Article

Design and Analysis of an Approximate Adder with Hybrid Error Reduction

Hyoju Seo ¹, Yoon Seok Yang ² and Yongtae Kim ^{1,*}

¹ School of Computer Science and Engineering, Kyungpook National University, Daegu 41566, Korea; hyoju@knu.ac.kr

² Intel Labs, Intel Corporation, Santa Clara, CA 95054, USA; yoonseok.yang@intel.com

* Correspondence: yongtae@knu.ac.kr; Tel.: +82-53-750-7574

Received: 1 February 2020; Accepted: 10 March 2020; Published: 11 March 2020

Abstract: This paper presents an energy-efficient approximate adder with a novel hybrid error reduction scheme to significantly improve the computation accuracy at the cost of extremely low additional power and area overheads. The proposed hybrid error reduction scheme utilizes only two input bits and adjusts the approximate outputs to reduce the error distance, which leads to an overall improvement in accuracy. The proposed design, when implemented in 65-nm CMOS technology, has 3, 2, and 2 times greater energy, power, and area efficiencies, respectively, than conventional accurate adders. In terms of the accuracy, the proposed hybrid error reduction scheme allows that the error rate of the proposed adder decreases to 50% whereas those of the lower-part OR adder and optimized lower-part OR constant adder reach 68% and 85%, respectively. Furthermore, the proposed adder has up to 2.24, 2.24, and 1.16 times better performance with respect to the mean error distance, normalized mean error distance (NMED), and mean relative error distance, respectively, than the other approximate adder considered in this paper. Importantly, because of an excellent design tradeoff among delay, power, energy, and accuracy, the proposed adder is found to be the most competitive approximate adder when jointly analyzed in terms of the hardware cost and computation accuracy. Specifically, our proposed adder achieves 51%, 49%, and 47% reductions of the power-, energy-, and error-delay-product-NMED products, respectively, compared to the other considered approximate adders.

Keywords: approximate adder; approximate computing; hybrid error reduction; low power; energy efficiency

1. Introduction

Energy efficiency has become a critical requirement in the design of modern computing systems and system-on-chips, and chip designers are being continually urged to develop energy-efficient design techniques to meet this requirement. Approximate computing can offer remarkable energy savings by trading-off accuracy [1]. This approach is based on the observation that not all applications require 100% computation accuracy. Specifically, many digital signal processing (DSP) applications are inherently error-resilient [2–4]. For example, humans may not recognize sporadic errors in digital image processing, such as lossy discrete cosine transform, since they are usually negligible because of human sensory limitations.

While approximate computing can be performed in all computing layers, ranging from software to circuit level [5,6], in this paper, we focus on approximate circuits, particularly an approximate adder. Certainly, an approximate adder is a fundamental arithmetic unit that is frequently used in many error-tolerant applications to decrease the overall energy consumption, and it has drawn much

attention from researchers. As a result, numerous approximate adders have been proposed in the literature [7–31], which will be reviewed in Section 2.

In this paper, we propose a novel approximate adder with a hybrid error reduction technique and systematically analyze and extensively compare our proposed adder with other approximate adders in terms of the hardware performance and computation accuracy. The proposed scheme improves the error rate (ER) of the lower-part OR adder (LOA) and error tolerant adder I (ETAI) from 68% to 50%. When the proposed adder is implemented in a 65-nm CMOS technology, its mean error distance (MED) and mean relative error distance (MRED) performances are also enhanced by more than 120% and more than 50%, respectively, compared to those of the ETAI at the cost of extremely low area and power overheads (<10%). Impressively, our adder shows an excellent tradeoff between the hardware cost and the computation accuracy and is clearly superior to the other considered approximate adders. Specifically, the power- and energy-normalized mean error distance (NMED) products of our approximate adder are up to 2.06 and 1.97 times, respectively, better than those of the other considered approximate adders.

The remainder of this paper is organized as follows. In Section 2, we introduce some of the related works that were recently published in the field of approximate adders. Then, we present the proposed approximate adder by providing illustrative examples in Section 3. The hardware architecture and error analysis of the adder are given in Section 3 as well. In Section 4, the results of hardware implementation together with the systematic analysis and extensive comparison with the other seven approximate adders in terms of the performance and accuracy are presented. In addition, the joint analysis of the approximate adders is provided in Section 4. Finally, we conclude the work in Section 5.

2. Related Works

Lu proposed an approximate adder wherein the carry for each sum bit is predicted by the limited number of its less significant bits to improve the overall speed [7]. Verma et al. reduced the area overheads of Lu's adder by sharing some components [8]. These adders are faster than the conventional accurate adders because of the shorter carry propagation chain.

An equal-segment-based approximate adder splits an n -bit adder into several equally partitioned smaller k -bit sub-adders that concurrently perform partial additions and partial carry generations using a limited number of input bits. It should be noted that sub-adders can be implemented in any type of conventional accurate adder, such as the ripple carry adder (RCA) and carry lookahead adder (CLA). The adders proposed in [9,10] utilize the previous k input bits only to predict the carry for sub-adders. Kim et al. [11,12] proposed an improved carry speculation scheme for sub-adders that leverages $2k$ and more input bits to generate carries, which leads to a much better computation accuracy than that of the adders in [9,10]. The accuracy-configurable adder proposed by Kahng et al. [13] includes multiple $2k$ -bit sub-adders and makes k bits overlap to generate approximate outputs.

In addition, approximate 1-bit full adders are utilized to add some least significant bits (LSBs) in a multibit adder. For example, the LOA, as shown in Figure 1, employs OR gates to approximately add several LSB inputs [14]. It divides an n -bit adder into a k -bit precise adder and an $(n-k)$ -bit approximate adder. An AND operation of two $(n-k-1)$ th LSB inputs is performed to predict the carry-in signal of the precise adder (i.e., C_{in}), and the precise adder takes the most significant k bit (MSB) inputs and the carry to generate accurate outputs for the MSBs. The $(n-k)$ -bit approximate adder is realized via bit-by-bit OR operations. The adder proposed by Albicoocco et al. [15], termed the LOAWA (i.e., LOA without AND operation) also uses the OR operation to realize the $(n-k)$ -bit approximate adder for the LSBs; however, the main difference with the LOA is the exclusion of the carry prediction scheme to reduce hardware cost at the expense of accuracy. The optimized lower-part constant OR adder (OLOCA) [16] is almost identical to the LOA. The approximate adder part utilizes the OR function but a few LSB outputs are forced to "1" to reduce hardware cost. In other words, only a few of the upper $(n-k)$ LSB outputs are generated by the OR operation, and the remaining lower bits of the LSB outputs are fixed to "1." The hardware optimized error reduced adder (HOERRA) proposed by Balasubramanian et al. [31]

is another optimized version of the LOA, which is suitable for both field programmable gate array and application specific integrated circuit-based implementations. The approximate adder part of the HOERAA is similar to that of the OLOCA in that the $(n-k-2)$ LSB outputs are set to “1.” The remaining two bits of the part leverages OR function, and a 2-to-1 multiplexer is used to produce $(n-k-1)$ th LSB output by selecting either an OR operation of two $(n-k-1)$ th LSB inputs or an AND operation of two $(n-k-2)$ th LSB inputs. The carry-in signal (i.e., C_{in}) serves as the selection input of the multiplexer.

Figure 1. Architecture of lower-part OR adder (LOA).

Similar to the LOA, the ETAI uses a k -bit precise adder and an $(n-k)$ -bit approximate adder, the latter of which is realized via its own modified XOR function [17] and is composed of a control block and a carry-free addition block, as shown in Figure 2. It is important to note that the ETAI does not include a carry prediction scheme for the precise adder and the carry is set to “0” (i.e., $C_{in} = 0$), which leads to poor overall computation accuracy compared to that of the LOA. To improve the accuracy, Kim developed the carry predicting ETA (CPETA) and the enhanced CPETA (ECPETA) in [18] and [19], respectively, by adding low-cost carry prediction techniques to the original ETAI architecture. Specifically, the CPETA uses the same AND-operation-based carry prediction scheme with the LOA, and the ECPETA utilizes both the $(n-k-1)$ th and $(n-k-2)$ th LSB inputs with an additional OR gate to predict the carry to improve the accuracy. The LOA, ETAI, and their variants show a good tradeoff between the computation accuracy and the power and area costs thanks to the simplification of the LSB additions.

Figure 2. Architecture of error tolerant adder I (ETAI).

In addition to design of the approximate adder, the evaluation of its accuracy is also an important task; some error metrics have been introduced for this purpose [32], such as the MED, MRED, and NMED. These metrics are widely adopted along with power, energy, delay, and area to evaluate and compare the performances of approximate adders [33].

3. Proposed Approximate Adder

In this section, we present our proposed approximate adder that enhances the computation accuracy of the LOA by application of a novel hybrid error reduction scheme, and the resultant adder is termed a hybrid error reduction LOA (HERLOA). We provide illustrative examples to effectively introduce the proposed adder and use the following notations. Let $A_{n-1:0}$, $B_{n-1:0}$, and $S_{n-1:0}$ denote, respectively, the two n -bit inputs and one n -bit output of the adder, and let $S'_{n-1:0}$ denote the n -bit intermediate approximate output prior to error reduction. Additionally, A_i , B_i , S_i , and S'_i denote the i th LSBs of $A_{n-1:0}$, $B_{n-1:0}$, $S_{n-1:0}$, and $S'_{n-1:0}$, respectively.

3.1. Operation of the Proposed Adder

Figure 3 shows the operation of the proposed adder with 16-bit inputs. An n -bit addition requires two different kinds of additions. The precise part is obtained using a k -bit accurate adder (e.g., RCA or CLA) for k MSBs, and the approximate part is obtained via OR and XOR operations of the remaining $(n-k)$ LSBs, where $k < n$. The example in Figure 3 splits a 16-bit input into an 8-bit precise part and an 8-bit approximate part. It is worth mentioning that the bit-widths of two parts do not need to be equal. The precise part performs an accurate addition using k MSB inputs, and the carry-in signal (C_{in}) is predicted via an AND operation of two $(n-k-1)$ th inputs (i.e., $C_{in} = A_{n-k-1} \text{ AND } B_{n-k-1}$). Similar to the LOA, the approximate part leverages the OR function to add the lower order input bits of two operands; however, the main difference with the LOA is that, in the proposed adder, the bit operation of $(n-k-1)$ th inputs is replaced with the XOR operation. This results in the formation of a half adder for $(n-k-1)$ th LSB inputs and effectively extends the length of accurate addition by one bit, which consequently leads to an overall improvement in accuracy compared to that of the LOA. In other words, the proposed n -bit adder always generates correct outputs for bit positions from $(n-1)$ to $(n-k-1)$ by replacing the OR gate with the XOR gate at the $(n-k-1)$ th LSB position. As a result, under the input shown in Figure 3, the proposed adder generates the approximate part output of “01011010”, whereas the LOA generates an output of “11011010.” Obviously, the output of the proposed adder is closer to the correct summation of “01101010” than the output of the LOA, and the error distance—which is defined as $|S_{approximate} - S_{accurate}|$, where $S_{approximate}$ and $S_{accurate}$ are the approximate and correct outputs, respectively—for the given input decreases from “1110000” (112) to “10000” (16).

Figure 3. Operation of proposed adder.

3.2. Proposed Hybrid Error Reduction Scheme

The proposed approximate adder performs an error reduction to further decrease the output errors when both $(n-k-2)$ th input bits are “1” (i.e., $A_{n-k-2} = 1$ and $B_{n-k-2} = 1$). Otherwise, it does not perform any error reduction, as shown in the example in Figure 3. In fact, our adder implements the error reduction logic, but this does not affect the final outputs of the adder in this case. The proposed hybrid error reduction is performed differently depending on the $(n-k-1)$ th inputs. In other words, the proposed adder checks the $(n-k-1)$ th output bit to determine which of the two error reduction schemes is applicable. If both the inputs are “1” or “0,” the reduction logic corrects the $(n-k-1)$ th

and $(n-k-2)$ th outputs to “1” and “0,” respectively (i.e., $S_{n-k-1:n-k-2} = “10”$). Otherwise, it sets all the outputs from S_{n-k-3} to S_0 as “1.”

The example inputs shown in Figure 4a yield the approximate part output of “01011010” after the normal addition described in Section 3.1; we term this output the intermediate approximate output. Then, the adder further checks the $(n-k-1)$ th output bit because both the $(n-k-2)$ th input bits are “1”. An intermediate approximate output S'_{n-k-1} of “0” implies that the corresponding input bits are identical because of the XOR operation performed at the $(n-k-1)$ bit position, and the final adder outputs at the $(n-k-1)$ to $(n-k-2)$ bit positions are “1” and “0”, respectively. In short, $S_{n-k-1:n-k-2} = “10”$, and this is, in fact, the correct summation of outputs at the corresponding positions under the given inputs. This scheme leads to a 2^{n-k-2} reduction in the error distance.

Figure 4. Operations of proposed error reduction when (a) A_{n-k-1} and B_{n-k-1} are identical and (b) A_{n-k-1} and B_{n-k-1} are exclusive.

On the other hand, if the intermediate approximate output S'_{n-k-1} is “1”, as illustrated in Figure 4b, all the remaining lower order bits are forced to “1,” which results in the approximate output of “1111111.” Under this given input condition, a carry is supposed to be generated in the $(n-k-2)$ th LSB and propagate to the precise part through the $(n-k-1)$ th LSB. However, as observed in Figure 4b, the carry does not actually propagate to the precise adder (i.e., $C_{in} = 0$) because the carry prediction is performed using an AND operation with only the $(n-k-1)$ th LSB inputs. This result means that the correct summation will always be larger than the proposed addition in this case. Therefore, forcing

all the outputs of the approximate part to “1” brings the approximate output closer to the correct summation. This reduction scheme enables up to a $2^{n-k-2} - 1$ decrease in the error distance.

3.3. Implementation of the Proposed Adder

Figure 5 shows the hardware implementation of the approximate part of the proposed HERLOA. It should be noted that the precise part is the same as that in Figure 1, and C_{in} is fed to the precise adder. In the approximate addition operation, the XOR and OR gates are used for the $(n-k-1)$ th LSB and the other lower order LSBs, respectively, to generate the intermediate approximate outputs $S'_{n-k-1:0}$. Furthermore, the output of the AND operation of the $(n-k-2)$ th LSB determines whether or not error reduction should be performed. The intermediate approximate outputs $S'_{n-k-1:0}$ are fed to the INV, AND, OR, and NAND gates to compute the final approximate outputs $S_{n-k-1:0}$. It is important to note that the intermediate approximate outputs will bypass the reduction logic when neither of the $(n-k-2)$ th inputs is “1.” The critical path delay of the approximate part, $t_{approximate}$, can be expressed simply as follows:

$$t_{approximate} = t_{XOR} + t_{INV} + t_{NAND} + t_{AND}, \quad (1)$$

where t_{INV} , t_{XOR} , t_{NAND} , and t_{AND} are the delays of an inverter, a two-input XOR gate, a NAND gate, and an AND gate, respectively.

Figure 5. Architecture of approximate part of proposed adder, i.e., hybrid error reduction LOA (HERLOA).

3.4. Error Rate Analysis

An output error occurs when any bit positions from $(n-k-3)$ to 0 of both the inputs, A and B, are “1”, which generates a carry for the higher bit position. Moreover, an error occurs when both the $(n-k-2)$ th LSB inputs are “1” and the two $(n-k-1)$ th LSB inputs are exclusive, as observed in Figure 4b. Therefore, the ER of the proposed HERLOA with the error reduction scheme under random input patterns is given as follows:

$$ER_{HERLOA}(n, k) = 1 - \frac{7}{8} \left(\frac{3}{4} \right)^{n-k-2}, \quad (2)$$

where n and k are the sizes of the entire adder and the precise adder, respectively.

4. Experimental Results

The proposed approximate adder with design parameters of $n = 16$ and $k = 8$ was designed in Verilog HDL and was synthesized using 65-nm CMOS technology and a standard cell library to evaluate the delay, area, power, power-delay product (PDP), and energy-delay product (EDP). The 8-bit RCA was employed as the precise adder. For comparison of our proposed adder with other

adders, two conventional accurate adders (RCA and CLA) as well as seven approximate adders (LOA, ETAI, and their variants: LOAWA, OLOCA, HOERAA, CPETA, and ECPETA) were designed and also synthesized using the same technology and library. For fair comparison, identical design parameters, i.e., $n = 16$ and $k = 8$, and the RCA structure were used for all the approximate adders. Furthermore, the bit-width of the constant part of the OLOCA and HOERAA were selected to be 6 [16,31].

In addition to implementing hardware, we constructed a software simulator to assess the accuracy performance of the approximate adders in terms of the ER, MED, NMED, and MRED. These error metrics are expressed as follows:

$$MED = \frac{1}{n} \sum_{i=1}^n ED_i, \quad (3)$$

$$MRED = \frac{1}{n} \sum_{i=1}^n \left| \frac{ED_i}{S_{i, accurate}} \right|, \quad (4)$$

$$NMED = \frac{MED}{D} = \frac{1}{n} \sum_{i=1}^n \frac{ED_i}{D}, \quad (5)$$

where n is the number of inputs, ED_i is the error distance for the (i)th item of input data, $S_{i, accurate}$ is the accurate output for the (i)th item of input data, and D is the maximum possible error value of the approximate adder. These error metrics were estimated by using two samples each comprising 10 million (i.e., 10^7) uniformly distributed random input numbers.

4.1. Performance Analysis

Table 1 summarizes the performance of the proposed approximate adder and those of the other eight adders. While the CLA is the fastest, the RCA has the longest delay because of the bit-by-bit carry propagation. This delay of the RCA causes it to consume the largest amount of energy (i.e., the highest PDP) even though its power dissipation is lower than that of the CLA, which consumes the second largest amount of energy. The LOA and its variants (i.e., LOAWA, OLOCA, and HOERAA) are more area-, power-, energy-, and EDP-efficient than the ETAI and its variants (i.e., CPETA and ECPETA) because of the relatively simpler approximation scheme (i.e., OR operation) for the lower half of the input bits in the case of the former category. Specifically, the OLOCA occupies the smallest area, and the LOAWA is the fastest as well as the most power- and energy-efficient among all the approximate adders. The simple AND-operation-based carry prediction in the LOA, OLOCA, HOERAA, CPETA, and our proposed adder results in a longer delay than those of the ETAI and LOAWA, which lack any carry prediction for the precise part. It should be noted that the critical path delay of these adders exists in the precise adder part, including the carry prediction (i.e., the 8-bit RCA with the AND gate). The delay of the HOERAA is insignificantly longer than that of the LOA, OLOCA, CPETA, and the proposed adder even though all these adders have the identical carry prediction scheme. It is because the carry prediction output (i.e., the AND gate output) of the HOERAA is fed into not only the precise adder but also the multiplexer. This causes a higher fan-out of the AND gate and therefore impacts on the delay. The more complicated carry prediction scheme adopted in the ECPETA, which utilizes two MSB inputs from the approximate part, leads to the worst area, delay, and power performances; as a result, this adder consumes the largest amount of energy and has the highest EDP among the seven approximate adders. The proposed HERLOA is comparable to the ETAI in terms of all the hardware performance metrics. It occupies 8% larger area and consumes 9% more power than the LOA while having the same speed. A comparison of our adder with the accurate adders reveals that our adder has up to 2.18, 1.96, 2.19, and 3.31 times greater efficiency in terms of area, delay, power, and energy, respectively.

Table 1. Summary of performances of various 16-bit adders with $n = 16$ and $k = 8$.

Adder	Area (μm^2)	Delay (ns)	Power (μW)	PDP (fJ)	EDP (fJ-s)	ER (%)	MED	MRED	NMED
RCA	157.8	2.23	45.0	100.4	2.24×10^{-7}	N/A	N/A	N/A	N/A
CLA	227.9	1.02	58.2	59.4	6.06×10^{-8}	N/A	N/A	N/A	N/A
LOA	97.0	1.14	24.4	27.8	3.17×10^{-8}	89.98	110.9	4.38	1.69×10^{-3}
LOAWA	88.6	1.09	22.0	24.0	2.62×10^{-8}	89.98	189.6	5.08	2.89×10^{-3}
OLOCA	85.4	1.14	22.9	26.1	2.98×10^{-8}	99.13	115.0	4.38	1.75×10^{-3}
HOERAA	89.3	1.16	23.8	27.6	3.20×10^{-8}	98.84	95.01	4.38	1.45×10^{-3}
ETAI	108.5	1.09	26.3	28.7	3.13×10^{-8}	89.98	177.1	5.08	2.70×10^{-3}
CPETA	113.9	1.14	28.0	31.9	3.64×10^{-8}	86.65	88.5	4.38	1.35×10^{-3}
ECPETA	116.5	1.26	29.7	37.4	4.71×10^{-8}	86.65	81.5	3.70	1.24×10^{-3}
HERLOA	104.6	1.14	26.6	30.3	3.45×10^{-8}	84.43	84.8	4.38	1.29×10^{-3}

Our adder shows the best ER performance, whereas the OLOCA shows the worst ER performance: it reaches over 99%, and the HOERAA has almost the same ER with the OLOCA. The ERs of the LOA, LOAWA, and ETAI are the same and the ER of the CPETA is identical to that of the ECPETA. The ETAI variants show better accuracy performance than the LOA and its variants in terms of all the error metrics, i.e., the ER, MED, MRED, and NMED, but consumes more area, power, and energy. The HOERAA shows the best accuracy performance in the metrics except the ER among the LOA and its variants. The MED and NMED of the proposed adder are comparable to those of the ECPETA, and the MRED of the proposed adder is the same as that of the LOA, OLOCA, HOERAA, and CPETA. Importantly, our proposed adder outperforms the other approximate adders in terms of all the accuracy metrics, the exceptions being its higher MED, MRED, and NMED than those of the ECPETA. Specifically, the proposed HERLOA shows 1.31, 2.09, and 2.24 times better MED than the LOA, ETAI, and LOAWA, respectively, and 1.16 times better MRED than the LOAWA and ETAI.

4.2. Accuracy Analysis

For evaluating the accuracy of the proposed HERLOA in comparison with those of the other seven approximate adders, we varied the design parameter k from 6 to 12 in order to alter the bit-width of the precise part of the 16-bit adders and to extract the value of the ER, MED, MRED, and NMED metrics.

Figure 6 shows the ERs of the approximate adders at various values of k . Clearly, the ER decreases as the precise adder size k increases. Regardless of k , the LOA, LOAWA, and ETAI have an identical ER, as do the CPETA and ECPETA, where the later ER is lower than that of the LOA. The proposed HERLOA has the lowest ER and the OLOCA has the highest ER. The HOERAA shows slightly better ER performance than the OLOCA and has the second highest ER. Specifically, the ERs of the OLOCA and HOERAA reach 85% and 81%, respectively, and that of our HERLOA decreases to 50% thanks to the proposed hybrid error reduction scheme at $k = 12$. The LOA and CPETA have ERs of 68% and 57%, respectively, at the same k . We also plotted the line of Equation (2) in Figure 6 to determine the accuracy of the ER of our HERLOA as derived by this equation. The line is in very good agreement with the simulated ERs at the various values of k .

Figure 6. Comparison of error rates of 16-bit approximate adders at various values of design parameter k .

As reported in Table 1, the proposed adder has the second-best MED performance at $k = 8$. To effectively demonstrate the MED of the proposed HERLOA and compare it with those of the other adders, we plotted the improvements in the MED of the proposed adder in comparison with those of the other seven approximate adders at various values of k in Figure 7. The MED improvement against all the adders except the ECPETA increases with increasing k . Although the OLOCA has the highest ER, its MED is similar to that of the LOA. Similarly, the HOERAA shows better MED performance than the LOA, LOAWA, and ETAI in spite of worse ER performance than those adders. Moreover, in terms of the MED, the HOERAA outperforms the OLOCA. The MED of our design is 30–43% better than those of the LOA and OLOCA. Furthermore, at all k values, the MED improvement of our adder reaches over 100% compared with the LOAWA and ETAI. Unfortunately, the proposed design shows 3–5% less MED performance against the ECPETA. The MED of the proposed design is comparable to those the ETA variants, which include their own carry prediction schemes; the MED difference in this case is $\pm 5\%$ at all k values.

Figure 7. Improvements in the mean error distance (MED) of the proposed HERLOA in comparison with those of seven approximate adders at various values of k .

Table 2 lists the MREDs of the seven approximate adders, including our proposed adder. For all the adders, the MRED decreases as k increases. The MREDs of the LOA, OLOCA, HOERAA, CPETA, and the proposed HERLOA are almost the same, whereas those of the LOAWA and ETAI are relatively higher and that of the ECPETA is lower. Interestingly, the MREDs of the adders are

almost identical when they have the same carry prediction scheme. For example, the LOA, OLOCA, HOERAA, and HERLOA include the AND-operation-based carry prediction scheme and have the same MRED, and the LOAWA and ETAI do not include any carry prediction scheme (i.e., $C_{in} = 0$) and have identical MREDs. The ECPETA includes the most accurate carry prediction scheme and shows the best MRED performance among all the adders considered in this paper. The MRED difference between the proposed adder and the ETAI/LOAWA remains unchanged as k increases. Specifically, this MRED difference is approximately 0.7 over the entire considered range of k values. Since the MRED decreases as k increases, the percentage of MRED improvement increases as k increases. For example, our design achieves MRED reductions of 12.3%, 16.0%, 25.3%, and 51.0%, at k values of 6, 8, 10, and 12, respectively.

Table 2. Mean relative error distances (MREDs) of 16-bit approximate adders at various values of design parameter k .

k	LOA	LOAWA	OLOCA	HOERAA	ETAI	CPETA	ECPETA	HERLOA
6	5.77	6.47	5.77	5.76	6.47	5.76	5.06	5.76
7	5.12	5.81	5.12	5.12	5.81	5.12	4.42	5.12
8	4.38	5.08	4.38	4.38	5.08	4.38	3.70	4.38
9	3.81	4.53	3.81	3.81	4.53	3.81	3.05	3.81
10	2.89	3.62	2.89	2.89	3.62	2.89	2.06	2.89
11	2.05	2.78	2.05	2.05	2.78	2.05	1.33	2.05
12	1.41	2.13	1.41	1.41	2.13	1.41	0.62	1.41

4.3. Joint Analysis of Performance and Accuracy

Power-NMED and energy-NMED products were introduced in [32] and [19], respectively, to assess the tradeoff among the power, energy, and accuracy of approximate adders. Here, we can take into account a new metric, the EDP-NMED product, to jointly analyze the tradeoff among energy, delay, and accuracy.

Figure 8 shows the power-NMED, energy-NMED, and EDP-NMED products of the seven approximate adders with $n = 16$ and $k = 8$. All three products are normalized using corresponding values of the LOA to effectively demonstrate the tradeoffs. Impressively, the proposed HERLOA shows the best tradeoff performance, whereas the ETAI has the largest values of all three products. Specifically, the power-, energy-, and EDP-NMED products of the ETAI are 72%, 65%, and 58%, respectively, larger than those of the LOA. In contrast, all three products of the proposed design are 16% smaller than those of the LOA. The HOERAA, which has slightly less energy- and EDP-NMED products than the proposed adder, also shows a good tradeoff and is comparable to the proposed HERLOA. Although the ECPETA demonstrates the best NMED performance, the additional delay and power consumption of this adder that originate from its carry prediction scheme prevent it from having the best tradeoff metrics. As an example, the EDP-NMED product of the ECPETA is 30% larger than that of the proposed design. Specifically, the power-, energy-, and EDP-NMED products of the proposed approximate adder are, respectively, 2.06, 1.97, and 1.88 times better than those of the ETAI. Clearly, the excellent tradeoff between hardware cost and computation accuracy makes our adder design the most competitive among all the considered approximate adders, which have similar hardware architectures.

Figure 8. Normalized power-normalized mean error distance (NMED), energy-NMED, and energy-delay product-NMED (EDP-NMED) products of approximate adders with $n = 16$ and $k = 8$.

5. Conclusions

In this paper, we have developed an accuracy enhanced lower-part OR adder with a hybrid error reduction scheme (termed the HERLOA) to significantly reduce the computation error while maintaining power and energy efficiencies. The proposed HERLOA replaces the OR gate with the XOR gate in the MSB of the approximate part and leverages two MSB inputs of the approximate part to decrease the approximation errors at the cost of a few digital gates. The proposed design is implemented in 65-nm technology to evaluate its performance; it is found to be 3, 2, and 2 times more energy-efficient, area-efficient, and power-efficient, respectively, than the RCA and CLA. In terms of accuracy, the proposed HERLOA outperforms the original LOA and the ETAI. Specifically, at a given design parameter value of $k = 12$, the ER of our adder decreases to 50%, whereas those of the other adders reach 68%, and at all the considered values of k , the MED improvement of our adder is more than 100% compared with the ETAI and LOAWA. Most importantly, the proposed adder shows an excellent design tradeoff between hardware cost and computation accuracy, as a result of which its power-NMED, energy-NMED, and EDP-NMED products are 2.06, 1.97, and 1.88 times better, respectively, than those of the ETAI. To sum up, our proposed adder outperforms all the approximate adders in a joint analysis of power, energy, EDP, and computation accuracy.

Consequently, the proposed approximate adder with the novel hybrid error reduction scheme is found to be highly power- and energy-efficient while also having good computation accuracy. Therefore, our design is highly suitable for application to inherently error-resilient energy-efficient computing, such as DSP, deep learning, and neuromorphic computing [2–4,34,35].

Author Contributions: Conceptualization, Y.K. and Y.S.Y.; methodology, H.S.; software, H.S. and Y.K.; validation, H.S.; formal analysis, Y.K.; investigation, Y.S.Y.; writing—original draft preparation, Y.K.; writing—review and editing, H.S. and Y.S.Y.; visualization, H.S.; supervision, Y.K.; funding acquisition, Y.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2019R111A3A01061266) and in part by the BK21 Plus Project (SW Human Resource Development Program for Supporting Smart Life) funded by the Ministry of Education, School of Computer Science and Engineering, Kyungpook National University, Korea (21A20131600005).

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

References

1. Moreau, T.; Sampson, A.; Ceze, L. Approximate Computing: Making Mobile Systems More Efficient. *IEEE Pervasive Comput.* **2015**, *14*, 9–13. [[CrossRef](#)]

2. 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](#)]
3. Jang, J.; Kim, H.; Shin, M.; Park, J.; Kim, J.; Paik, J. Vision-Based Railway Inspection System using Multiple Object Detection and Image Registration. *IEIE Trans. Smart Process. Comput.* **2018**, *7*, 440–447. [[CrossRef](#)]
4. 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](#)]
5. Mittal, S. A Survey of Techniques for Approximate Computing. *ACM Comput. Survey* **2016**, *48*, 62:1–62:33. [[CrossRef](#)]
6. Xu, Q.; Mytkowicz, T.; Kim, N.S. Approximate Computing: A Survey. *IEEE Des. Test* **2016**, *33*, 8–22. [[CrossRef](#)]
7. Lu, S.-L. Speeding Up Processing with Approximation Circuits. *Computer* **2004**, *37*, 67–73.
8. Verma, A.K.; Brisk, P.; Ienne, P. Variable Latency Speculative Addition: A New Paradigm for Arithmetic Circuit Design. In Proceedings of the Design, Automation and Test in Europe, Munich, Germany, 10–14 March 2008; pp. 1250–1255.
9. Zhu, Z.; Goh, W.L.; Yeo, K.S. An Enhanced Low-Power High-Speed Adder for Error-Tolerant Application. In Proceedings of the International Symposium on Integrated Circuits (ISIC), Suntec City, Singapore, 14–16 December 2009; pp. 69–72.
10. Du, K.; Verman, P.; Mohanram, K. High Performance Reliable Variable Latency Carry Select Addition. In Proceedings of the Design, Automation and Test in Europe, Dresden, Germany, 12–16 March 2012; pp. 2157–2162.
11. 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 (ICCAD), San Jose, CA, USA, 18–21 November 2013; pp. 130–137.
12. 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](#)]
13. Kahng, A.B.; Kang, S. Accuracy-Configurable Adder for Approximate Arithmetic Designs. In Proceedings of the IEEE/ACM Design Automation Conference, San Francisco, CA, USA, 3–7 June 2012; pp. 820–825.
14. 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. Papers* **2010**, *57*, 850–862. [[CrossRef](#)]
15. Albicocco, P.; Cardarilli, G.C.; Nannarelli, A.; Petricca, M.; Re, M. Imprecise Arithmetic for Low Power Image Processing. In Proceedings of the Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 4–7 November 2012; pp. 983–987.
16. 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](#)]
17. 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.
18. Kim, Y. An Accuracy Enhanced Error Tolerant Adder with Carry Prediction for Approximate Computing. *IEIE Trans. Smart Process. Comput.* **2019**, *8*, 324–330. [[CrossRef](#)]
19. 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](#)]
20. Liu, W.; Chen, L.; Wang, C.; O’Neill, M.; Lombardi, F. Design and Analysis of Inexact Floating-Point Adders. *IEEE Trans. Comput.* **2016**, *65*, 308–314. [[CrossRef](#)]
21. Miao, J.; He, K.; Gerstlauer, A.; Orshansky, M.E. Modeling and Synthesis of Quality-Energy Optimal Approximate Adders. In Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD), San Jose, CA, USA, 5–8 November 2012; pp. 728–735.
22. Kim, S.; Kim, Y. Novel XNOR-based Approximate Computing for Energy-Efficient Image Processors. *IEIE J. Semicond. Technol. Sci.* **2018**, *18*, 602–608. [[CrossRef](#)]
23. 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, 7–11 June 2015; pp. 81:1–81:6.

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. 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](#)]
26. Hu, J.; Qian, W. A New Approximate Adder with Low Relative Error and Correct Sign Calculation. In Proceedings of the Design, Automation and Test in Europe, Grenoble, France, 9–13 March 2015; pp. 1449–1454.
27. Camus, V.; Schlachter, J.; Enz, C. A Low-Power Carry Cut-Back Approximate Adder with Fixed-Point Implementation and Floating-Point Precision. In Proceedings of the IEEE/ACM Design Automation Conference, Austin, TX, USA, 5–9 June 2016; pp. 127:1–127:6.
28. Huang, N.-C.; Chen, S.-Y.; Wu, K.-C. Sensor-Based Approximate Adder Design for Accelerating Error-Tolerant and Deep-Learning Applications. In Proceedings of the Design, Automation and Test in Europe, Florence, Italy, 25–29 March 2019; pp. 692–697.
29. 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](#)]
30. Ye, R.; Wang, T.; Yuan, F.; Kumar, R.; Xu, Q. On Reconfiguration-Oriented Approximate Adder Design and Its Application. In Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD), San Jose, CA, USA, 18–21 November 2013; pp. 48–54.
31. Balasubramanian, P.; Maskell, D.L. Hardware Optimized and Error Reduced Approximate Adder. *Electronics* **2019**, *8*, 1212. [[CrossRef](#)]
32. Liang, J.; Han, J.; Lombardi, F. New Metric for the Reliability of Approximate and Probabilistic Adders. *IEEE Trans. Comput.* **2013**, *62*, 1760–1771. [[CrossRef](#)]
33. Jiang, H.; Liu, C.; Liu, L.; Lombardi, M.; Han, J. A Review, Classification, and Comparative Evaluation of Approximate Arithmetic Circuits. *ACM J. Emerg. Technol. Comput. Syst.* **2017**, *13*, 60:1–60:34. [[CrossRef](#)]
34. 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](#)]
35. Kim, Y.; Zhang, Y.; Li, P. A Reconfigurable Digital Neuromorphic Processor with Memristive Synaptic Crossbar for Cognitive Computing. *ACM J. Emerg. Technol. Comput. Syst.* **2015**, *11*, 38:1–38:25. [[CrossRef](#)]

© 2020 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 (<http://creativecommons.org/licenses/by/4.0/>).