IEICE TRANSACTIONS

on Electronics

VOL. E103-C NO. 3
MARCH 2020

A PUBLICATION OF THE ELECTRONICS SOCIETY

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

IEICE TRANS. ELECTRON., VOL.E103–C, NO.3 MARCH 2020

PAPER Special Section on Low-Power and High-Speed Chips

An Accuracy-Configurable Adder for Low-Power Applications**

Tongxin YANG[†*a)], **Toshinori SATO**[††], ***and*** **Tomoaki UKEZONO**[††], ***Members***

SUMMARY Addition is a key fundamental function for many error-tolerant applications. Approximate addition is considered to be an efficient technique for trading off energy against performance and accuracy. This paper proposes a carry-maskable adder whose accuracy can be configured at runtime. The proposed scheme can dynamically select the length of the carry propagation to satisfy the quality requirements flexibly. Compared with a conventional ripple carry adder and a conventional carry look-ahead adder, the proposed 16-bit adder reduced the power consumption by 54.1% and 57.5%, respectively, and the critical path delay by 72.5% and 54.2%, respectively. In addition, results from an image processing application indicate that the quality of processed images can be controlled by the proposed adder. Good scalability of the proposed adder is demonstrated from the evaluation results using a 32-bit length.

key words: *approximate computing, accuracy-configurable adder, carry-maskable adder, low-power adder*

1. Introduction

Many increasingly popular applications, such as image processing and recognition, which are computationally demanding, have created challenges relative to power consumption. Most of these applications are inherently tolerant of small inaccuracies; therefore, there are unprecedented opportunities to reduce power consumption. Addition is a fundamental arithmetic function for such applications [1], [2]. Approximate computing is an efficient approach for error-tolerant applications because it can trade accuracy for power. Currently, this tradeoff plays a significant role in such application domains [3]. Since the quality requirements of an application may vary significantly at runtime, it is advantageous to design quality-configurable systems that are able to trade off computation quality according to application requirements [4], [5].

We focused on the structure of an accuracy-configurable adder design from the aspect of power consumption [6]. Our primary contribution is to achieve accuracy configurability efficiently by slightly modifying a conventional adder so that some of its logic gates can be reused. We have proposed a carry-maskable adder (CMA) in which the generation circuit of each bit of its sum can be dynamically configured to function as a full adder or as an OR gate [6]. This configurability was realized by masking the carry propagation. We implemented a 16-bit CMA, a 16-bit conventional ripple carry adder (RCA), and a 16-bit carry look-ahead adder (CLA) in Verilog HDL, by using a 45-nm library, and evaluated their power consumptions, critical path delays, and design areas. The comparisons with the conventional RCA and CLA showed that, with a 1.95% mean relative error distance (MRED), the proposed adder reduces power consumption by 54.1% and 57.5%, respectively. We provided a crosswise comparison to demonstrate the superiority of the 16-bit CMA compared to the existing approach. We implemented one of the established accuracy-configurable adders to evaluate power consumption, design area, critical path delay, and accuracy. We also evaluated the quality of these two accuracy-configurable adders under a real image processing application [6]. We added a detail power analysis for the 16-bit CMA with different configuration settings. We analyzed randomly generated patterns, which are used for power evaluation by using a mathematical method to address the aspect of quantity. We also implemented and evaluated a 32-bit CMA to demonstrate its scalability.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 introduces the structure of CMA. Section 4 evaluates the 16-bit CMA in terms of power consumption, critical path delay, design area, and accuracy. An application in image processing is also used to evaluate its quality. Section 5 analyzes and discusses the power consumption and scalability of the CMA in detail. Conclusions are presented in Sect. 6.

2. Related Work

Gupta et al. [7] discussed how to simplify the complexity of a conventional mirror adder cell at the transistor level. Mahdiani et al. [8] proposed a lower-part-OR adder, which utilizes OR gates for the addition of the lower bits and precise adders for the addition of the upper bits. Venkatesan et al. [9] proposed to construct an equivalent untimed circuit that represents the behavior of an approximate circuit. Miao et al. [10] introduced an aligned fixed internal-carry structure and then proposed a dithering approximate adder by trading off error magnitude and error frequency. Du

Manuscript received June 28, 2019.
Manuscript revised October 25, 2019.

[†]The author was with Graduate School of Information and Control Systems, Fukuoka University, Fukuoka-shi, 814–0180 Japan.
[††]The authors are with Department of Electronics Engineering and Computer Science, Fukuoka University, Fukuoka-shi, 814–0001 Japan.
[*]Presently, with Logic Research Co., Ltd., Japan.

[**]An earlier version of this paper [6] was presented at 19th International Symposium on Quality Electronic Design, pp.347–352, March 2018.

a) E-mail: yangtongxin@logic-research.co.jp

DOI: 10.1587/transele.2019LHP0002

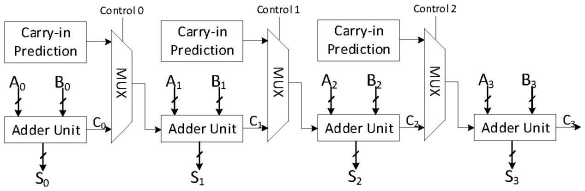

Fig. 1 Accuracy gracefully-degrading adder in [5].

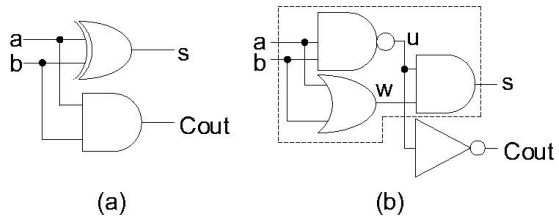

Fig. 2 (a) Conventional half adder, and (b) equivalent circuit of a half adder.

Table 1 Truth table for the equivalent circuit of a half adder.

Inputs		Internal signals		Outputs	
a	b	u	w	s	Cout
0	0	1	0	0	0
0	1	1	1	1	0
1	0	1	1	1	0
1	1	0	1	0	1

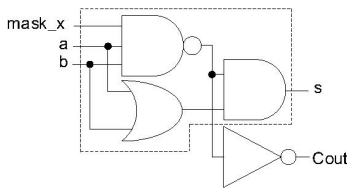

Fig. 3 Carry-maskable half adder.

et al. [11] described a speculative carry select adder with reliable variable latency to detect errors and recover results.

In practice, the computation quality requirement of an application may vary significantly at runtime. The aforementioned static approximate designs [7]–[11] with a fixed accuracy may fail to meet the application quality requirements or otherwise waste power when a high quality is not required. This means that approximate adders should be dynamically configurable to match the different quality requirements of different program phases. To adapt to varying accuracy requirements from different workloads, Kahng et al. [4] proposed accuracy-configurable adder (ACA) based on a pipeline structure. The correction scheme of the ACA proceeds from stage 1 to stage 4. This means that, if the most significant bits of the results are required to be correct, all of the four stages should be performed.

Motivated by the above, Ye et al. [5] proposed an accuracy gracefully-degrading adder (GDA). As illustrated in Fig. 1, each sub-adder block, except the rightmost one, has its own carry-in prediction block, adder unit, and multiplexer. Carry-out signals can be selected from either the adder units or carry-in prediction blocks by control signals in any order. Similar to [5], the adder proposed in this paper does not consider a pipeline structure. To generate outputs with different levels of computation accuracy and to obtain the configurability of accuracy, some multiplexers and additional logic blocks are required in [5]. The additional logic blocks cause area overhead and power waste when their outputs are not used to generate a sum. As shown in the GDA in Fig. 1, if all sums (S0, S1, S2, and S3) are required to be accurate, the power consumption of the carry-in prediction logic blocks will be wasted. To tackle this problem, only a carry-mask signal was added to our proposed adder in order to achieve the configurability of accuracy. Therefore, no additional circuits, such as a carry-in prediction or error recovery logic blocks, are required.

Sato et al. [15] proposed a carry-predicting adder (CPredA), which also requires no additional circuits nor recovery logic blocks for carry-in prediction. CPredA improves its accuracy at the expense of energy consumption, and hence it targets different application domains from the CMA, such as those where higher accuracy is prioritized over a lower power usage.

3. Carry-Maskable Adder

A conventional half adder is shown in Fig. 2 (a). A 2-input XOR gate is used to generate the sum s and a 2-input AND gate is used to generate carry Cout. An equivalent circuit of a half adder is shown in Fig. 2 (b). The dashed frame represents an equivalent circuit of a 2-input XOR gate. Since there is a 2-input NAND gate in the dashed frame, we reuse it and add an INV gate to generate the carry signal Cout. The outputs of the 2-input NAND and OR gates in the dashed frame are named u and w, respectively. Table 1 presents the truth table for the equivalent circuit of a half adder.

As shown in Fig. 2 (b) and Table 1, when the internal signal u is 1, the sum s is equal to a *OR* b and the carry Cout is 0. This means that, if u is controllable and can be controlled to 1, the carry propagation will be masked and the sum s will be equal to a *OR* b. The sum s (= a *OR* b) is different from the accurate sum (= a *XOR* b) only when both a and b are 1. In other words, the sum s (= a *OR* b) can be considered as an approximate sum. The selectivity between the accurate and approximate sums can be achieved by a control signal, which is used to control u to be a *NAND* b, or to be 1.

We add a signal named mask_x as the control signal and use a 3-input NAND gate to replace the 2-input one in the dashed frame. This is called a carry-maskable half adder (CMHA) and shown in Fig. 3. When mask_x = 0, the sum s = a *OR* b, and the carry Cout = 0; otherwise, when mask_x = 1, the sum s = a *XOR* b, and the carry Cout = a *AND* b. Similar considerations apply to a full adder, which is shown in Fig. 4. When mask_x = 0 and Cin = 0, the sum s = a *OR* b, and the carry Cout = 0, then obviously switching activities become smaller, and dynamic power consumption is reduced. This full adder is called a carry-maskable full

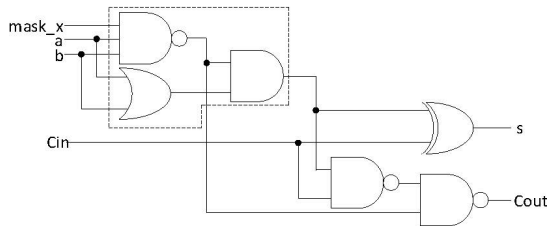

Fig. 4 Carry-maskable full adder.

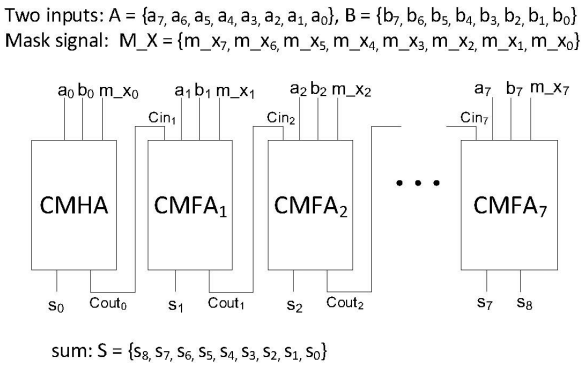

Fig. 5 An 8-bit carry-maskable adder.

adder (CMFA). An n-bit adder, which is implemented using one CMHA and (n-1) CMFA, is called an n-bit CMA.

Figure 5 shows an example of an 8-bit CMA. The carry-mask signal M_X comprises eight bits, which are denoted as m_x_0, m_x_1, $\cdots$, m_x_7. The left is the least significant bit in Fig. 5. The sum and carry of the CMHA are s_0 and $Cout_0$, respectively. Cin_1 is connected to $Cout_0$. When m_x_0 is equal to 0, $s_0 = a_0$ *OR* b_0, and $Cin_1 = Cout_0 = 0$. When both m_x_1 and m_x_0 are equal to 0, $s_0 = a_0$ *OR* b_0, $Cin_1 = Cout_0 = 0$, $s_1 = a_1$ *OR* b_1, and $Cout_1 = 0$ (Cin_2 is also 0). In other words, the carry propagation from CMHA to CMFA1 is masked. By expanding the above equations to CMFA7, when all m_x_0, m_x_1, $\cdots$, m_x_7 are 0, all $Cout_0$, $Cout_1$, $\cdots$, $Cout_7$ are 0, and $s_0 = a_0$ *OR* b_0, $s_1 = a_1$ *OR* b_1, $\cdots$, $s_7 = a_7$ *OR* b_7, and $s_8 = 0$ ($s_8 = Cout_7$). Thus, the carry propagation from CMHA to CMFA7 is masked. Note that there are two conditions for masking the carry propagation of a CMFA: both m_x and Cin′s being 0. Considering the above 8-bit CMA, if we want to mask the carry propagation from CMHA to CMFA3, we should set m_x_0, m_x_1, m_x_2, and m_x_3 to 0 (not set only m_x_3 to 0) to ensure that Cin_1, Cin_2, and Cin_3 are equal to 0.

Each CMFA, as well as the CMHA has its own carry-mask signal in a CMA. Considering a 16-bit CMA, a 16-bit M_X signal (m_x_0, m_x_1, $\cdots$, m_x_{15}) is required. To simplify the structure of a CMA, we can also group some CMFAs as a sub-adder unit. Figure 6 shows a 16-bit CMA with four sub-adder units. Each sub-adder unit has four CMFAs (except for sub-adder unit 0: one CMHA and three CMFAs) and 1-bit carry-mask signal to mask carry propagation. There is no carry-mask signal for sub-adder unit 3 in this example. The structure of sub-adder unit 1 is shown in Fig. 7

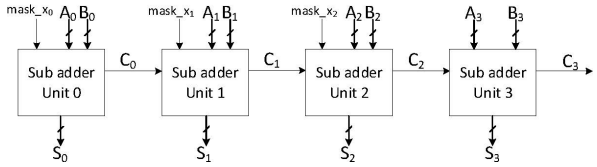

Fig. 6 A 16-bit CMA with four sub-adder units.

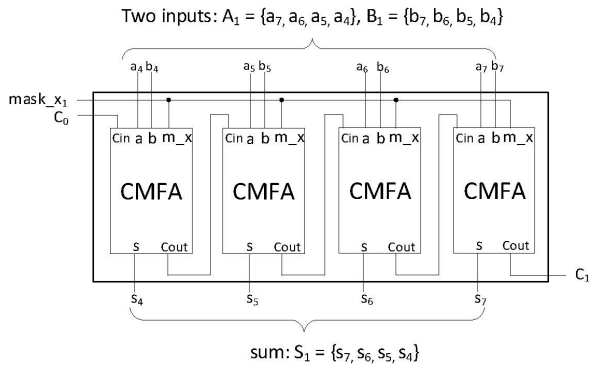

Fig. 7 Structure of sub-adder unit 1.

as an example. C_0 is the output of sub-adder unit 0 and 1-bit $mask\_x_1$ is the carry-mask signal for sub-adder unit 1. If $mask\_x_1 = 0$ and $C_0 = 0$, we can obtain $C_1 = 0$ and $S_1 = A_1$ *OR* B_1 (4-bit parallel OR function). Note that the bit length of each sub-adder unit can be different.

4. Experimental Results

4.1 Experimental Setup

In this section, the proposed adder is evaluated in terms of computational accuracy, power consumption, critical path delay, and design area. To clarify the contributions to the power saving of the proposed adder, we implemented and evaluated a CMA and the conventional RCA, CLA, and GDA [5]. All of them are 16-bit adders. We implemented a full adder of the 16-bit RCA with CMFA (Fig. 4), except for the 3-input NAND gate in the dashed frame replacing the 2-input NAND gate in RCA.

The 16-bit CLA is implemented using five 4-bit carry look-ahead units: four 4-bit carry look-ahead units in stage 1 and one 4-bit carry look-ahead unit in stage 2. The bit lengths of the sub-adder units in the GDA and CMA are both set to four bits. The numbers of carry-in prediction bits in GDA and carry unmasked bits in CMA are both set to 0, 4, 8, and 12 bits. Thus, the configuration settings of GDA and CMA are the same. The adders are referred to as GDA1, GDA2, GDA3, GDA4, CMA1, CMA2, CMA3, and CMA4. For example, in Fig. 6, CMA1 denotes that sub-adder units 0, 1, and 2 are all masked ($mask\_x_0 = mask\_x_1 = mask\_x_2 = 0$), and the accuracy of CMA1 will be the worst among the CMAs. CMA2 denotes that sub-adder units 0 and 1 are masked ($mask\_x_0 = mask\_x_1 = 0$), but sub-adder unit 2 is unmasked ($mask\_x_2 = 1$). CMA4 denotes that sub-adder

units 0, 1, and 2 are all unmasked (mask_x_0 = mask_x_1 = mask_x_2 = 1). With these settings, accurate results can be obtained.

The adders were coded using Verilog HDL. Synopsys VCS was used to simulate the designs and generate value change dump (VCD) files to evaluate the power consumption precisely. Synopsys Design Compiler was used to synthesize the adders with the NanGate 45nm Open Cell Library [12]. The power consumption was evaluated at a frequency of 0.5 GHz. The operating conditions for synthesis employed typical conditions (1.00 process factor, 1.1V power supply, and 25°C operating temperature). All designs were synthesized and optimized using the default compile options. Synopsys Power Compiler was used to estimate power consumption from switching activity interchange format files generated from the VCD files. The Synopsys VCS was used to evaluate the accuracy and power consumption of all of the adders. For accuracy evaluation, $2^{16} \times 2^{16}$ input patterns were used for these 16-bit adders. One million randomly generated input patterns were used for power evaluation. The reason for using one million patterns is because a VCD file of 16-bit adders with one million patterns is already very large. Whether one million patterns are sufficient for the power evaluation is analyzed in Sect. 5.

4.2 Accuracy Results

The error distance (ED) and normalized mean error distance (NMED) are proposed for the evaluation of the performance of the approximate arithmetic circuits [13]. ED is defined as the arithmetic difference between the accurate sum (S) and the approximate sum (S′): ED=|S-S′|. The mean error distance (MED) is the average of EDs for a set of outputs. NMED is defined as NMED=MED/S_{max}, where S_{max} is the maximum magnitude of the output of an accurate adder. The relative error distance (RED) is the ED divided by the accurate output: RED=|S-S′|/S, whereas MRED is the average of REDs and can be obtained similarly to MED. The error rate (ER) is the percentage of inaccurate outputs among all outputs generated from all combinations of inputs. These three metrics (i.e., NMED, MRED, and ER) are used to evaluate the adders.

Table 2 compares the accuracy of the test results and shows that the accuracy of both the CMA and GDA changed widely based on the configuration settings; smaller values represent better results. Both the NMED and MRED of the CMA are smaller than those of the GDA at each setting. As expected, there are no errors in the CMA4 and GDA4. Although the ER value of the CMA is larger than that of the GDA in each accuracy configuration setting, the NMED and MRED of the CMA are about 50% of the GDA. This is because the ED is larger in the GDA than in the CMA. While the GDA generates inexact results less frequently than the CMA does, the negative impact of each error is much larger in the GDA than in the CMA.

Table 2 Accuracy comparison.

	NMED (10^{-4})	MRED (10^{-4})	ER (%)
CMA1	78.11	202.82	96.83
CMA2	4.86	13.40	89.99
CMA3	0.29	0.79	68.36
CMA4	0.00	0.00	0.00
GDA1	156.21	403.92	85.01
GDA2	9.73	26.31	5.86
GDA3	0.57	1.55	0.18
GDA4	0.00	0.00	0.00

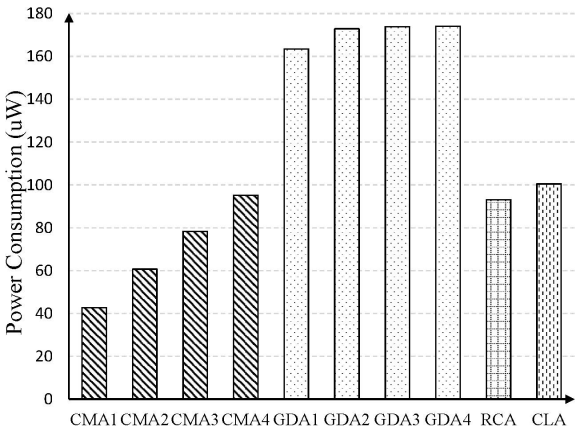

Fig. 8 Power consumption results.

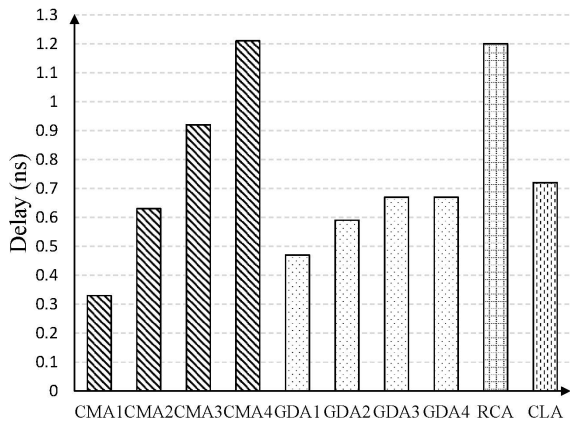

Fig. 9 Critical path delay results.

4.3 Power, Delay, and Area Results

The comparisons of the power consumption and critical path delay for the different adders are shown in Figs. 8 and 9. The x-axes denote the adders with different configuration settings of the CMA and GDA, as well as the conventional accurate RCA and CLA, whereas the y-axes denote the power consumption and critical path delay.

As shown in Fig. 8, the CMA1 has the smallest power consumption among the adders. Compared with the RCA and CLA, the CMA1 delivers 54.1% and 57.5% of power consumption reductions, respectively. Owing to the carry-maskable structure of the CMA, power consumption in-

creases in a linear manner from CMA1 to CMA4. This linearity is analyzed in detail in Sect. 5. The power consumption of the CMA4 is slightly larger than that of the RCA. Remember that our proposed CMA is an accuracy-configurable adder and the CMA4 delivers an accurate result. Compared with another accuracy-configurable adder GDA with the same configuration settings, the power consumption of the GDA4 is 1.8 times larger than that of the CMA4. Furthermore, the power consumption of the GDA1 is 3.8 times larger than that of the CMA1.

Figure 9 demonstrates that the CMA1 has the smallest delay among the adders. The linearity of the delay can also be found in the order from CMA1 to CMA4, with the delay of the CMA4 being the largest among the adders. As can be seen, just as the delay of the CMA4 is close in value to that of the RCA, the delay of the GDA4 is close to that of the CLA, demonstrating that the accuracy configurability of the CMA is based on the structure of the RCA and, likewise, that of the GDA is based on the structure of the CLA. The delay of the CMA4 is slightly larger than that of the RCA. The critical path of an adder is the carry propagation path, and the critical paths of both adders are taken from the inputs at bit position 1 (a_1, b_1) to the sum at bit position 15 (s_{15}). The internal delay of the CMFA at bit position 1 in the CMA4 is slightly larger than that of the full adder at bit position 1 in the RCA because the CMFA is implemented using a 3-input NAND gate and the RCA is implemented using a 2-input NAND gate. Although the delay of the GDA1 is larger than that of the CMA1, the GDA delivers unmistakably good results with regard to delay as a whole. However, the GDA has the largest power consumption among all of the adders.

From the aspect of energy, the power-delay product (PDP) is proposed as a way to evaluate approximate arithmetic circuits [2]. Figure 10 shows a comparison of PDP results relative to the MRED, in order to clarify the contributions of the proposed adder to power saving and accuracy. The circles and triangles represent CMA and GDA, respectively. Smaller values represent better results in energy savings. The CMA1 delivers the best results. The PDP of the GDA1 is 4.4 times larger than that of the CMA1, and the PDP of the GDA4 is 1.2% larger than that of the CMA4. As can be seen, the CMAs with all of the different configuration settings are plotted at the bottom left of Fig. 10. This means that when the same accuracy (MRED) is required, the energy consumption (PDP) of a CMA is smaller than that of a GDA; when the same limited energy is supplied, the accuracy of a CMA is higher than that of a GDA. Figure 10 demonstrates that the proposed CMA definitely achieves good energy savings.

A comparison of the design areas, in square microns, is shown in Table 3. Note that the accuracy configuration setting does not have any effect on the design areas of the CMA and GDA. Although the CMA is slightly larger than the RCA, its area is 76.5% of the CLA and 48.1% of the GDA. As expected, the design area of the RCA is the smallest among the adders. The accuracy comparison results of

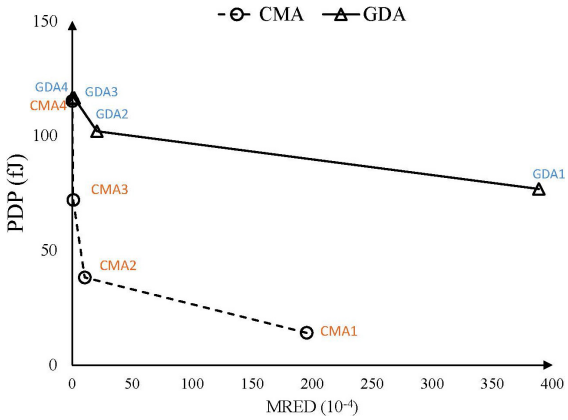

Fig. 10 PDP results relative to MRED.

Table 3 Area results.

(um^2)	CMA	GDA	RCA	CLA
Area	83.79	174.23	78.47	109.59

the adders demonstrate that the CMA consumes the smallest power while maintaining the small area.

4.4 Image Processing

In this section, an image processing application of the proposed adder was evaluated. An image sharpening algorithm [14] that is popular in the evaluation of approximate adders was used. Six 512×512 8-bit grayscale bitmap images collected from the Internet were used. Only the additions were replaced by the tested adders, while all of the other operations (i.e., multiplication, subtraction, and division) were accurate. Similar to [5], the processed image quality was measured using the peak signal-to-noise ratio (PSNR). This is usually used to measure the quality of reconstructive processes that involves information loss.

Table 4 presents the PSNR results of these configurable adders in dB. Larger values represent better quality images. As can be seen, all PSNR values of the CMA are larger than those of the GDA at the same configuration settings, excepted for the results of image No. 2 wherein the values of the CMA2 and GDA2 are close; this demonstrates that the CMA delivers better quality images than the GDA. The CMA4 and GDA4 are accurate, with no PSNR results. Note that more detailed PSNRs can be obtained by changing the bit length of the sub-adders.

5. Analysis of Power and Scalability

5.1 Power Analysis

As shown in Fig. 8, the differences in power consumption between the CMA2 and CMA1, between the CMA3 and CMA2, and between the CMA4 and CMA3 are 17.94, 17.65, and 16.88 uW, respectively. The differences are defined as P_{2-1}, P_{3-2}, and P_{4-3}, respectively; we obtain the values of P_{2-1} that are nearly the same as those of P_{3-2}, and

Table 4 PSNR results of the CMA and GDA with different configuration settings in dB.

Image No. & Description	CMA1	CMA2	CMA3	GDA1	GDA2	GDA3
1. Lena	7.79	27.01	49.60	7.45	25.86	40.58
2. Some peppers	8.88	27.83	51.49	8.32	27.90	41.86
3. A bridge	12.11	28.44	51.38	11.05	25.44	42.15
4. A truck on grassland	9.62	27.79	51.68	8.79	26.61	40.18
5. A bird standing in a stream	11.20	27.35	49.67	10.34	26.12	40.09
6. A view of a small town	8.88	27.24	49.60	8.43	24.93	39.45

P_{2-1} and P_{3-2} are slightly larger than P_{4-3}.

In each sub-adder unit, when the mask signal is equal to 0, the values of the carries are 0; when the mask signal is equal to 1, the value of the output carry of each CMFA (or CMHA) depends on the two inputs and the input carry. The probability of carry occurrence (the value of a carry is equal to 1) can be obtained as:

$$P_{C_0} = \frac{1}{4}, i = 0 \quad (1)$$

$$P_{C_i} = \frac{1}{4} + \frac{1}{2}P_{C_{i-1}}, i > 0 \quad (2)$$

where i is the bit position, and P_{C_i} is the probability for carry occurrence.

Due to the fact that power is consumed when the values of the carries are changed (switching), the probability of carry changing is used to analyze power consumption and it can be calculated from the probability of carry occurrence. From Eq. (1), we can obtain that the probability of $c_0 = 1$ is 0.25; thus, the probability of $c_0 = 0$ is 0.75. When c_0 is changed from 1 to 0, the probability of c_0 changing from 1 to 0 can be obtained by 0.25×0.75; and when c_0 is changed from 0 to 1, the probability of c_0 changing from 0 to 1 can be obtained by 0.75×0.25. Thus, the probability of c_0 changing can be obtained by $2 \times (0.25 \times 0.75)$ and the result is 37.5%. The probability of carry changing can be obtained by:

$$P_{C_i\_chg} = 2 \times P_{C_i} \times (1 - P_{C_i}) \quad (3)$$

where i is the bit position, and $P_{C_i\_chg}$ is the probability of carry changing at the bit position i. Table 5 summarizes the probabilities of carry changing for the CMA1, CMA2, CMA3, and CMA4. The differences of the probabilities of carry changing between the CMA2 and CMA1, between the CMA3 and CMA2, and between the CMA4 and CMA3 can be found from bit positions 12 to 15 (the blue numbers) in the CMA2, CMA3, and CMA4, respectively. The values of these blue numbers are very close; thus, the differences in the power consumptions (P_{2-1}, P_{3-2}, and P_{4-3}) will be nearly the same if the corresponding circuits (capacities) are the same. Note that although $P_{C_0\_chg}$ in the CMA4 is equal to $P_{C_4\_chg}$ in the CMA3, $P_{C_8\_chg}$ in the CMA2, and $P_{C_{12}\_chg}$ in the CMA1, the adder at bit position 0 is CMHA while the adders at other bit positions are CMFAs. Since the circuit in a CMHA is smaller than that of a CMFA, the power consumption of a CMHA is lower than that of a CMFA when the probabilities of carry changing are the same. Therefore, P_{2-1} is nearly the same as P_{3-2}, and P_{2-1} and P_{3-2} are slightly larger than P_{4-3}.

Table 5 Probabilities of carry changing for CMAs with different accuracy configurations.

Probability of carry changing (%)	CMA1	CMA2	CMA3	CMA4
$P_{C_0\_chg}$	0.00	0.00	0.00	37.50
$P_{C_1\_chg}$	0.00	0.00	0.00	46.88
$P_{C_2\_chg}$	0.00	0.00	0.00	49.22
$P_{C_3\_chg}$	0.00	0.00	0.00	49.80
$P_{C_4\_chg}$	0.00	0.00	37.50	49.95
$P_{C_5\_chg}$	0.00	0.00	46.88	49.99
$P_{C_6\_chg}$	0.00	0.00	49.22	50.00
$P_{C_7\_chg}$	0.00	0.00	49.80	50.00
$P_{C_8\_chg}$	0.00	37.50	49.95	50.00
$P_{C_9\_chg}$	0.00	46.88	49.99	50.00
$P_{C_{10}\_chg}$	0.00	49.22	50.00	50.00
$P_{C_{11}\_chg}$	0.00	49.80	50.00	50.00
$P_{C_{12}\_chg}$	37.50	49.95	50.00	50.00
$P_{C_{13}\_chg}$	46.88	49.99	50.00	50.00
$P_{C_{14}\_chg}$	49.22	50.00	50.00	50.00
$P_{C_{15}\_chg}$	49.80	50.00	50.00	50.00

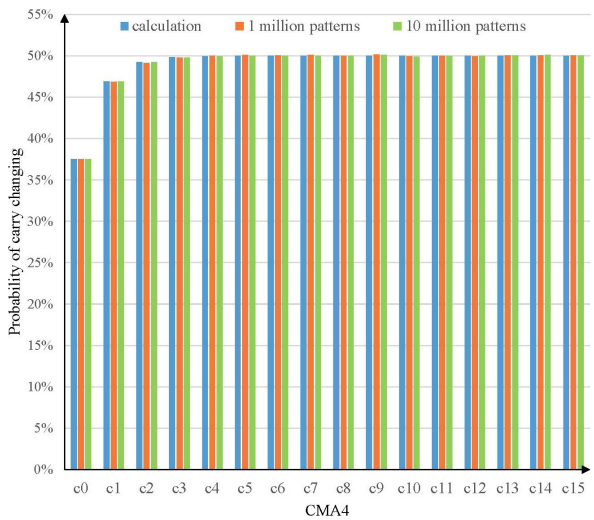

Fig. 11 Probabilities of carry changing for CMA4

Two simulations with 1 million and 10 million random patterns are performed, and the probabilities of carry changing for the CMA4 from calculating (blue bars) and from using 1 million random patterns (orange bars) and 10 million random patterns (green bars) are shown in Fig. 11. The x-axis is the carry at each bit position in CMA4 and the y-axis is the probability of carry changing. As can be seen, the heights of three bars in the carry at each bit position are nearly the same. This demonstrates that both 1 million and 10 million random patterns can be used for power analy-

sis. Because the VCD file of 10 million random patterns is very large, we use 1 million random patterns to evaluate the power consumption of the full suite of adders.

5.2 Scalability Analysis

A 16-bit CMA with four different configuration settings has been previously evaluated. To clarify the scalability of the CMA, we implement and evaluate a 32-bit CMA and a 32-bit RCA in terms of power, delay, area, and accuracy. Similar to the 16-bit CMA (Fig. 6), the 32-bit CMA also employs a 4-bit adder unit (Fig. 7). Therefore, the 32-bit CMA has eight sub-adder units. Since there is no mask signal for the sub-adder unit at the MSB side (from bit positions 28 and 31), seven carry-mask signals are required. The full adder that is used in the 32-bit RCA is the same as that in the 16-bit RCA, as described in Sect. 4.1. The 32-bit CMA with eight configuration settings is summarized in Table 6. Since the critical path delays of the 32-bit adders are larger than those of the evaluated 16-bit adders, we conduct a synthesis and power evaluation for the 32-bit adders at a frequency of 0.25 GHz to obtain results with no timing violation.

Table 7 shows the accuracy results of the 32-bit CMAs with eight configuration settings. Except for the "0.00" in each column in the CMA32_0, all other 0.00s (the gray ones) indicate that the values are very close to 0.00 in the table. As expected, the values of NMED, MRED, and ER become larger from CMA32_0 to CMA32_28. The accuracy of the CMA32_28 is the worst; however, the values of its NMED and MRED are only 1.56% and 2.03%, respectively. While the values of the ER are larger than 90%, except for CMA32_0 and CMA32_4, the NMED and MRED are small. This is not a surprising result, because the masked bit positions are approximated and thus on average the ED is larger in the cases of wide masks than in narrow masks.

Figure 12 shows the results of power consumption and critical path delay. The x-axis denotes the 32-bit CMA with different configuration settings, and RCA32. The y-axis at the left side plots the power consumption and the opposite axis on the right side plots the delay. The blue bar and orange line represent the power consumption and delay, respectively. The values of both power and delay of the 32-bit CMA clearly become smaller from CMA32_0 to CMA32_28. Similar to the 16-bit CMA in Sect. 4.2, the linearities of the power consumption and critical path delay can be also seen in Fig. 12. These linearities benefit from the masking carry propagation of each 4-bit sub-adder unit. This demonstrates that the CMA has good scalability. Compared with the RCA32, the CMA32_28 delivers the lowest power consumption and shortest critical path, and the reductions of power and delay are 62.6% and 86.3%, respectively. In the evaluated 16-bit CMA, the CMA1 delivers the largest reductions of power and delay, and the reductions are 54.1% and 72.5%, respectively. Both reductions of the 32-bit CMA are larger than those of the 16-bit CMA. This indicates that the CMA will deliver larger reductions in power and delay when the bit length of an adder becomes larger. The results

Table 6 32-bit CMA with eight configuration settings.

32-bit CMA	Configuration setting of mask signals	Description
CMA32_0	1111111	Non masked
CMA32_4	1111110	Masked lower 4 bits
CMA32_8	1111100	Masked lower 8 bits
CMA32_12	1111000	Masked lower 12 bits
CMA32_16	1110000	Masked lower 16 bits
CMA32_20	1100000	Masked lower 20 bits
CMA32_24	1000000	Masked lower 24 bits
CMA32_28	0000000	Masked lower 28 bits

Table 7 Accuracy results for the 32-bit CMA.

	NMED (10^{-4})	MRED (10^{-4})	ER (%)
CMA32_0	0.00	0.00	0.00
CMA32_4	0.00	0.00	68.43
CMA32_8	0.00	0.00	90.01
CMA32_12	0.00	0.00	96.83
CMA32_16	0.04	0.05	98.99
CMA32_20	0.61	0.84	99.68
CMA32_24	9.77	13.48	99.86
CMA32_28	156.36	203.07	99.86

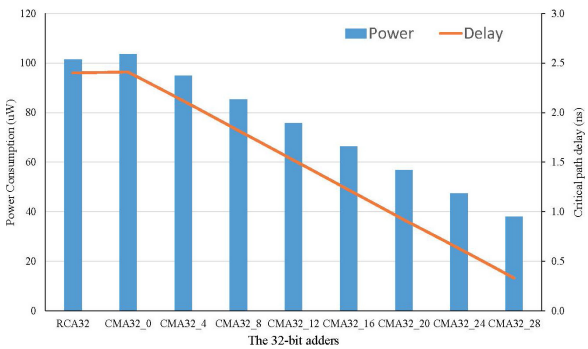

Fig. 12 Power and delay results for the 32-bit adders.

of the design area test of the 32-bit CMA and RCA32 are 168.11 and 159.33 in square microns, respectively. The area overhead is approximately 5.5%.

For some approximate applications that required very large bit length adders (e.g., an adder with 128-bit length or more), the CMA can easily estimate power consumption and critical path delay without comprehensive simulations because of its linearities.

6. Conclusions

This paper proposed an accuracy-configurable approximate adder that does not require any additional logic blocks to achieve accuracy configuration. The experimental results demonstrated that the proposed 16-bit carry-maskable adder can deliver more significant energy savings than the conventional RCA and CLA while maintaining a significantly small circuit area. The experimental results from both the circuit and application levels demonstrate that our proposed adder delivers greater improvements in energy savings, design area, and accuracy than other previously studied adders. The scalability of the proposed CMA is also demonstrated

from the comparisons of the 32-bit adders.

Our ongoing studies will focus on the improvement in performance and accuracy of the CMA. For applications that require high-speed adders, an accuracy-configurable CLA [16] is more desirable than the CMA. Since the CMHA is a basic component of these adders, it can be used to construct the approximate CLA. In contrast, for applications that prioritizes accuracy rather than power, an accuracy-improvement scheme for the CMA [17] is beneficial. Another direction would be to construct accuracy-configurable multipliers [18]. Since adders are basic conponent for multipliers, the approximate multipliers are built by adopting the CMA.

Acknowledgments

This work is supported by JSPS KAKENHI Grant Number JP17K00088 and by the Fukuoka University Internal Research Competitive Funds (Grant No.175007 and 177005). This work is also supported by VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Synopsys, Inc.

References

[1] S. Cotofana, C. Lageweg, and S. Vassiliadis, "Addition related arithmetic operations via controlled transport of charge," IEEE Trans. Comput., vol.54, no.3, pp.243–256, March 2005.
[2] V. Beiu, S. Aunet, J. Nyathi, R.R. Rydberg, and W. Ibrahim, "Serial Addition: Locally Connected Architectures," IEEE Trans. Circuits Syst.-I: Regular papers, vol.54, no.11, pp.2564–2579, Nov. 2007.
[3] S. Venkataramani, V.K. Chippa, S.T. Chakradhar, K. Roy, and A. Raghunathan, "Quality programmable vector processors for approximate computing," 46th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), pp.1–12, Dec. 2013.
[4] A.B. Kahng and S. Kang, "Accuracy-configurable adder for approximate arithmetic designs," IEEE/ACM Design Automation Conference (DAC), pp.820–825, June 2012.
[5] R. Ye, T. Wang, F. Yuan, R. Kumar, and Q. Xu, "On reconfiguration-oriented approximate adder design and its application," IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp.48–54, Nov. 2013.
[6] T. Yang, T. Ukezono, and T. Sato, "A low-power configurable adder for approximate applications," in 19th International Symposium on Quality Electronic design (ISQED), pp.347–352, May 2018.
[7] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-Power digital signal processing using approximate adders," IEEE Transactions on Comptuer-Aided Design of Integrated Circuits and Systems, vol.32, no.1, pp.124–137, Jan. 2013.
[8] 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, Regular Papers, vol.57, no.4, pp.850–862, April 2010.
[9] R. Venkatesan, A. Agarwal, K. Roy, and A. Raghunathan, "MACACO: modeling and analysis of circuits for approximate computing," IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp.667–673, Nov. 2011.
[10] J. Miao, K. He, A. Gerstlauer, and M. Orshansky, "Modeling and Synthesis of Quality-Energy Optimal Approximate Adders," IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp.728–735, Nov. 2012.
[11] K. Du, P. Varman, and K. Mohanram, "High performance reliable variable latency carry select addition," IEEE/ACM Design, Automation Test in Europe (DATE), pp.1257–1262, March 2012.
[12] Silicon Integration Initiative, Inc., NanGate Open Cell Library, https://projects.si2.org/openeda.si2.org/projects/nangatelib, [Accessed on June 25, 2019].
[13] J. Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate and probabilistic adders," IEEE Trans. Comput., vol.62, no.9, pp.1760–1771, Sept. 2013.
[14] M.S.K. Lau, K.-V. Ling, and Y.-C. Chu, "Energy-aware probabilistic multiplier: design and analysis," International Conference on Compliers, architecture, and synthesis for embedded systems, pp.281–290, Oct. 2009.
[15] T. Sato, T. Yang, and T. Ukezono, "Trading accuracy for power with a configurable approximate adder," IEICE Trans. Electron., vol.E102-C, no.4, pp.260–268, April 2019.
[16] T. Yang, T. Ukezono, and T. Sato, "A low-power yet high-speed configurable adder for approximate computing," 51st International Symposium on Circuits and Systems (ISCAS), pp.1–5, May 2018.
[17] T. Ukezono, "Evaluations of CMA with error corrector in image processing circuit," International Journal of Networking and Computing, vol.9, no.2, pp.301–317, July 2019.
[18] H. Baba, T. Yang, M. Inoue, K. Tajima, T. Ukezono, and T. Sato, "A low-power and small-area multiplier for accuracy-scalable approximate computing," IEEE Computer Society Annual Symposium on VLSI (ISVLSI), pp.569–574, July 2018.

Tongxin Yang received his BS degree from Harbin University of Science and Technology, China in 2008 and PhD degree in information and control systems from Fukuoka University, Japan in 2019. He is working with Logic Research Co., Ltd., Japan since 2008. His research interests include approximate computing systems and low power circuits and systems. He is a member of IEEE.

Toshinori Sato received his PhD degree in electronic engineering from Kyoto University in 1999. He is currently a professor of Department of Electronics Engineering and Computer Science at Fukuoka University. His research interests include computer architecture and design methodology. He is a senior member of ACM and IEEE and a member of IPSJ.

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.