

IEICE **TRANSACTIONS**

on Electronics

VOL. E102-C NO. 4
APRIL 2019

The usage of this PDF file must comply with the IEICE Provisions on Copyright.

The author(s) can distribute this PDF file for research and educational (nonprofit) purposes only.

Distribution by anyone other than the author(s) is prohibited.

A PUBLICATION OF THE ELECTRONICS SOCIETY



The Institute of Electronics, Information and Communication Engineers
Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3chome, Minato-ku, TOKYO, 105-0011 JAPAN

Trading Accuracy for Power with a Configurable Approximate Adder*

Toshinori SATO^{†a)}, Member, Tongxin YANG^{††}, Student Member, and Tomoaki UKEZONO[†], Member

SUMMARY Approximate computing is a promising paradigm to realize fast, small, and low power characteristics, which are essential for modern applications, such as Internet of Things (IoT) devices. This paper proposes the Carry-Predicting Adder (CPredA), an approximate adder that is scalable relative to accuracy and power consumption. The proposed CPredA improves the accuracy of a previously studied adder by performing carry prediction. Detailed simulations reveal that, compared to the existing approximate adder, accuracy is improved by approximately 50% with comparable energy efficiency. Two application-level evaluations demonstrate that the proposed approximate adder is sufficiently accurate for practical use.

key words: approximate computing, low-power circuit, approximate adder

1. Introduction

Some contemporary applications, such as image processing and deep learning, have an inherent error tolerance, and, typically, such applications process noisy and redundant data, output an acceptable range of results rather than a unique accurate result, and employ statistical computations that can mitigate the impact of errors [1]. Approximate computing is a promising approach to improve performance, circuit area, and power efficiency on the computing of the error tolerance applications [2], [3]. Approximate computation targets potentially error tolerant regions of such applications. By intentionally generating predictable errors, accuracy is compromised for performance, circuit area, and power efficiency. While approximate computing can be applied at any layer of a computing stack, this paper focuses on the elemental circuit level and proposes an accuracy-scalable adder. Note that identifying error tolerant regions is beyond the scope of this paper and is expected to be the focus of future work. Here, it is assumed that a designer or programmer identifies appropriate error tolerant regions and determines how to apply approximate computing.

Several approximate adders have been proposed previously [3]–[8]. Unfortunately, some applications cannot benefit from these approximate adders because most do not include accuracy-scalability [3]. Dynamic configura-

bility is required because different applications and different program phases in an application have different accuracy requirements. Even though high accuracy is not required, fixed-accuracy adder cannot exploit opportunities to reduce power consumption. In addition, with applications which rely on iterative methods, accuracy requirements can vary among iterations [9], [10]. Some existing adders are statically [4], [5] and dynamically [6]–[8] configurable; however, some applications may require further scalability relative to accuracy. This study extends a previous study [11] and focuses on a low-power small adder for accuracy-scalable approximate computing. The current study compares two types of proposed 4-bit adders and includes additional evaluations using two image processing applications.

The remainder of this paper is organized as follows. Related work is reviewed in Sect. 2. The proposed accuracy-scalable adder is described in Sect. 3. The experimental methodology is explained in Sect. 4, and evaluation results are presented in Sect. 5. Conclusions and suggestions for future work are given in Sect. 6.

2. Related Work

A good survey on approximate adders can be found in the literature [3]. This section focuses on accuracy-configurable adders.

Mahdiani et al. [4] proposed the Lower-part-OR adder (LOA), that hybridized conventional precise adders for the upper bits and OR gates for the lower bits, which, in terms of accuracy, are less important than upper bits. Although their LOA is very simple, it is moderately accurate and is power- and area-efficient [3].

Shafique et al. [5] proposed the Generic Accuracy Configurable (GeAr) adder that uses multiple sub-adders to construct a wide adder. The GeAr is a generalized model; therefore, it can be configured as several types of approximate adders.

The LOA and GeAr are not configurable dynamically, i.e., configuration is determined statically during design. In contrast, the targets of this paper are dynamically configurable adders.

Angizi et al. [6] proposed an adder based on spintronic technology that consumes 20.4% less power in approximate mode than in precise mode. Note that, while spintronic circuits represent a promising technology, this paper focuses on CMOS circuits.

Manuscript received July 17, 2018.

Manuscript revised November 29, 2018.

[†]The authors are with Department of Electronics Engineering and Computer Science, Fukuoka University, Fukuoka-shi, 814-0180 Japan.

^{††}The author is with Graduate School of Information and Control Systems, Fukuoka University, Fukuoka-shi, 814-0180 Japan.

*Parts of this paper were presented at SASIMI workshop held in March, 2018.

a) E-mail: toshinori.sato@computer.org

DOI: 10.1587/transele.2018CDP0001

Ye et al. [7] proposed Gracefully-Degrading Adder (GDA), which can dynamically configure its carry prediction scheme. Note that, essentially, the prediction scheme is the carry lookahead circuit of a carry lookahead adder (CLA). It is approximated by reducing the number of bits for carry generation. The weaknesses of the GDA are its large area and power consumption.

2.1 Carry-Maskable Adder

Yang et al. [8] proposed a dynamically configurable Carry-Maskable Adder (CMA). Different from the GDA, the CMA does not rely on redundant circuits or multiplexers. Instead, the CMA combines a conventional precise adder and an approximate one into a simple structure. Thus, its power consumption is comparable to that of a ripple carry adder (RCA) even when configured to operate at the highest possible accuracy.

Figure 1 shows a Carry-Maskable Full Adder (CMFA) [8]. In approximate mode, where the signal mask equals 1 (mask equals 0), the CMFA functions as an OR gate between x and y and C_{out} becomes 0. Note that C_{in} and C_{out} are assumed to be 0 in the approximate mode. Otherwise (mask equals 1), the CMFA functions as a precise full adder (FA).

As shown in Fig. 2, the hierarchical 16-bit CMA is constructed from three 4-bit CMA and a conventional 4-bit precise adder. The 4-bit CMA is built using four CMFAs. Only in the case of a least significant sub-adder, the Carry-Maskable Half Adder (CMHA) is used in the least significant bit. The CMHA is shown in Fig. 3. Similar to the CMFA, the CMHA functions as an OR gate and its carry-out becomes 0 when mask equals 0. Otherwise, it functions as a precise half adder (HA).

The most significant four bits are almost calculated accurately and the remaining 12 bits are approximated. As

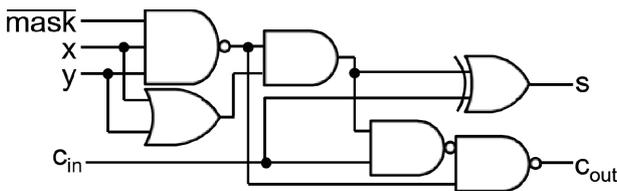


Fig. 1 Carry-maskable full adder.

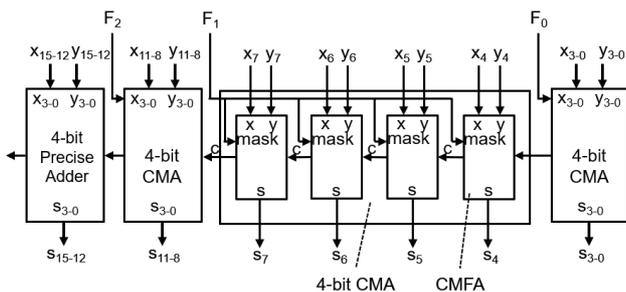


Fig. 2 16-bit carry-maskable adder.

shown in Fig. 2, F_2 , F_1 , and F_0 are configuration bits connected into mask ports in the CMFAs and CMHAs. A single F_n ($n = 0, 1, 2$) configures all carry-maskable adders in a 4-bit sub-adder. For example, when the set of $\{F_2, F_1, F_0\}$ is $\{0, 0, 1\}$, the lower four bits function as OR gates and the remaining 12 bits function as precise FAs.

In the approximate mode, the critical path changes, as shown in Fig. 4. Here, each gray rectangle is a 1-bit adder, two arcs at the top of the rectangle represents an augend and an addend pair, an arc at the bottom is the sum of the pair, and middle horizontal arc represent is a carry. In Fig. 4, thick lines are the carry propagation for different configurations of the mask. In the approximate mode the carry-out is not generated; thus, the critical path begins at the input of the least significant bit that is not in approximate mode and ends at the carry output of the 16-bit adder.

3. Approximate Adder with Carry Prediction

This section proposes an alternative approximate adder, which is referred to as the Carry-Predicting Adder (CPredA). To enhance the accuracy-scalability, the CMA is modified into the CPredA. In this section, the 1-bit Carry-Predicting Full Adder (CPFA) is introduced. Then, multi-bit approximate adders with carry prediction are constructed.

3.1 Carry-Predicting Full Adder

Figure 5 shows the CPFA. The mask bit of the CMFA is re-

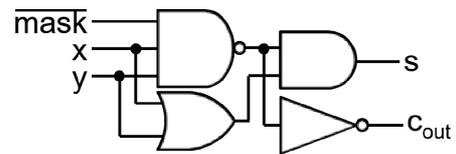


Fig. 3 Carry-maskable half adder.

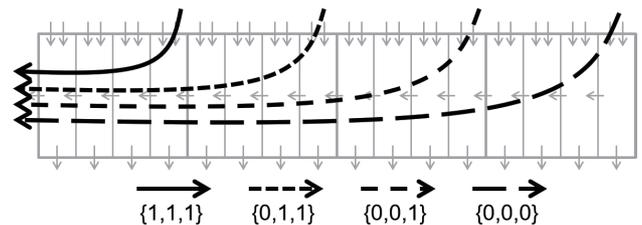


Fig. 4 Critical path of CMA.

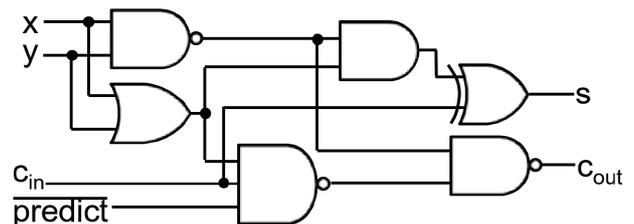


Fig. 5 Carry-predicting full adder.

Table 1 Truth table of FA, CMFA, and CPFA.

cin	FA		CMFA		CPFA	
	x	y	s	cout	s	cout
0	0	0	0	0	0	0
0	0	1	1	0	1	0
0	1	0	1	0	1	0
0	1	1	0	1	0	1
1	0	0	1	0	1	0
1	0	1	0	1	0	0
1	1	0	0	1	0	0
1	1	1	1	1	1	1

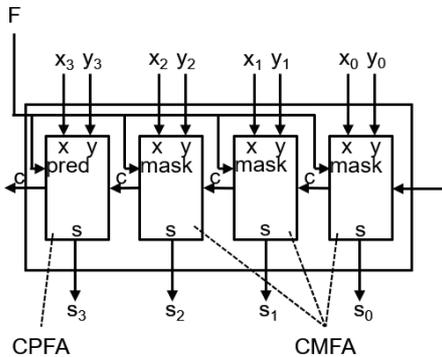


Fig. 6 4-bit carry-predicting adder.

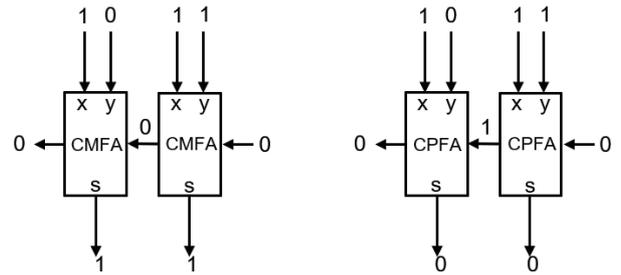


Fig. 7 Carry disappearance in CPredA.

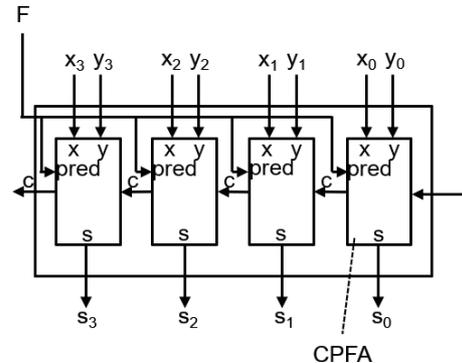


Fig. 8 Alternative 4-bit carry-predicting adder.

placed by the predict bit. When the signal predict equals 1 (predict equals 0), C_{out} is predicted by AND'ing x and y . Note that the sum S is always precise. This is referred to as prediction mode. Otherwise (i.e., predict equals 1), it works as the precise FA.

Table 1 shows the truth table of the conventional precise FA, the approximate mode of the CMFA, and the prediction mode of the CPFA. Recall that C_{in} is always 0 in the approximate mode of the CMFA; thus, the cells where C_{in} is 1 are hatched. In Table 1 ovals represent cases where the approximate adders generate incorrect outputs.

3.2 Multi-Bit Adder with Carry Prediction

The CPredA can be constructed in a manner that is similar to that of multi-bit CMAs. A candidate 4-bit CPredA is shown in Fig. 6, where the most significant bit utilizes the CPFA, and the three trailing bits use the CMFA. This combination is selected because it is expected that generating a carry would improve the accuracy of a 4-bit adder. The configuration signal F is connected to the predict port, which is shown as "pred" in Figs. 6, 8, and 9, in the CPFA and the mask ports in the CMFAs. When F is 1, the CPredA works in the prediction mode, otherwise, it works as the precise 4-bit adder.

The alternative 4-bit adder comprising four CPFAs in Fig. 8 is also a candidate for the 4-bit CPredA. However, there is a concern. Table 1 presents the truth table of the conventional precise FA, the approximate mode of the CMFA, and the prediction mode of the CPFA. In the case where a carry is propagated over two bits on the precise FA, the carry

may disappear on the CPFA. An example is shown in Fig. 7. Here, consider binary addition of $11 + 01$. While its correct answer is 100, the 2-bit CMA (Fig. 7, left) and CPredA (Fig. 7, right) answers are 011 and 000, respectively. The error distances from the correct value is larger with the CPredA. This possible disappearance of carry propagation is a serious issue. Considering the above, it is expected that the heterogeneous CPredA (Fig. 6) would be more accurate than the homogeneous CPredA (Fig. 8).

A 16-bit CPredA is constructed from one conventional precise 4-bit adder and three 4-bit CPFAs, as shown in Fig. 9. Note that this is the same as the 16-bit CMA. When the heterogeneous CPredA is used (Fig. 9(a)), the least significant CMFA is replaced by the CMHA, and, when the homogeneous CPredA is used (Fig. 9(b)), the least significant CPFA is replaced by the conventional precise HA, which does not require the signal predict. These heterogeneous and homogeneous 16-bit CPredAs are called Type 1 and Type 2, respectively. Considering the above discussion about the carry disappearance, it is expected that Type 1 will yield better accuracy than Type 2. On the other hand, with Type 1, some CMFAs receive a carry-in of 1 because the carry-out from the CPFA in the lower 4-bit CPredA is connected to the carry-in of the CMFA in the upper CPredA. This means that the hatched cells in Table 1 should be considered and that an incorrect S may be generated. It is expected that the impact of the incorrect sum is not serious because it occurs infrequently. The above expectations will be evaluated later.

The critical path changes in the prediction mode, as shown in Fig. 10, where the rectangles, arcs, and lines are

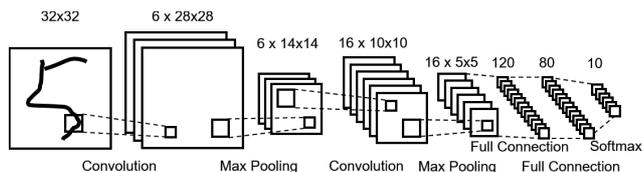


Fig. 11 LeNet-5 convolutional neural network.

of a reconstructed image [15] and evaluate the quality of an application's output. The PSNR metric assesses the quality of reconstructive processes that involve information loss. According to the literature [15], typical PSNR values are between 30 dB and 50 dB in lossy image compression (values greater than 40 dB are considered good). Therefore, in this paper, PSNR values over 30 dB are considered acceptable and those greater than 40 dB are regarded good. These values are used to evaluate information loss in the application. PSNR is defined as follows:

$$PSNR = 10 \cdot \log_{10} \left(\frac{As_{max}^2}{MSE} \right)$$

$$MSE = \frac{1}{X \cdot Y} \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} (s[x, y] - \tilde{s}[x, y])^2$$

where As_{max} , $s[x, y]$, and $\tilde{s}[x, y]$ are the maximum, precise, and approximate values of each pixel, respectively, and X and Y are the image dimensions.

The handwritten digit recognition application utilizes a convolutional neural network. A slight variation of LeNet-5 [16] is implemented in darknet [17]. The network comprises seven layers, as shown in Fig. 11. The input is a 32x32 pixel image. The first layer is a convolutional layer with 5x5 filters, resulting in six 28x28 feature maps. The second layer is a max pooling layer that performs a down-sampling operation using 2x2 filters, resulting in six 14x14 feature maps. The third layer is a convolutional layer with sixteen 10x10 feature maps. The fourth layer is another max pooling layer with sixteen 5x5 feature maps. The next two layers are fully-connected layers with 120 and 84 units, respectively. The last layer is a softmax layer with ten outputs, each of which is associated with a digit. Note that Relu is not used; however, tanh is employed as its activation function. It is expected that smaller numbers are desirable for 16-bit fixed-point operations and tanh outputs are between -1.0 and 1.0. Here, the training is performed using single-precision floating-point operations. Only inference phase is experienced to evaluate the approximate adders. Only add operations in the convolutional and fully-connected layers are calculated using 16-bit fixed-point adders and then are approximate. LeNet-5 includes 341k MAC operations [18], and convolutions account for more than 90% of the overall computation [19]. The training and test sets comprise 60,000 images and 10,000 images, respectively, from the MNIST database [20]. In this evaluation, a declining recognition rate is used as a metric to evaluate the approximate adders.

Table 2 Maximum ED.

Configuration	{0, 0, 1}	{0, 1, 1}	{1, 1, 1}
CPredA Type 1	7	375	6,135
CPredA Type 2	20	340	5,460
CMA	15	255	4,095
GDA	4,096	4,096	4,368

Table 3 MRED ($\times 10^{-6}$).

Configuration	{0, 0, 1}	{0, 1, 1}	{1, 1, 1}
CPredA Type 1	37	911	14,048
CPredA Type 2	74	1,338	20,759
CMA	79	1,340	20,282
GDA	155	2,631	40,392

5. Results

5.1 Accuracy

Table 2 summarizes the maximum ED for the approximate adders. The maximum ED of the GDA is the worst regardless of configuration. The maximum ED of CPredA Type 1 is the greatest among the remaining three adders, except for the {0, 0, 1} configuration. This is an unexpected result.

Table 3 summarizes the MRED for the approximate adders. As shown, all approximate adders have large scalability relative to accuracy. For example, {0, 0, 1} configuration is two orders of magnitude more accurate than the {1, 1, 1} configuration. When the adders are compared to each other, the CPredA Type 2 is comparably accurate to the CMA. In contrast, the CPredA Type 1 shows improved MRED over the CMA by 30% or more. From these observations, the carry propagated from the approximate part to the precise part is most important. For every configuration, CPredA Type 1 is more accurate than Type 2. In addition, Type 2 becomes increasingly less accurate as the number of approximate bits increases. Note that these observations match the expectation discussed in Sect. 3. The accuracy of the multi-bit CPredA suffers from the carry disappearance.

When the accuracy of the two CPredAs is compared, the different metrics result in different winner. The maximum ED indicates that Type 2 shows better accuracy, and the MRED indicates that Type 1 is better. The observation that the MRED is small despite the large maximum ED means that a large ED is rare. Therefore, hereafter, CPredA Type 1 is used as the representative CPredA. However, in this context, there is a concern. Designers and programmers should consider that CPredA Type 1 is occasionally less accurate than the CPredA Type 2 and CMA; thus, when possible, it should not be used in such cases.

5.2 Delay and Circuit Area

Figure 12 shows the delays and circuit areas of the approximate and conventional precise adders. For each group of two bars, the blue one presents delay and the red bar indicates the circuit area. Note that all values are normalized by

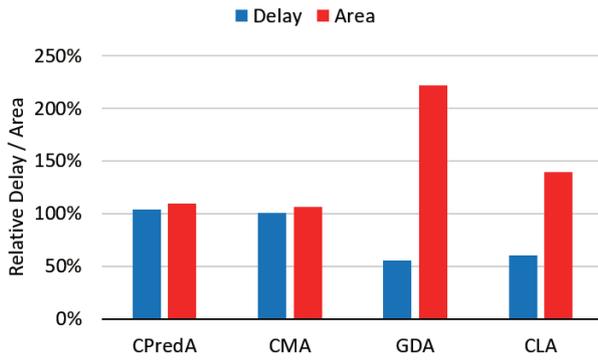


Fig. 12 Delay and area relative to RCA.

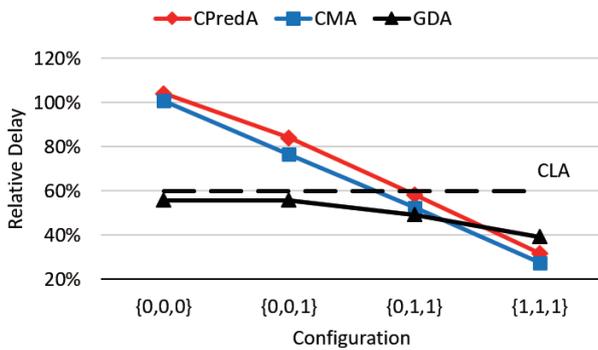


Fig. 13 Impact of configuration on dynamic delay.

those of the RCA; thus, the RCA results are not included in Fig. 12. Relative to delay, the GDA is only comparable to the CLA and it is 45% less than the delay of the CMA and the CPreDA. In contrast, both the CMA and CPreDA show delay to is nearly equivalent to that of the RCA. While both the CMA and CPreDA are based on the RCA, their overhead due to the configurability is very small i.e., less than 5%.

Relative to circuit area, the GDA suffers from the configurability. The area of the GDA is 122% greater than that of the RCA and 59% greater than that of the CLA. Note that such increased area is not negligible when approximate adders are required in area-constrained applications such as Internet of Things (IoT) devices. In contrast, the CMA and CPreDA demonstrate small area, both of which are less than 10% greater than the area of the RCA.

As discussed in Sects. 2.1 and 3.2, the critical path (thus, the longest path delay) changes according to the configuration because some paths in a configuration are always inactive in the other configurations. Such paths should be considered false paths when the longest path delay is evaluated. Figure 13 shows how the delay changes dynamically according to the configuration. Here, the horizontal axis indicates the configuration and the vertical axis is the relative delay normalized by that of the RCA, as in Fig. 12. The red, blue, and black lines represent the CPreDA, the CMA, and the GDA, respectively. That of the CLA is also shown as straight broken lines, because it is not configurable. As can be seen, scalability relative to delay is greater with the CPreDA and CMA than with the GDA. Surprisingly, in the

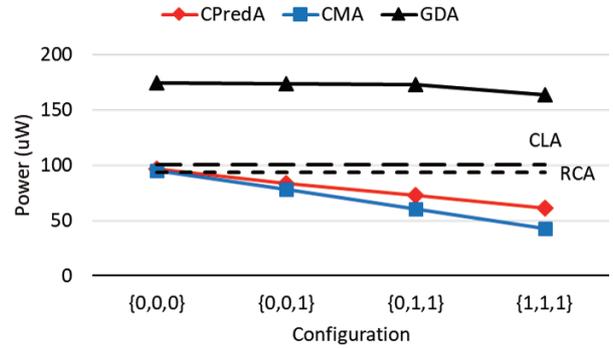


Fig. 14 Impact of configuration on power consumption.

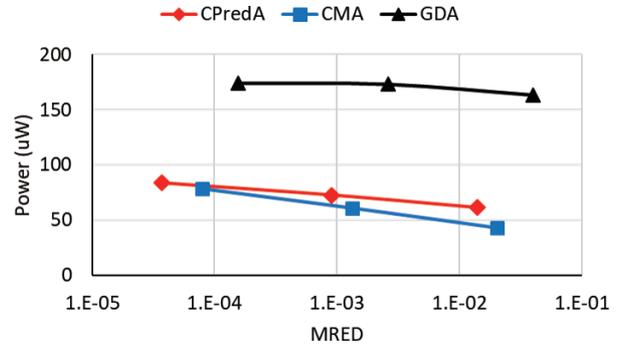


Fig. 15 Relationship between MRED and power consumption.

{1, 1, 1} configuration, both the CPreDA and CMA are faster than the GDA. Compared to the CMA, the delays and areas of the CPreDA are nearly the same.

5.3 Power Consumption

Figure 14 shows the impact of the configurations for the approximate adders relative to their power consumption. Here, the red, blue, and black lines represent the CPreDA, CMA, and GDA, respectively. Those of the RCA and CLA are also shown as straight continuous and broken lines, respectively, because they are not configurable. The horizontal axis indicates the configuration and the vertical axis is power consumption in μW . As can be seen, the GDA consumes much more power than the other adders. This result is expected because the GDA has the largest circuit area, which will be highly active. Note that its precise sub-adders and the carry predictors always work simultaneously. As shown, power-scalability is greater with the CPreDA and CMA than with the GDA.

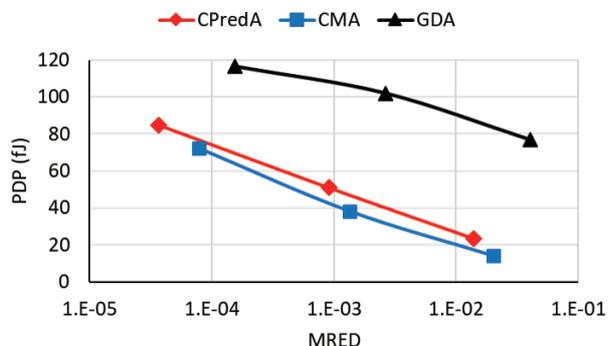
The relationship between accuracy in MRED and power consumption is presented in Fig. 15, where horizontal axis indicates the MRED and the vertical axis indicates power in μW . Note that the horizontal axis follows a logarithmic scale. The red, blue, and black lines indicate the CPreDA, CMA, and GDA, respectively. Naturally, higher accuracy requires larger power consumption. Note that power-scalability is greater in the CMA than in the CPreDA. In contrast, accuracy-scalability is greater in the CPreDA

Table 4 PSNR of image sharpening (dB).

Config.	Baboon	Barbara	F16	Lena	Peppers
CPredA					
{0, 0, 1}	55.50	55.48	55.46	55.42	55.44
{0, 1, 1}	30.42	30.35	29.78	30.13	30.01
{1, 1, 1}	11.69	12.49	12.98	11.84	11.96
CMA					
{0, 0, 1}	50.76	50.74	50.66	50.69	50.71
{0, 1, 1}	24.25	24.51	24.03	23.99	24.05
{1, 1, 1}	9.62	10.56	12.10	9.80	10.37
GDA					
{0, 0, 1}	40.39	40.77	39.18	40.34	40.67
{0, 1, 1}	23.50	24.21	23.76	23.25	23.29
{1, 1, 1}	9.25	10.09	11.82	9.45	9.99

Table 5 Recognition rate (%).

Rate	Floating-point		Fixed-point	
	97.37	97.32	97.32	97.32
CPredA				
{0, 0, 1}	Rate	94.73	70.86	37.05
	Drop	2.59	26.46	60.27
{0, 1, 1}	Rate	10.32	10.32	9.87
	Drop	87.00	87.00	87.45
{1, 1, 1}	Rate	10.07	10.37	10.00
	Drop	86.65	86.95	87.32

**Fig. 16** Relationship MRED and PDP.

than in the CMA. Recall that the MRED is presented in logarithmic scale. As shown in Table 3, the CPredA demonstrates a 379.7 times smaller MRED with the {0, 0, 1} configuration than with the {1, 1, 1} configuration while the CMA demonstrates a 256.7 times smaller MRED with the {0, 0, 1} configuration than with the {1, 1, 1} configuration. As discussed in Sect. 1, some applications require higher accuracy than is provided by the CMA. The above observations indicate that the CPredA is a good selection for the target.

For battery-powered devices, energy is more important than power, because large energy consumption makes providing a continuous power supply difficult. Therefore, the power delay product (PDP), which equals energy, is investigated. Figure 16 shows the relationship between MRED and PDP, where the horizontal axis is the MRED and the vertical axis is the PDP in fJ . Note that the horizontal axis employs a logarithmic scale. Different from power consumption, two approximate adders have similar scalability in PDP. The CPredA and CMA are comparable from an energy efficiency perspective. Relative to the MRED, scalability is greater with the CPredA than with the CMA, as discussed previously.

According to their requirements, each application can effectively select an approximate adder from the CPredA and CMA. Applications that prioritize power should use the CMA, and those that prioritize accuracy should use the CPredA because it demonstrates greater accuracy-scalability than the CMA.

5.4 Image Processing

Table 4 shows the PSNR results for the five images, which are sharpened using the approximate adders under different configurations. Note that the PSNR for the {0, 0, 0} configuration is not given because each configurable adder in the configuration is precise. As can be seen, CPredA demonstrates the highest PSNR for all images and configurations.

Figure 17 shows the processed images when the CPredA is used. Here, from left to right, four images processed under configurations {0, 0, 0} (precise), {0, 0, 1}, {0, 1, 1}, and {1, 1, 1} are shown. For comparison, the original images (prior to sharpening) are shown at the far right. It can be seen in Fig. 17 that the precise and approximate results are nearly indistinguishable visually when the PSNR is greater than 30 dB. Only the PSNR of configuration {1, 1, 1} is less than 20 dB. Its processed image is clearly inferior in quality to the images processed by the other configurations. Thus, this configuration would be unacceptable in image sharpening cases. However, it is expected that it will be useful for applications under the situation of tight battery.

Table 5 shows the recognition rate of handwritten digits, where the upper part shows the rates of the convolutional neural networks, which perform single-precision floating-point and 16-bit fixed-point add operations, respectively. Note that both operations are precise. The lower part of the table summarizes the rates and the declined points of the networks, which perform approximate additions. The declined points show how the recognition rate is lower than the case with precise fixed-point addition.

When the recognition is based on the precise floating-point addition, the score is 97.37%. If floating-point addition is replaced by fixed-point addition, it becomes negligibly smaller. Thus, the score with fixed-point addition is selected as the baseline. When the CPredA is used with the {0, 0, 1} configuration, the recognition rate is reduced by 2.59 point from the baseline to reach 94.73%. This rate is practically sufficient compared to previously reported results [20]. However, a reduced recognition rate is profitable relative to power reduction. In contrast, both the CMA and GDA under the same configuration severely reduce the recognition rate; thus, are inapplicable for practical use. The {0, 1, 1} and {1, 1, 1} configurations are out of account for all approximate adders.

It is interesting that the CMA noticeably reduces the recognition rate compared to the CPredA, while the differ-

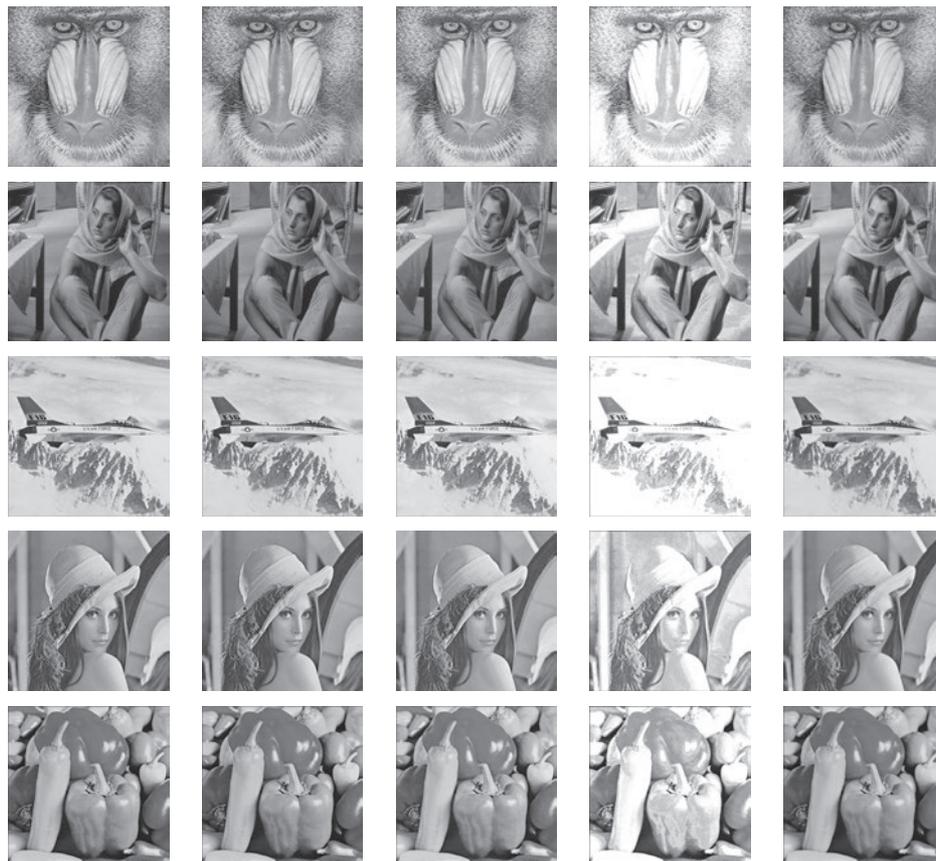


Fig. 17 Processed $(\{0, 0, 0\}, \{0, 0, 1\}, \{0, 1, 1\}, \{1, 1, 1\})$ and original images.

ences in PSNR between the CPredA and CMA cases (Table 4) are insignificant. This is because the CMA does not handle signed values effectively. For example, consider the addition of 2049 and -2048 using the CMA with the $\{1, 1, 1\}$ mask, which is processed as follows:

$$\begin{array}{rcccccl}
 & 0000 & 1111 & 1111 & 1111 & \text{mask} \\
 & 0000 & 1000 & 0000 & 0001 & (2049_{10}) \\
 + & 1111 & 1000 & 0000 & 0000 & (-2048_{10}) \\
 \hline
 & 1111 & 1000 & 0000 & 0001 & (-2047_{10})
 \end{array}$$

While the precise sum is 1, the approximate sum is -2047 . Note that these sums have opposite signs, and their ED is large. For convolutional neural networks, an output of a neuron, which is calculated as the sum of the products of the weights and the inputs, may be opposite to its correct value; thus, the neuron would not fire. This results in incorrect recognition. In contrast, in this case, the CPredA yields an answer of 1, which is the correct sum.

6. Conclusions

This paper has proposed the CPredA, which approximates addition by predicting carry generation. Compared to the existing CMA and GDA, the proposed CPredA demonstrates higher accuracy with comparable delay and area and

with slightly larger power. From an energy efficiency perspective, the proposed CPredA is comparable to the existing CMA. The CPredA is a promising approximate adder that can function as an alternative to the existing CMA.

A future direction of this study will be investigating iterative methods [9], [10]. Currently, reconfigurability is not always utilized completely, and only some scenarios about a residual amount of a battery are considered. Dynamic configuration, which is fine grain regarding time, is an interesting research topic.

Acknowledgements

The authors thank Hiroyuki Baba, Masahiro Inoue, and Kaori Tajima for valuable discussions. This work was supported by JSPS KAKENHI Grant Number JP17K00088, by funds (No.175007 and 177005) from the Central Research Institute of Fukuoka University, and by VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Synopsys, Inc.

References

- [1] V.K. Chippa, S.T. Chakradhar, K. Roy, and A. Raghunathan, "Analysis and characterization of inherent application resilience for approximate computing," Design Automation Conference, 2013. doi:10.1145/2463209.2488873

- [2] K. Roy and A. Raghunathan, "Approximate computing: An energy-efficient computing technique for error resilient applications," IEEE Computer Society Annual Symposium on VLSI, 2015. doi:10.1109/ISVLSI.2015.130
- [3] J. Han, "Part I: Approximate arithmetic circuits and applications," Tutorial at Embedded Systems Week, 2017. http://www.ece.ualberta.ca/~jhan8/publications/IntroAC_ArithmeticCircuits_Tutorial_Part1.pdf [Accessed on July 9, 2018]
- [4] H.R. Mahdiani, A. Ahmadi, S.M. Fakhraie, and C. Lucas, "Bio-inspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications," IEEE Trans. Circuits Syst. I, vol.57, no.4, pp.850–862, 2010. doi:10.1109/TCSI.2009.2027626
- [5] M. Shafique, W. Ahmad, R. Hafiz, and J. Henkel, "A low latency generic accuracy configurable adder," Design Automation Conference, pp.1–6, 2015. doi:10.1145/2744769.2744778
- [6] S. Angizi, Z. He, R.F. DeMara, and D. Fan, "Composite spintronic accuracy-configurable adder for low power digital signal processing," International Symposium on Quality Electronic Design, pp.391–396, 2017. doi:10.1109/ISQED.2017.7918347
- [7] R. Ye, T. Wang, F. Yuan, R. Kumar, and Q. Xu, "On reconfiguration-oriented approximate adder design and its application," International Conference on Computer-Aided Design, pp.48–54, 2013. doi:10.1109/ICCAD.2013.6691096
- [8] T. Yang, T. Ukezono, and T. Sato, "A low-power configurable adder for approximate applications," International Symposium on Quality Electronic Design, pp.347–352, 2018. doi:10.1109/ISQED.2018.8357311
- [9] Q. Zhang, F. Yuan, R. Ye, and Q. Xu, "ApproxIt: An approximate computing framework for iterative methods," Design Automation Conference, pp.1–6, 2014. doi:10.1109/DAC.2014.6881424
- [10] H.-J. Wunderlich, C. Braun, and A. Schöll, "Pushing the limits: How fault tolerance extends the scope of approximate computing," International Symposium on On-Line Testing and Robust System Design, pp.133–136, 2016. doi:10.1109/IOLTS.2016.7604686
- [11] H. Baba, T. Yang, M. Inoue, K. Tajima, T. Ukezono, and T. Sato, "A carry-predicting full adder for accuracy-scalable computing," Workshop on Synthesis and System Integration of Mixed Information Technologies, 2018. <http://sasimi.jp/files/2018/archive/SASIMI2018/pdf/R1-11.pdf> [Accessed on July 9, 2018]
- [12] C. Liu, J. Han, and F. Lombardi, "A low-power, high-performance approximate multiplier with configurable partial error recovery," Design, Automation & Test in Europe, pp.1–4, 2014. doi:10.7873/DATE.2014.108
- [13] NanGate Inc., "FreePDK45 open cell library," <http://www.nangate.com/> [Accessed on July 9, 2018]
- [14] R. Szeliski, Computer Vision - Algorithms and Applications, Springer-Verlag, 2011.
- [15] D. Bull, Communicating Pictures - A Course in Image and Video Coding, Academic Press, 2014.
- [16] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proc. IEEE, vol.86, no.11, pp.2278–2324, Nov. 1998. doi:10.1109/5.726791
- [17] J. Redmon, "Darknet: Open source neural networks in C," <https://pjreddie.com/darknet/> [Accessed on July 9, 2018]
- [18] V. Sze, Y.-H. Chen, T.-J. Yang, and J.S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," Proc. IEEE, vol.105, no.12, pp.2295–2329, Dec. 2017. doi:10.1109/JPROC.2017.2761740
- [19] J. Emer, V. Sze, and Y.-H. Chen, "Hardware architectures for deep neural networks," Tutorial at International Symposium on Computer Architecture, 2017. <http://www.rle.mit.edu/eems/wp-content/uploads/2017/06/ISCA-2017-Hardware-Architectures-for-DNN-Tutorial.pdf> [Accessed on July 9, 2018]
- [20] Y. LeCun, C. Cortes, and C.J.C. Burges, "The MNIST database of handwritten digits," <http://yann.lecun.com/exdb/mnist/> [Accessed on July 9, 2018]



Toshinori Sato received the B.S., M.S., and Ph.D. degrees in electronic engineering from Kyoto University in 1989, 1991, and 1999, respectively. From 1991 to 1999, he was with Toshiba Corporation, where he was engaged in the research and developments of processors. He is currently a professor of Department of Electronics Engineering and Computer Science at Fukuoka University. He previously served on the faculty of Department of Artificial Intelligence at Kyushu Institute of Technology and of System LSI Research Center at Kyushu University. His research interests include computer architecture and design methodology. He is a senior member of ACM and IEEE and a member of IPSJ.



Tongxin Yang received B.S. degree from Harbin University of Science and Technology, China and joined Logic Research Co., Ltd, Japan in 2008. Since 2016, he has been working toward the Ph.D. degree in Graduate School of Information and Control Systems, Fukuoka University, Japan. His research interests include approximate computing systems and low power circuits and systems. He is a student member of IEEE.



Tomoaki Ukezono graduated from the School of Information Science, Japan Advanced Institute of Science and Technology (JAIST). He received the Ph.D. degree from JAIST in 2010. Tomoaki joined Center for Highly Dependable Embedded Systems Technology, JAIST as researchers in 2010. Tomoaki joined School of Information Science, JAIST as assistant professors in 2011. Currently, Tomoaki is with Department of Electronics Engineering and Computer Science, Fukuoka University as assistant professors from 2015. His current research interests include computer architecture and operating system. Tomoaki is a member of the Institute of Electronics, Information and Communication Engineers (IEICE) of Japan.