease in the error distance. Note that we considered the condition  $C_{in} = 1$ , but the OR operation for the carry and error reduction does not affect the final output when  $C_{in} = 0$ . Thus, the intermediate output becomes the final output.

### 2.3. Error Rate Analysis

The error rate is one of the essential error metrics for characterizing approximate adders. To formulate the error rate of the proposed adder, we first define events of input conditions, where the adder always produces the correct outputs. Then, we calculate the error rate by the complement probabilities of the events. We consider two events where the adder generates correct outputs according to the accurate part's LSB output bit (i.e.,

$S_{n-k} = 1$  or  $S_{n-k} = 0$ ). When  $S_{n-k} = 1$ , the proposed adder generates the correct results if  $A_i \neq 1$  and  $B_i \neq 1$  where  $n - k - 1 < i < n - k - l$  and  $A \neq B$  where  $n - k - l - 1 < i < 0$ . Therefore, an event  $E_{CO,S_{n-k}=1}$  that the outputs are correct when  $S_{n-k} = 1$  is formulated as follows:

$$E_{CO,S_{n-k}=1} = \prod_{i=n-k-l}^{n-k-1} (A_i \bar{B}_i + \bar{A}_i B_i + \bar{A}_i \bar{B}_i) \cdot \prod_{i=0}^{n-k-l-1} (A_i \bar{B}_i + \bar{A}_i B_i) \quad (1)$$

We assume that the two input operands A and B are bitwise independent. Then, the probability of this event under random inputs is given by

$$\begin{aligned} \mathbf{P}(E_{CO,S_{n-k}=1}) &= \mathbf{P}\left(\prod_{i=n-k-l}^{n-k-1} (A_i \bar{B}_i + \bar{A}_i B_i + \bar{A}_i \bar{B}_i)\right) \mathbf{P}\left(\prod_{i=0}^{n-k-l-1} (A_i \bar{B}_i + \bar{A}_i B_i)\right) \\ &= \left(\frac{3}{4}\right)^l \left(\frac{1}{2}\right)^{n-k-l} \end{aligned} \quad (2)$$

When  $S_{n-k} = 0$ , it means the MSB output of the adder's inaccurate part (i.e.,  $S_{n-k-1}$ ) will always be correct regardless of the input operands of the corresponding bit position. The rest of the output bits (i.e.,  $S_{n-k-2:0}$ ) are correct if the input conditions of the corresponding bit position are the same as  $E_{CO,S_{n-k}=1}$ . Then, an event  $E_{CO,S_{n-k}=0}$  in which the outputs are correct when  $S_{n-k} = 0$  is similarly defined, and its probability is calculated as  $\mathbf{P}(E_{CO,S_{n-k}=0}) = (3/4)^{l-1} (1/2)^{n-k-l}$ . Since the probability to be  $S_{n-k} = 1$  and  $S_{n-k} = 0$  is identical and they are mutually exclusive, the error rate of the proposed adder  $\mathbf{ER}_{\text{COREA}}$  is calculated by the complement probabilities of the two events as follows:

$$\begin{aligned} \mathbf{ER}_{\text{COREA}}(n, k, l) &= 1 - \frac{1}{2} (\mathbf{P}(E_{CO,S_{n-k}=1}) + \mathbf{P}(E_{CO,S_{n-k}=0})) \\ &= 1 - \left(\frac{7}{8}\right) \left(\frac{3}{4}\right)^{l-1} \left(\frac{1}{2}\right)^{n-k-l} \end{aligned} \quad (3)$$

### 3. Experimental Results

The proposed approximate adder was designed by structural and gate-level modeling in Verilog-HDL and synthesized with commercial 65-nm CMOS technology and the standard cell library to analyze its circuit characteristics, such as area, delay, power, and energy [26]. The earlier works revealed that the approximation of the range of 7 to 9 LSBs offers acceptable processing quality with great power and energy saving for digital image and video processing applications, where 16-bit adders are mainly used [10,21,27,28]. Thus, a 16-bit adder divided into two identically-sized accurate and inaccurate parts was implemented (i.e.,  $n = 16$  and  $k = 8$ ). Additionally, an RCA-based precise adder was employed in the accurate part [10–12].

To evaluate the accuracy performance of the proposed adder, a software-based simulation was conducted to extract various error metrics, such as error rate, mean error distance (MED), normalized MED (NMED), and mean relative error distance (MRED). These metrics were obtained by applying 10 million (i.e.,  $10^7$ ) uniformly generated random input pairs to the adder.

#### 3.1. Performance Analysis

The hardware performance and accuracy of the proposed adder vary according to the design parameter  $l$ . Particularly, the area, power, and energy increase as  $l$  increases under a given  $n$  and  $k$  because a larger  $l$  requires more logic gates for the adder. Note that the delay remains constant because it is affected by the other design parameters  $n$  and  $k$ .

Figure 3 shows the performance analysis of the proposed adder with different values of  $l$ . Under the given  $n = 16$  and  $k = 8$ , we adjusted  $l$  from 1 to 7, which prevents the approximate output from being all constant bits (i.e.,  $l = 0$ ) or all non-constant bit (i.e.,

$l = 8$ ). As expected, the area, power, and energy linearly increase as  $l$  increases. The area increases more rapidly than the power and energy since the area, power, and energy increase by 27%, 17%, and 17%, respectively, when  $l$  increases from 1 to 7. The error rate improves as  $l$  increases because the OR-based approximation impacts more on the overall outputs than the constant truncation in the higher value of  $l$ . In addition, the line of Equation (3) is plotted to prove the correctness of the derived error rate formula. The line perfectly matches the simulated error rate at various values of  $l$ . Unlike the error rate, the accuracy performance in terms of NMED and MRED is not incrementally enhanced as  $l$  increases. The NMED and MRED values were normalized using the corresponding value of the adder with  $l = 1$  to effectively compare them with different  $l$ . The proposed adder's NMED and MRED show an almost identical trend according to  $l$ . The NMED and MRED sharply decrease from  $l = 1$  to  $l = 3$  and gradually increase after  $l = 4$ . Therefore, the best accuracy was made at  $l = 3$ . Note that the lower NMED and MRED values represent better accuracy.

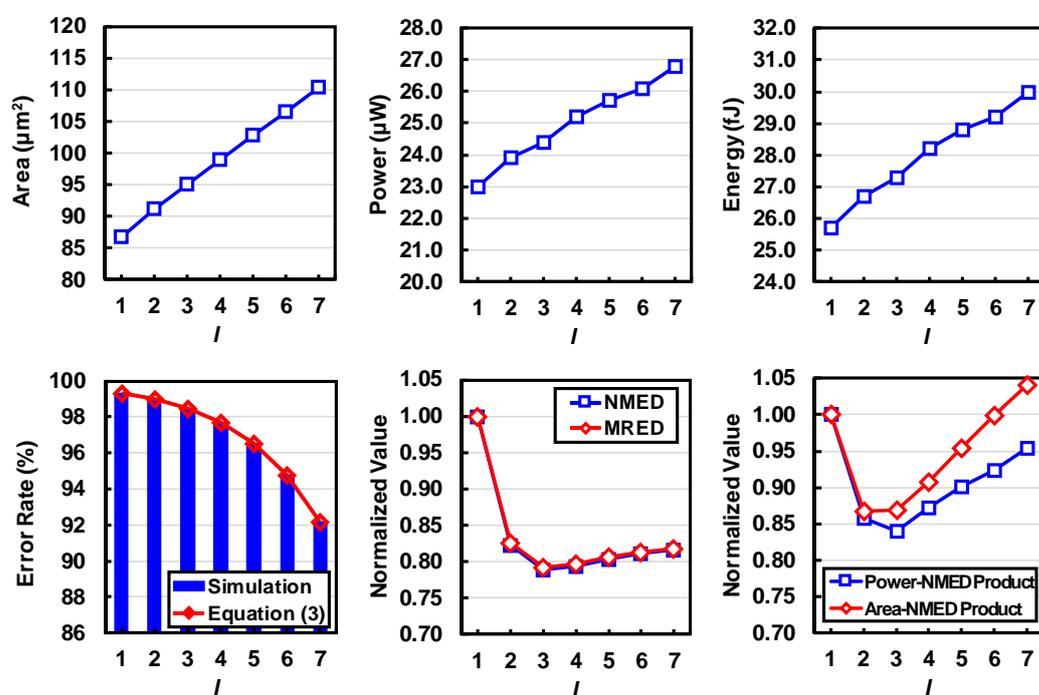


Figure 3. Performance analysis of the proposed adder under various values of  $l$ , ranging from 1 to 7.

To determine the best tradeoff between the hardware and accuracy performance of the proposed adder, the hardware-accuracy joint metrics can be considered. The power-NMED product was suggested in [29] to assess the power and accuracy collectively. Similarly, an area-NMED product can be defined. In fact, we also considered MRED-involved joint metrics; however, they were excluded since the proposed adder shows almost the same trend in NMED and MRED. The power- and area-NMED products with respect to  $l$  are also shown in Figure 3, and the values are normalized as well. The proposed adder shows the best power-NMED product value at  $l = 3$ , and its area-NMED product values at  $l = 2$  and  $l = 3$  are the same. This result recommends that setting the lower five output bits to “1” achieves the best tradeoff performance at the given  $n$  and  $k$ . Therefore, we will use the proposed adder configuration with  $n = 16$ ,  $k = 8$ , and  $l = 3$  for comparison with other approximate adders.

### 3.2. Performance Comparison with Other Approximate Adders

To compare the hardware resource consumption of the proposed adder and other adders, we also designed an accurate adder (RCA) and the nine existing approximate adders based on the same split architecture (AMA5, LOA, OLOCA, HOERAA, HOANED,

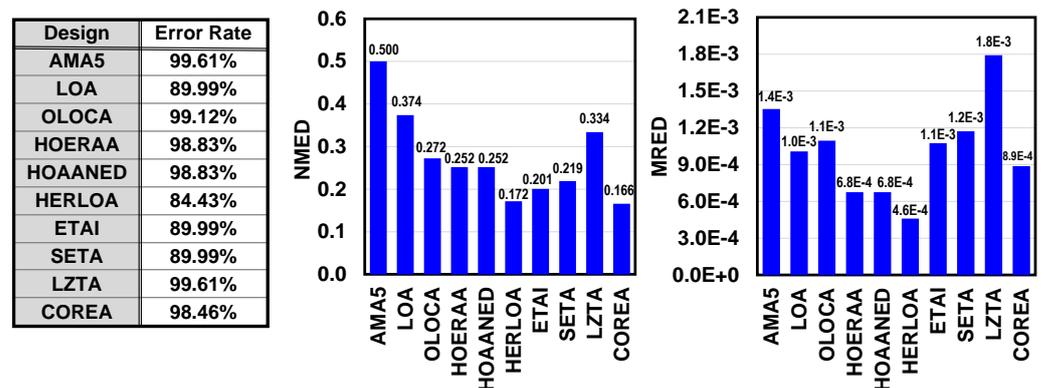
HERLOA, ETAI, SETA, and LZTA) by the same design methodology. For fair comparisons, we used the same 65-nm CMOS technology and standard cell library to synthesize them, which are 16-bit adders with an 8-bit RCA-based precise adder, using Synopsys Design Compiler. While the ETAI presented in [9] involves some transistor level design of the control logic, it can be implemented by gate-level design and, thus, we designed the ETAI by the same structural and gate-level modeling [22]. The OLOCA with the design parameter  $l = 2$  was implemented [11]. The error metrics were obtained by applying the identical input pairs to the adders except for the RCA.

Table 1 summarizes the hardware performance of various adders in terms of area, delay, power, energy, area-delay product (ADP), and EDP. The RCA requires a FA in each bit position, and many FAs are necessary to build a multi-bit RCA, leading to the largest area occupation and power consumption among the adders. Furthermore, the longest delay stems from the bit-by-bit carry propagation from the LSB to MSB. The greatest area, delay, energy, and power consumption causes the worst ADP and EDP performance. The LZTA occupies the smallest area, leading to the lowest ADP value owing to its simple structure for the approximate part, whereas the ETAI has the largest. The OLOCA is the second-best in area and ADP. The AMA5, HOERAA, HOANNED, SETA, and the proposed adder COREA occupy a similar area, slightly larger than the OLOCA, whereas the area of the HERLOA is almost the same as that of the ETAI. The accurate parts of the ETAI and SETA do not take any carry input from the inaccurate part, and this lack of the carry prediction makes them the fastest adders. On the other hand, the proposed adder delay is the same as that of the ETAI and SETA, although its accurate part uses the AND-based carry input. To avoid increasing the proposed adder delay, it effectively adds the incoming carry at the accurate part LSB by ORing of the carry and the precise adder's LSB output. The LOA, OLOCA, HOANNED, and HERLOA have the same delay because they adopt the identical AND-based carry prediction, and the AMA5's delay is slightly lower than their delay due to the use of one from its inaccurate part's MSB input pair as the carry. The LZTA's slightly longer delay than theirs stems from the OR-based carry prediction scheme. While the LZTA dissipates the lowest power, the HERLOA is the largest among the approximate adders. The power shows a similar trend with the area. The proposed adder's shortest delay leads to excellent performance of the energy and delay-involved products, whereas the HERLOA has the worst values for these metrics. For example, the proposed adder is the best in energy and EDP together with the SETA, while it shows better area and ADP performance than the SETA. Also, our adder shows the second-best ADP, which is only 2.9% larger than that of the LZTA.

Figure 4 shows the accuracy performance comparisons in error rate, NMED, and MRED aspects. The error rate, NMED, and MRED values show different trends. For example, the proposed adder COREA shows one of the worst adders in error rate perspective, but it is the best in NMED and has a moderate MRED value. The AMA5, OLOCA, HOERAA, HOANNED, LZTA, and proposed adder generate over 98% errors on their additions due to few LSB outputs are fixed to a constant value or one of each corresponding input pair. The LOA, SETA, and ETAI have an identical error rate of 89.99%, and the HERLOA produces the lowest error rate of 84.43%. While the AMA5 has the worst NMED value, the proposed adder does the best. The OLOCA, HOERAA, and HOANNED have a similar NMED value and the HERLOA's NMED value is close to that of the proposed adder. The NMEDs of the ETAI and SETA are in between those of OLOCA/HOERAA/HOANNED and HERLOA. The HERLOA shows the best MRED performance, whereas the LZTA is the worst. The MREDs of the LOA, OLOCA, ETAI, and SETA show similar results, and that of the AMA5 is slightly larger than them.

**Table 1.** Hardware performance summary of various 16-bit adders.

Design	Area ( $\mu\text{m}^2$ )	Delay (ns)	Power ( $\mu\text{W}$ )	Energy (fJ)	ADP ( $\mu\text{m}^2 \cdot \text{s}$ )	EDP (fJ $\cdot$ s)
RCA	162.6	2.27	46.2	104.9	$3.69 \times 10^{-7}$	$2.38 \times 10^{-7}$
AMA5	94.7	1.17	25.0	29.3	$1.11 \times 10^{-7}$	$3.43 \times 10^{-8}$
LOA	101.8	1.18	25.5	30.0	$1.20 \times 10^{-7}$	$3.55 \times 10^{-8}$
OLOCA	90.2	1.18	24.1	28.4	$1.06 \times 10^{-7}$	$3.35 \times 10^{-8}$
HOERAA	94.1	1.19	24.9	29.6	$1.12 \times 10^{-7}$	$3.52 \times 10^{-8}$
HOANNED	93.1	1.18	24.9	29.5	$1.10 \times 10^{-7}$	$3.47 \times 10^{-8}$
HERLOA	113.0	1.18	28.8	33.9	$1.33 \times 10^{-7}$	$4.01 \times 10^{-8}$
ETAI	113.3	1.12	27.2	30.4	$1.27 \times 10^{-7}$	$3.42 \times 10^{-8}$
SETA	97.6	1.12	24.4	27.3	$1.09 \times 10^{-7}$	$3.06 \times 10^{-8}$
LZTA	86.4	1.19	23.6	28.1	$1.03 \times 10^{-7}$	$3.34 \times 10^{-8}$
COREA	95.0	1.12	24.4	27.3	$1.06 \times 10^{-7}$	$3.06 \times 10^{-8}$

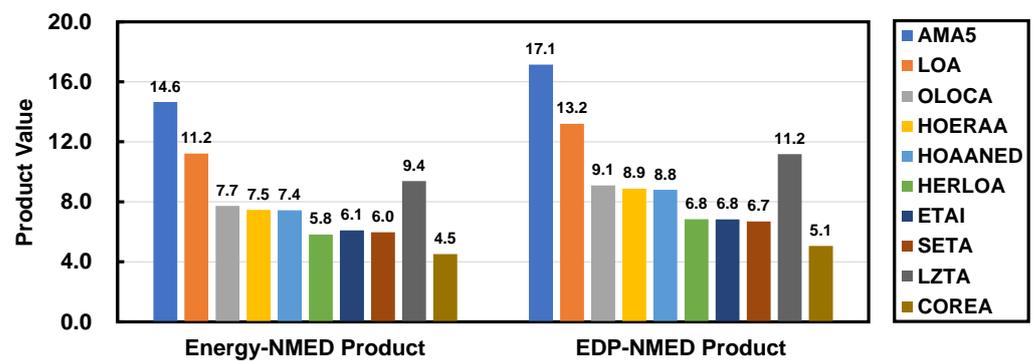
**Figure 4.** Comparisons of error rate, normalized mean error distance (NMED), and mean relative error distance (MRED) of approximate adders.

### 3.3. Tradeoff Analysis and Comparison

In addition to the power-NMED product in [29], energy- and EDP-NMED products were introduced to demonstrate tradeoff performance between energy-efficiency and computation accuracy for approximate adders [12,23].

Figure 5 exhibits the two products of the nine existing approximate adders and the proposed adder. Obviously, the proposed adder outperforms all other approximate adders, whereas the AMA5 has the largest value of each product. Specifically, the energy- and EDP-NMED products of the proposed adder are 69% and 70% smaller than those of the AMA5, respectively. Although the AMA5's energy and EDP performance are better than the LOA, HOERAA, HOANNED, and HERLOA, poor accuracy deteriorates its tradeoff performance, resulting in larger product values than them. The OLOCA, HOERAA, and HOANNED have almost identical product values and so do the HERLOA, ETAI, and SETA; however, the values of the LOA and LZTA are between those of the AMA5 and OLOCA.

In summary, the results confirm that the proposed adder is found to have the best hardware-accuracy tradeoff performance among the approximate adders considered herein. Specifically, energy- and EDP-NMED products of the proposed adder are 69% and 70% less than those of the AMA5, respectively.



**Figure 5.** Comparisons of energy-normalized mean error distance (NMED) product and energy-delay product-NMED (EDP-NMED) product of approximate adders.

#### 4. Case Study

To assess the efficacy of the proposed approximate adder in practical applications, we applied our adder design to a machine learning algorithm where addition and subtraction are heavily performed. In particular, we considered  $k$ -means clustering. The other approximate adders were also adopted in the same application to compare their performance. We used the accurate adder to obtain the golden reference for the application.

$k$ -means clustering is one of the most popular unsupervised machine learning algorithms, which is widely used for cluster analysis in data mining, such as image classification. The objective of the  $k$ -means is to group similar data points by dividing the data into different categories to analyze underlying patterns. Here,  $k$  is the number of cluster centroids, each of which is the location representing the center of the corresponding cluster in the dataset. The algorithm takes an unlabeled dataset and partitions all data points of the set into  $k$  clusters. When clustering, every data point is allocated to each cluster by reducing the within-cluster sum of squares (WCSSs). The WCSS value is the sum of the distances between each data point and the centroids, and we applied the approximate adders to calculate the WCSS value for the clustering [25]. We considered an unlabeled dataset containing 1000 data points with  $k = 5$  in [30].

Figure 6 illustrates the original dataset and  $k$ -means clustering outputs using the accurate and approximate adders as a 2D visualized form. We also inserted the WCSS values below each result using the corresponding adder to analyze the clustering quality. A lower WCSS value means better processing quality, and we used the WCSS value of the clustering produced by the accurate adder as the golden reference [25]. The LZTA shows the worst clustering result in terms of WCSS, and its value is  $3.11 \times$  greater than the one produced by the accurate adder. In addition, the ETAI produces slightly better WCSS value than the LZTA, which are still  $2.34 \times$  greater than the one produced by the accurate adder. The AMA5 and SETA yield better clustering qualities, but their results are still much different from the golden reference. The LOA and OLOCA exhibit a similar quality of the clustering result. While the proposed adder achieves the best clustering result and its WCSS is only 2.11% greater than that of the golden reference, the outputs using the HOERAA, HOANED, and HERLOA are close to the one using the proposed adder.

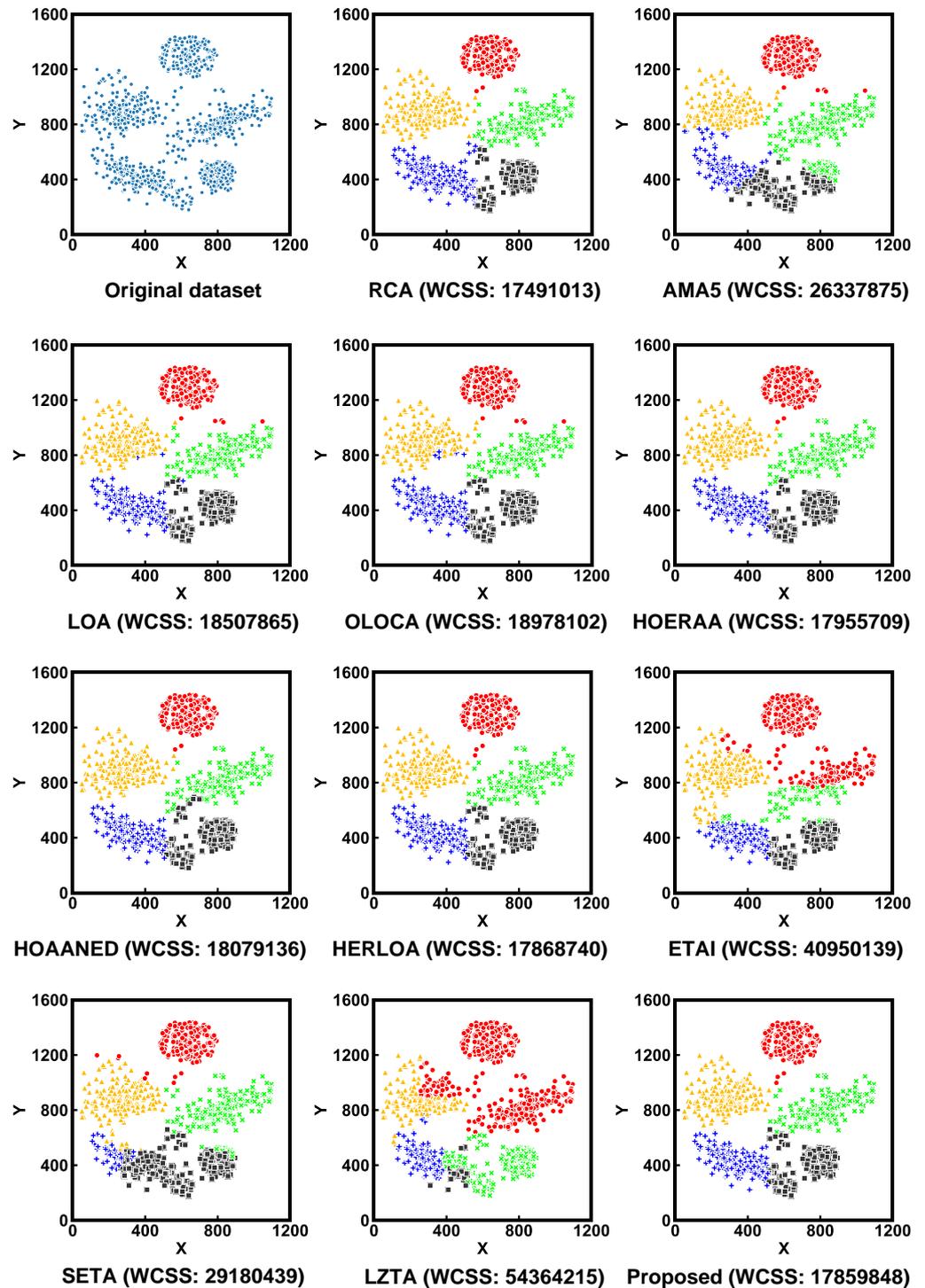


Figure 6. Original dataset and  $k$ -means clustering outputs produced using accurate and approximate adders.

To sum up, the proposed adder COREA outperforms the other approximate adders in  $k$ -means clustering algorithm. It is worth noting that in addition to the excellent performance in the practical application, the proposed adder demonstrated the significantly reduced hardware resource consumption, such as delay, energy, and EDP (see Table 1).

## 5. Conclusions

In this paper, we have presented the design of an energy-efficient approximate adder leveraging the effective carry speculation with error reduction. The incoming carry generated by the inaccurate part is OR-ed with the LSB output of the accurate part to reduce the delay. Additionally, the error reduction scheme improves the computation accuracy under a certain input condition at the cost of a few logic gates. The proposed design has been designed and synthesized using 65-nm CMOS technology and was found to be  $3.84\times$  and  $7.79\times$  more energy- and EDP-efficient than the RCA. Moreover, the proposed adder achieves 69% and 70% reductions in the energy- and EDP-NMED products, respectively, compared to the existing approximate adders. As a case study, the proposed adder has been adopted in  $k$ -means clustering algorithm, and its efficacy has been demonstrated. The proposed design achieves the best clustering result over the other approximate adders. Accordingly, the proposed adder design with the effective carry speculation and error reduction is suitable for error-resilient applications requiring high energy-efficiency, such as multimedia processing, data mining, and machine learning.

**Author Contributions:** Conceptualization, Y.K.; methodology, Y.K.; software, H.S. (Hyelin Seok); validation, H.S. (Hyelin Seok) and J.L.; formal analysis, H.S. (Hyoju Seo); investigation, H.S. (Hyelin Seok), H.S. (Hyoju Seo) and J.L.; resources, Y.K.; data curation, Y.K.; writing—original draft preparation, H.S. (Hyelin Seok), H.S. (Hyoju Seo) and J.L.; writing—review and editing, Y.K.; visualization, H.S. (Hyelin Seok); supervision, Y.K.; project administration, Y.K.; funding acquisition, Y.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Acknowledgments:** This work was supported in part by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2019R1I1A3A01061266) and in part by the BK21 FOUR project (AI-driven Convergence Software Education Research Program) funded by the Ministry of Education, School of Computer Science and Engineering, Kyungpook National University, Korea (4199990214394).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Alom, A.Z.; Taha, T.M.; Yakopcic, C.; Westberg, S.; Sidike, P.; Nasrin, M.S.; Hasan, M.; Van Essen, B.C.; Awwal, A.A.S.; Asari, V.K. State-of-the-Art Survey on Deep Learning Theory and Architectures. *Electronics* **2019**, *8*, 292. [[CrossRef](#)]
2. Ma, X.; Hu, S.; Liu, S.; Fang, J.; Xu, S. Remote Sensing Image Fusion Based on Sparse Representation and Guided Filtering. *Electronics* **2019**, *8*, 303. [[CrossRef](#)]
3. Wang, Q.; Li, P.; Kim, Y. A Parallel Digital VLSI Architecture for Integrated Support Vector Machine Training and Classification. *IEEE Trans. Very Large Scale. Integr. (VLSI) Syst.* **2015**, *23*, 1471–1484. [[CrossRef](#)]
4. Khan, I.; Choi, S.; Kwon, Y.-W. Earthquake Detection in a Static and Dynamic Environment Using Supervised Machine Learning and a Novel Feature Extraction Method. *Sensors* **2020**, *20*, 800. [[CrossRef](#)]
5. Lee, J.; Khan, I.; Choi, S.; Kwon, Y.-W. A Smart IoT Device for Detecting and Responding to Earthquakes. *Electronics* **2019**, *8*, 1546. [[CrossRef](#)]
6. Mittal, S. A Survey of Techniques for Approximate Computing. *ACM Comput. Survey* **2016**, *48*, 62:1–62:33. [[CrossRef](#)]
7. Pashaeifar, M.; Kamal, M.; Afzali-Kusha, A.; Pedram, M. Approximate Reverse Carry Propagation Adder for Energy-Efficient DSP Applications. *IEEE Trans. Very Large Scale. Integr. (VLSI) Syst.* **2018**, *26*, 2530–2541. [[CrossRef](#)]
8. Mahdiani, H.; Ahmadi, A.; Fakhraie, S.M.; Lucas, C. Bio-Inspired Imprecise Computational Blocks for Efficient VLSI Implementation of Soft-Computing Applications. *IEEE Trans. Circuits Syst. I Reg. Pap.* **2010**, *57*, 850–862. [[CrossRef](#)]
9. Zhu, N.; Goh, W.L.; Zhang, W.; Yeo, K.S.; Kong, Z.H. 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.* **2010**, *18*, 1225–1229.
10. Gupta, V.; Mohapatra, D.; Raghunathan, A.; Roy, K. Low-Power Digital Signal Processing Using Approximate Adders. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.* **2013**, *32*, 124–137. [[CrossRef](#)]
11. Dalloo, A.; Najafi, A.; Garcia-Ortiz, A. Systematic Design of an Approximate Adder: The Optimized Lower Part Constant-OR Adder. *IEEE Trans. Very Large Scale. Integr. (VLSI) Syst.* **2018**, *26*, 1595–1599. [[CrossRef](#)]
12. Seo, H.; Yang, Y. S.; Kim, Y. Design and Analysis of an Approximate Adder with Hybrid Error Reduction. *Electronics* **2020**, *9*, 471. [[CrossRef](#)]
13. Lee, J.; Seo, H.; Kim, Y.; Kim, Y. Approximate Adder Design with Simplified Lower-part Approximation. *IEICE Electron. Express* **2020**, *17*, 1–3. [[CrossRef](#)]

14. Balasubramanian, P.; Maskell, D.L. Hardware Optimized and Error Reduced Approximate Adder. *Electronics* **2019**, *8*, 1212. [[CrossRef](#)]
15. Balasubramanian, P.; Nayar, R.; Maskell, D.L.; Mastorakis, N.E. An Approximate Adder With a Near-Normal Error Distribution: Design, Error Analysis and Practical Application. *IEEE Access* **2020**, *9*, 4518–4530. [[CrossRef](#)]
16. Lee, J.; Seo, H.; Kim, Y.; Kim, Y. Design of a Low-Cost Approximate Adder with a Zero Truncation. In Proceedings of the International System-on-Chip (SOC) Design Conference, Yeosu, Korea, 21–24 October 2020; pp. 69–70.
17. Kim, Y.; Zhang, Y.; Li, P. An Energy Efficient Approximate Adder with Carry Skip for Error Resilient Neuromorphic VLSI Systems. In Proceedings of the IEEE/ACM International Conference on Computer-Aided Design, San Jose, CA, USA, 18–21 November 2013; pp. 130–137.
18. Kim, Y.; Zhang, Y.; Li, P. Energy Efficient Approximate Arithmetic for Error Resilient Neuromorphic Computing. *IEEE Trans. Very Large Scale. Integr. (VLSI) Syst.* **2015**, *23*, 2733–2737. [[CrossRef](#)]
19. Shafique, M.; Ahmad, W.; Hafiz, R.; Henkel, J. A Low Latency Generic Accuracy Configurable Adder. In Proceedings of the IEEE/ACM Design Automation Conference, San Francisco, CA, USA, 8–12 June 2015; pp. 81:1–81:6.
20. Camus, V.; Cacciotti, M.; Schlachter J.; Enz, C. Design of Approximate Circuits by Fabrication of False Timing Paths: The Carry Cut-Back Adder. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2018**, *8*, 4, 746–757. [[CrossRef](#)]
21. Ebrahimi-Azandaryani, F.; Akbari, O.; Kamal, M.; Afzali-Kusha, A.; Pedram, M. Block-Based Carry Speculative Approximate Adder for Energy-Efficient Applications. *IEEE Trans. Circuits Syst. II Exp. Briefs* **2020**, *67*, 137–141. [[CrossRef](#)]
22. Kim, Y. An Accuracy Enhanced Error Tolerant Adder with Carry Prediction for Approximate Computing. *IEIE Trans. Smart Process. Comput.* **2019**, *8*, 324–330. [[CrossRef](#)]
23. Kim, Y. A Novel Approximate Adder with Enhanced Low-cost Carry Prediction for Error Tolerant Computing. *IEIE Trans. Smart Process. Comput.* **2019**, *8*, 506–510. [[CrossRef](#)]
24. Akbari, O.; Kamal, M.; Afzali-Kusha, A.; Pedram, M. RAP-CLA: A Reconfigurable Approximate Carry Look-Ahead Adder. *IEEE Trans. Circuits Syst. II Exp. Briefs* **2018**, *65*, 1089–1093. [[CrossRef](#)]
25. Hu, J.; Li, Z.; Yang, M.; Huang, Z.; Qian, W. A High-Accuracy Approximate Adder with Correct Sign Calculation. *Integration* **2019**, *65*, 370–388. [[CrossRef](#)]
26. Bhatnagar, H. *Advanced ASIC Chip Synthesis: Using Synopsys Design Compiler Physical Compiler and Prime Time*, 2nd ed.; Kluwer Academic Publishers: Dordrecht, The Netherlands, 2002.
27. Raha, A.; Jayakumar, H.; Raghunathan, V. Input-Based Dynamic Reconfiguration of Approximate Arithmetic Units for Video Encoding. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2016**, *24*, 846–857. [[CrossRef](#)]
28. Soares, L.B.; da Rosa, M.M.A.; Diniz, C.M.; da Costa, E.A.C.; Bampi, S. Design Methodology to Explore Hybrid Approximate Adders for Energy-Efficient Image and Video Processing Accelerators. *IEEE Trans. Circuits Syst. I Reg. Pap.* **2019**, *66*, 2137–2150. [[CrossRef](#)]
29. Liang, J.; Han, J.; Lombardi, F. New Metric for the Reliability of Approximate and Probabilistic Adders. *IEEE Trans. Comput.* **2013**, *62*, 1760–1771. [[CrossRef](#)]
30. Clustering Benchmark. Available online: <http://github.com/deric/clustering-benchmark> (accessed on 25 July 2021).