LETTER

DCPA: approximate adder design exploiting dual carry prediction

Woong Choi^{1†}, Minseob Shim^{2†}, Hyelin Seok³, and Yongtae Kim^{3a)}

Abstract This letter presents a novel approximate adder that significantly improves computation accuracy by utilizing a dual carry prediction and error reduction scheme. In our experiments, the proposed adder improves mean error distance (MED) and mean relative error distance (MRED) by up to 58.6% and 58.5%, respectively, when compared with existing approximate adders. Also, when implemented in 65-nm CMOS technology, the proposed adder reduces area, delay, and power by 37%, 48%, and 41%, respectively, compared with the traditional adder. Furthermore, the effectiveness of our design over existing adders is investigated using a digital image processing application.

key words: approximate adder, dual carry prediction, energy efficiency

Classification: Integrated circuits (memory, logic, analog, RF, sensor)

1. Introduction

Recently, approximate computing, which trades computation accuracy for energy efficiency, has been investigated for use in error-resilient applications, such as image processing and machine learning [1, 2, 3, 4, 5]. The opportunity for efficiency lies in the fact that these applications do not require full precision accuracy but can tolerate a certain amount of errors in computations. One of the most important and heavily used arithmetic operations is addition and approximate adders improve energy efficiency by reducing hardware complexity in approximate computing [6, 7, 8]. Therefore, an efficient approximate adder is the key enabler and a considerable number of related works have been presented in the literature [9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24]. One of the most efficient approximate adder architecture is to split an adder into two parts: accurate and inaccurate parts [9, 13]. In general, the former yields accurate outputs for higher-order bits using a precise adder, such as ripple carry adder (RCA), whereas the latter yields approximate outputs for lower-order counterparts by utilizing approximate logics. Additionally, to improve overall computation accuracy, the

accurate part often uses a carry input predicted by the inaccurate part [9]. This architecture was adopted by the lower-part OR adder (LOA), and the OR operation is used to approximately add lower-order bits with the AND-based carry prediction [9]. The optimized lower-part constant OR adder (OLOCA) is identical to the LOA except that a few least significant bit (LSB) outputs are set to 1 regardless of the corresponding inputs, resulting in hardware cost savings with a slight degradation in accuracy [10]. The hardware optimized adder having a near-normal error distribution (HOANED) is a design variant of the OLOCA that improves error characteristics [11]. The approximate mirror adder 5 (AMA5) utilizes a mirror operation for both approximate addition and carry prediction [12]. The error tolerant adder I (ETAI) uses a modified XOR operation without a carry prediction, whereas the simplified ETA (SETA) optimized the ETAI's operation to decrease hardware complexity [13, 14].

In this letter, we present an approximate adder that uses a dual carry speculation, which utilizes two input pairs to predict the carry, to significantly improve the accuracy. Also, the proposed error reduction scheme improves error characteristics using only a few logic gates. To demonstrate the efficacy of our design, we systematically examine it and extensively compare it to existing adders in terms of hardware resource, computation accuracy, and practical application.

2. Proposed approximate adder design

Fig. 1 shows the proposed adder architecture, termed dual carry prediction adder (DCPA). The n -bit adder divides a k -bit accurate part to yield higher-order accurate outputs $S_{n-1:n-k}$ and carry output C_{out} using a precise adder and an $(n-k)$ -bit inaccurate part to obtain lower-order approximate ones $S_{n-k-1:0}$, which is done by the OR operation except for its most significant bit (MSB), where the XOR is used to improve the accuracy by forming a half adder. Also, the constant truncation for l LSBs is employed to reduce hardware complexity. While the conventional LOA and its variants use only $(n-k-1)^{th}$ input to predict a carry for the accurate part, in our design, its previous $(n-k-2)^{th}$ input is also utilized to enhance the prediction accuracy. The carry C_{in1} generated from $(n-k-1)^{th}$ input is directly fed into the precise adder. The other carry C_{n-k-2} generated from $(n-k-2)^{th}$ input should be propagated through $(n-k-1)^{th}$ bit position to pass it to the accurate part. Since an XOR plays a role of the propagate signal, there-

[†]These authors contributed equally to this work

¹Department of Electronics Engineering, Sookmyung Women's University, Seoul 04310, Korea

²Korea Electrotechnology Research Institute (KERI), Changwon, Gyeongnam 51543, Korea

³School of Computer Science and Engineering, Kyungpook National University, Daegu 41566, Korea

a) yongtae@knu.ac.kr

Fig. 1. Proposed approximate adder architecture.

 Fig. 2. An example operation of the proposed adder when $C_{in2} = 1$.

fore, C_{n-k-2} and $(n-k-1)^{th}$ XOR output are ANDed to produce the second carry C_{in2} (see solid-filled gates). Importantly, to avoid an increase in the adder's delay, this carry is ORed with the precise adder's LSB output to produce the final LSB (*i.e.*, $S_{n-k} = C_{in2} \text{ OR } S'_{n-k}$). Moreover, the error reduction logic (see pattern-filled gates) decreases the error distance (ED) metrics when the second carry C_{in2} is 1.

Fig. 2 illustrates an operation of our 16-bit adder with $k = 8$ and $l = 4$ when $C_{in2} = 1$. The accurate part adds k MSB inputs and the carry C_{in1} to produce an intermediate output $S'_{n-1:n-k}$. The inaccurate part produces a lower-order intermediate output $S'_{n-k-1:0}$ by conducting XOR/OR operations for the higher four bits with constant truncation for the lower four. Then, the second carry C_{in2} is ORed with the precise adder's LSB output to produce the final LSB output S_{n-k} . When $C_{in2} = 1$, the intermediate output $S'_{n-k-1:n-k-2}$ is always 11_2 but its correct output is 00_2 . Therefore, the error reduction logic decreases the ED by forcing the output to 00_2 . Note that this reduction occurs when $S'_{n-k} = 0$ because C_{in2} does not affect the accurate part's output when $S'_{n-k} = 1$. Also, this logic does not affect the output when $C_{in2} = 0$ and therefore, the intermediate output becomes the final. Under a given input in Fig. 2, the ED and relative ED (RED) are significantly reduced from 62 to 2 and from $1.14e-3$ to $3.67e-5$, respectively, by more than 96%.

3. Experimental results

The proposed adder, an accurate adder (RCA), and six existing approximate adders (LOA, OLOCA, HOAANED, AMA5, ETAI, and SETA) were designed in Verilog-HDL and synthesized with a 65-nm CMOS technology using Syn-

 Fig. 3. Performance analysis of the proposed adder according to l values.

Table I. Performance summary of various adders.

Design	Area (μm^2)	Delay (ns)	Power (μW)	MED	MRED ($1e-3$)
RCA	162.6	2.27	46.2	-	-
LOA	101.8	1.18	25.5	47.9	1.01
OLOCA	90.2	1.18	24.1	52.0	1.09
HOAANED	93.1	1.18	24.9	32.0	0.68
AMA5	94.7	1.17	25.0	64.0	1.35
ETAI	113.3	1.12	27.2	51.2	1.07
SETA	97.6	1.12	24.4	55.8	1.17
DCPA	102.1	1.18	27.3	26.5	0.56

opsys Design Compiler. We considered 16-bit adders with an 8-bit precise adder (*i.e.*, $n = 16$ and $k = 8$) [10, 25, 26, 27]. Also, the error metrics, such as mean ED (MED) and mean RED (MRED), were extracted under 10^7 random inputs.

Fig. 3 shows the performance analysis of the proposed adder with various l from 0 to 6. All the values were normalized using the value at $l = 0$. The area and power decrease as l increases because a larger l require fewer OR gates. Note that the delay is not affected by l and is consistent. The adder's MED and MRED trend is the same over l . They linearly decrease as l increases from 0 to 4 and drastically increase from $l = 5$ to $l = 6$. The accuracy performance is almost identical at $l = 4$ and $l = 5$ but a larger l brings a greater hardware benefit. Therefore, $l = 5$ can be chosen for the best tradeoff between hardware and accuracy and the proposed adder with $l = 5$ will be used for comparison with others.

Table I summarizes the performance of various adders. The RCA has the longest delay due to its long carry chain and consumes the most area and power. Because of its simple approximation with constant truncation, the area and power of the OLOCA are the smallest. The lack of carry prediction makes the ETAI and SETA the fastest adder. The HOAANED and AMA5 use a similar hardware resource, as do the LOA and our DCPA. The MED and MRED values for the OLOCA, ETAI, and SETA are similar, but the LOA's accuracy shows slightly better than these adders. Certainly, our adder design achieves the best in terms of the MED and MRED performance. Specifically, the proposed DCPA outperforms the AMA5 by $2.42\times$ and $2.41\times$, respectively.

To assess the efficacy of the proposed adder in practical applications, we applied our design to a 3×3 edge detection filter in [28]. We filtered the well-known 8-bit grayscale *cameraman* image using our adder with $n = 12$, $k = 6$, and $l = 3$ to avoid biased inputs. The same parameters were used for others. The output images filtered using accurate and approximate adders are shown in Fig. 4. The peak signal-to-

Fig. 4. Output images with PSNRs filtered using various adders.

noise ratio (PSNR) was used to assess the image quality, and the values were extracted against the output image filtered using the accurate adder [29, 30]. The OLOCA and SETA produce nearly identical output images that are visually indistinguishable with a PSNR difference ≤ 0.1 dB from the LOA and ETAI, respectively, and thus, the images filtered using them were excluded from this analysis. The filtering using the LOA and AMA5 do not properly detect edges and pixel inversions occurred in some regions. The PSNRs of the HOAANED and ETAI are higher, at 16.7 dB and 15.1 dB, respectively, but there is some noise and unclear edges. The proposed DCPA achieves the best performance with a PSNR of 19.1 dB in the edge detection because the resultant image is the closest to the one filtered using the accurate adder.

4. Conclusion

This letter proposed an approximate adder that utilizes a dual carry prediction to significantly improve the overall accuracy with the error reduction scheme. The proposed adder, when implemented in 65-nm CMOS technology, reduced area, delay, and power by 37%, 48%, and 41%, respectively, and achieved up to a 2.4 $\times$ better MED and MRED performance than the AMA5. Furthermore, the efficacy of the proposed design was clearly demonstrated in the edge detection, where our adder outperforms existing approximate adders.

Acknowledgments

This work was supported by Basic Science Research Program through National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2019R1I1A3A01061266) and Institute of Information & Communications Technology Planning & Evaluation (IITP) funded by the Korea government (MSIT) (2021-0-00903).

References

- [1] J. Chang, *et al.*: "An efficient implementation of 2D convolution in CNN," *IEICE Electron. Express* **14** (2017) 1.
- [2] Q. Wang, *et al.*: "A parallel digital VLSI architecture for integrated support vector machine training and classification," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **23** (2015) 1471.
- [3] Y.-L. Shen, *et al.*: "Pipelined implementation of AI-based Loeffler DCT," *IEICE Electron. Express* **10** (2013) 1.
- [4] Y.-L. Shen, *et al.*: "Loeffler DCT accelerator for small portable devices," *IEICE Electron. Express* **12** (2015) 1.
- [5] Y. Kim, *et al.*: "A reconfigurable digital neuromorphic processor with memristive synaptic crossbar for cognitive computing," *ACM J. Emerg. Technol. Comput. Syst.* **11** (2015) 38.
- [6] D. Liu *et al.*: "Delay-optimized floating point fused add-subtract unit," *IEICE Electron. Express* **12** (2015) 1.
- [7] B. K. Mohanty: "Efficient fixed-width adder-tree design," *IEEE Trans. Circuits Syst. II, Exp. Briefs* **66** (2018) 292.
- [8] S. Mittal: "A survey of techniques for approximate computing," *ACM Comput. Survey* **48** (2016) 62.
- [9] H. Mahdiani, *et al.*: "Bio-inspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications," *IEEE Trans. Circuits Syst. I, Reg. Papers* **57** (2010) 850.
- [10] A. Dalloo, *et al.*: "Systematic design of an approximate adder: the optimized lower part constant-OR adder," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **26** (2018) 1595.
- [11] P. Balasubramanian, *et al.*: "An approximate adder with a near-normal error distribution: design, error analysis and practical application," *IEEE Access* **9** (2021) 4518.
- [12] V. Gupta, *et al.*: "Low-lower digital signal processing using approximate adders," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.* **32** (2013) 124.
- [13] N. Zhu, *et al.*: "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.* **18** (2010) 1225.
- [14] J. Lee, *et al.*: "Approximate adder design with simplified lower-part approximation," *IEICE Electron. Express*, **17** (2020) 1.
- [15] W. Liu, *et al.*: "Design and analysis of inexact floating-point adders," *IEEE Trans. Comput.* **65** (2016) 308.
- [16] H. Seo, *et al.*: "Design and analysis of approximate adder with hybrid error reduction," *Electronics* **9** (2020) 471.
- [17] M. Shafique, *et al.*: "A low latency generic accuracy configurable adder," *Proc. Design Autom. Conf. (DAC)* (2015) 81.
- [18] Y. Kim: "An accuracy enhanced error tolerant adder with carry prediction for approximate computing," *IEICE Trans. Smart Process. Comput.* **8** (2019) 324.
- [19] Y. Kim: "A novel approximate adder with enhanced low-cost carry prediction for error tolerant computing," *IEICE Trans. Smart Process. Comput.* **8** (2019) 506.
- [20] H. Waris, *et al.*: "High-performance approximate half and full adder cells using NAND logic gate," *IEICE Electron. Express* **16** (2019) 1.
- [21] H. Seo, *et al.*: "COREA: delay- and energy-efficient approximate adder using effective carry speculation," *Electronics* **10** (2020) 2234.
- [22] Y. Kim, *et al.*: "Energy efficient approximate arithmetic for error resilient neuromorphic computing," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **23** (2015) 2733.
- [23] T. Sato, *et al.*: "Trading accuracy for power with a configurable approximate adder," *IEICE Trans. Electron.* **102** (2019) 260.
- [24] T. Yang, *et al.*: "An accuracy-configurable adder for low-power applications," *IEICE Trans. Electron.* **103** (2020) 68.
- [25] F. Ebrahimi-Azandaryani, *et al.*: "Block-based carry speculative approximate adder for energy-efficient applications," *IEEE Trans. Circuits Syst. II, Exp. Briefs* **67** (2019) 137.
- [26] J. Lee, *et al.*: "A novel approximate adder design using error reduced carry prediction and constant truncation," *IEEE Access* **9** (2021) 1.
- [27] B. Garg, *et al.*: "Reconfigurable carry look-ahead adder trading accuracy for energy efficiency," *J. Signal Process. Syst.* **93** (2021) 99.
- [28] X. Wang: "Laplacian operator-based edge detectors," *IEEE Trans. Pattern Anal. Mach. Intell.* **29** (2007) 886.
- [29] O. Akbari, *et al.*: "RAP-CLA: a reconfigurable approximate carry look-ahead adder," *IEEE Trans. Circuits Syst. I, Reg. Papers* **65** (2018) 1089.
- [30] H. A. F. Almurib, *et al.*: "Approximate DCT image compression using inexact computing," *IEEE Trans. Comput.* **67** (2018) 149.