

A Low-Power Configurable Adder for Approximate Applications

Tongxin Yang¹ Tomoaki Ukezono² Toshinori Sato³

¹ Graduate School of Information and Control Systems, Fukuoka University, Japan

^{2,3} Department of Electronics Engineering and Computer Science, Fukuoka University, Japan

¹Email: td166502@cis.fukuoka-u.ac.jp ²Email: tukezo@fukuoka-u.ac.jp

³Email: toshinori.sato@computer.org

Abstract

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 power consumption by 54.1% and 57.5% and critical path delay by 72.5% and 54.2%, respectively. In addition, results from an image processing application indicate that the quality of the processed images can be controlled by the proposed adder.

Keywords

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 off 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 preferable to design quality-configurable systems that are able to trade off computation quality according to application requirements [4] [5].

In this paper, we focus on the structure of an accuracy-configurable adder design from the aspect of power consumption. 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 propose an adder in which the generation circuit of each bit of its sum can be dynamically configured to function as a full adder or an OR gate. This configurability is realized by masking carry propagation. We implemented the proposed adder, a conventional ripple carry adder (RCA), and a conventional carry look-ahead adder (CLA) in Verilog HDL using a 45-nm library and evaluated their power consumption, critical path delays, and design areas.

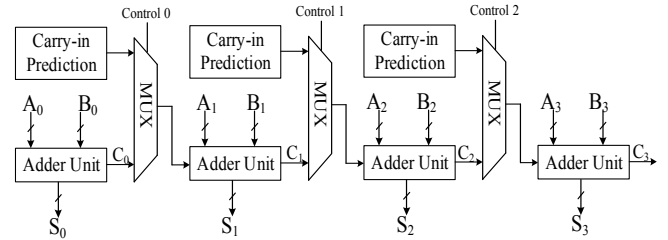


Figure 1: Accuracy gracefully-degrading adder in [5].

Comparisons with the conventional RCA and CLA show that, with a 1.95% mean relative error distance (MRED), the proposed adder reduces power consumption by 54.1% and 57.5%, respectively. We provide a crosswise comparison to demonstrate the superiority of the proposed adder 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 in a real image processing application.

2. Related Work

Gupta et al. [6] discussed how to simplify the complexity of a conventional mirror adder cell at the transistor level. Mahdiani et al. [7] proposed a lower-part-OR adder, which utilizes OR gates for the addition of the lower bits and precise adders for addition of the upper bits. Venkatesan et al. [8] proposed to construct an equivalent untimed circuit that represents the behavior of an approximate circuit. Miao et al. [9] introduced an aligned fixed internal-carry structure and then proposed a dithering approximate adder by trading off error magnitude and error frequency. Du et al. [10] 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 above static approximate designs [6-10] with fixed accuracy may fail to meet application quality requirement or waste power when 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 of different workloads, Kahng et al. [4] proposed an 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 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 to achieve accuracy configurability. To the best of our knowledge, this is the first study that has achieved an accuracy-configurable adder without multiplexers to select approximate and accurate sums. Therefore, no additional circuits, such as carry-in prediction or error recovery logic blocks, are required.

3. Carry-Maskable Adder

A conventional half adder is shown in Fig. 2(a). A 2-input XOR gate is used to generate 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 is 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.

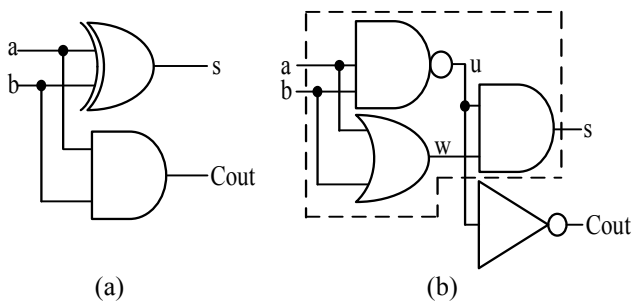


Figure 2: (a) Conventional half adder, and (b) equivalent circuit of a half adder.

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

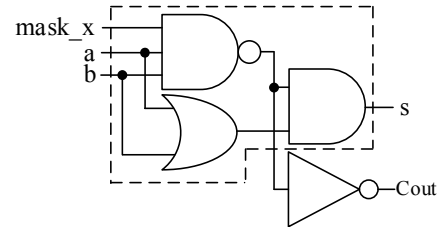


Figure 3: Carry-maskable half adder.

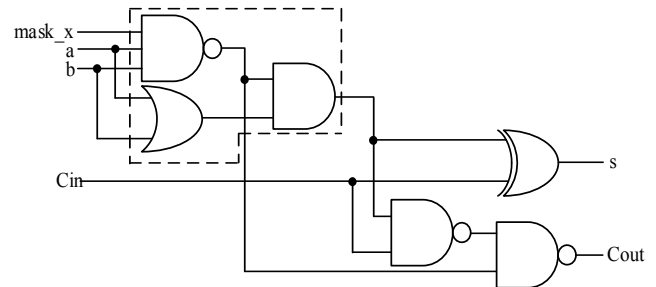


Figure 4: Carry-maskable full adder.

Two inputs: $A = \{a_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0\}$, $B = \{b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0\}$
 Mask signal: $M_X = \{m_{x7}, m_{x6}, m_{x5}, m_{x4}, m_{x3}, m_{x2}, m_{x1}, m_{x0}\}$

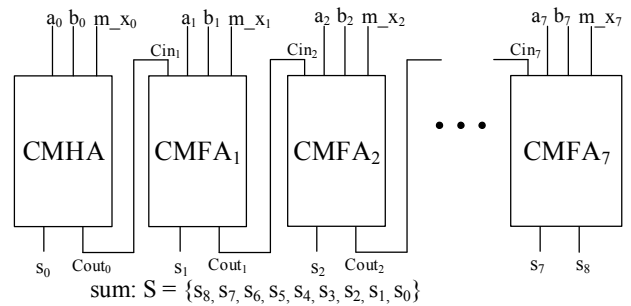


Figure 5: An 8-bit carry-maskable adder.

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

Fig. 5 shows an 8-bit CMA as an example. The carry mask signal M_X comprises eight bits, which are denoted as $m_{x_0}, m_{x_1}, \dots, m_{x_7}$. The left is the least significant bit in Fig. 5. The sum and carry of the CMHA are s_0 and C_{out_0} , respectively. C_{in_1} is connected to C_{out_0} . When m_{x_0} is equal to 0, $s_0 = a_0 \text{ OR } b_0$, and $C_{in_1} = C_{out_0} = 0$. When both m_{x_1} and m_{x_0} are equal to 0, $s_0 = a_0 \text{ OR } b_0$, $C_{in_1} = C_{out_0} = 0$, $s_1 = a_1 \text{ OR } b_1$, and $C_{out_1} = 0$ (C_{in_2} is also 0). In other words, carry propagation from CMHA to CMFA₁ is masked. By expanding the above equations to CMFA₇, when all $m_{x_0}, m_{x_1}, \dots, m_{x_7}$ are 0, all $C_{out_0}, C_{out_1}, \dots, C_{out_7}$ are 0, and $s_0 = a_0 \text{ OR } b_0, s_1 = a_1 \text{ OR } b_1, \dots, s_7 = a_7 \text{ OR } b_7, s_8 = 0$ ($s_8 = C_{out_7}$). 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 C_{in} '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 C_{in_1}, C_{in_2} , and C_{in_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 ($m_{x_0}, m_{x_1}, \dots, m_{x_{15}}$) is required. To simplify the structure of a CMA, we can also group some CMFAs as a sub adder unit. Fig. 6 is 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 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 $S_1 = A_1 \text{ OR } B_1$ (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 CMA, the conventional RCA, CLA and GDA [5]. We implemented a full adder of RCA as with CMFA (Fig. 4), except for the 3-input NAND gate in the dashed frame replacing the 2-input NAND gate in RCA.

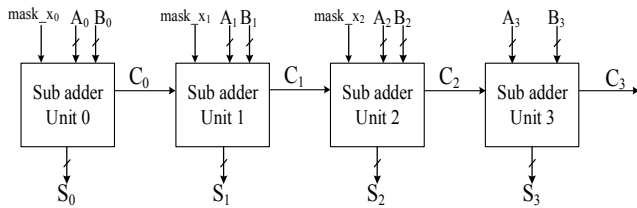


Figure 6: A 16-bit CMA with four sub adder units.

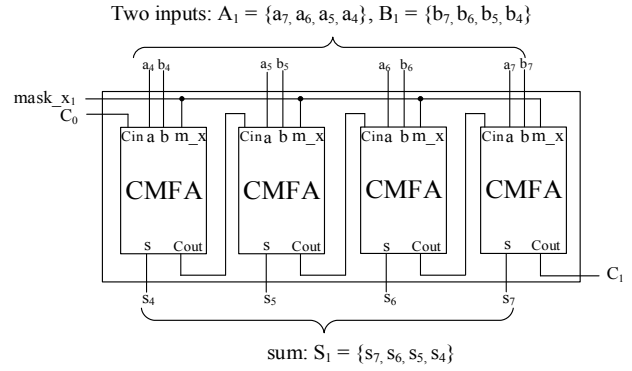


Figure 7: Structure of sub adder unit 1.

All of these adders are 16-bit. 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 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, on the basis of Fig. 6, CMA1 means that sub adder units 0, 1, and 2 are all masked ($mask_{x_{0,1,2}} = 0$), and the accuracy of CMA1 will be the worst among the CMAs. CMA2 means that sub adder units 0 and 1 are masked ($mask_{x_{0,1}} = 0$), but sub adder unit 2 is unmasked ($mask_{x_2} = 1$). CMA4 means that sub adder units 0, 1, and 2 are all unmasked ($mask_{x_{0,1,2}} = 1$). With these settings, accurate results are obtained.

The adders were coded using Verilog HDL. The Synopsys VCS was used to simulate the designs and generate value change dump (VCD) files to evaluate the power consumption precisely. The Synopsys Design Compiler was used to synthesize the adders with the NanGate 45nm Open Cell Library [11]. The power consumption was evaluated at a frequency of 0.5GHz. The operating conditions for synthesis employed typical conditions (a 1.00 process factor, 1.1V power supply, and 25°C operating temperature). All designs were synthesized and optimized with default compile options. The 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 numerical outputs of all of the adders with one million randomly generated input patterns.

4.2. Accuracy Results

The ED and MED are proposed for the evaluation of the performance of approximate arithmetic circuits [12]. ED is defined as the arithmetic difference between the accurate sum (S) and the approximate sum (S'): $ED = |S - S'|$. MED is the average of EDs for a set of outputs. 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., MED, MRED, and ER) are used to evaluate the adders.

Table 2 compares the accuracy of the results and shows that the accuracy of both CMA and GDA changed widely according to the settings of the configuration. Both the MED and MRED of CMA are smaller than those of GDA at each setting. As expected, there are no errors in CMA4 and GDA4. Although the ER value of CMA is larger than that of GDA in each accuracy configuration setting, the MED and MRED of CMA are about 50% of GDA.

4.3. Power, Delay and Area Results

Comparisons of the power consumption and critical path delay for the different adders are shown in Fig. 8 and Fig. 9. The x-axes denote the adders with different configuration settings of 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 power consumption of CMA1 is the smallest among the adders. Compared with RCA and CLA, CMA1 delivers 54.1%, and 57.5% of power consumption reductions, respectively. Owing to the carry-maskable structure of CMA, power consumption increases in a linear manner in the order of CMA1 to CMA4. The power consumption of CMA4 is slightly larger than that of RCA. Remember that our proposed CMA is an accuracy-configurable adder and CMA4 delivers an accurate result. Compared with another accuracy-configurable adder GDA with the same configuration settings, the power consumption of GDA4 is 1.8 times larger than that of CMA4. Furthermore, the power consumption of GDA1 is 3.8 times larger than that of CMA1.

Fig. 9 demonstrates that the delay of CMA1 is the smallest among the adders. The linearity of delay can also be found in the order of CMA1 to CMA4, with the delay of CMA4's being the largest among the adders. As can be seen, just as the delay of CMA4 is close to that of RCA, the delay of GDA4 is close to that of CLA, demonstrating that the accuracy configurability of CMA is based on the structure of RCA and that of GDA is based on the structure of CLA. The delay of CMA4 is slightly larger than that of RCA. The critical path of an adder is the carry propagation path, and the critical paths of both adders are from the inputs at bit position 1 (a_1, b_1) to the sum at bit position 15 (s_{15}). The internal delay of CMFA at bit position 1 in CMA4 is slightly larger than that of the full adder at bit position 1 in RCA because CMFA is implemented using a 3-input NAND gate and RCA is implemented using a 2-input NAND gate. Note that the other full adders and CMFAs from bit positions 2 to 15 do not have any effect on the carry propagation. Thus, the critical paths of the two adders are the same from bit positions 2 to 15.

The power-delay product (PDP) is proposed to evaluate approximate arithmetic circuits [2]. The results of PDP for CMA and GDA are shown in Fig. 10 and Fig. 11 for a better overview of the circuit characteristics. The circles and triangles represent CMA and GDA, respectively. Smaller

values represent better results in energy savings. CMA1 delivers the best results. Compared to CLA and RCA, CMA1 delivers 80.5% and 87.4% PDP reduction, respectively. Compared to the GDAs, the PDP of GDA1 is 4.4 times larger than that of CMA1, and the PDP of GDA4 is 1.2% larger than that of CMA4. The PDP of CMA4 is 3% larger than that of RCA. Fig. 11 performs a comparison of PDP results relative to MRED in order to clarify the contributions to the power saving and accuracy of the proposed adder. As can be seen, CMAs with all of the different configuration settings are at the bottom left of Fig. 11. It means 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. Fig. 10 and Fig. 11 demonstrate that our proposed CMA definitely achieves good results in energy savings.

Table 2: Accuracy Comparison.

	MED	MRED (10^{-4})	ER (%)
CMA1	1012.62	195.14	95.95
CMA2	58.38	10.38	88.36
CMA3	3.72	0.79	68.15
CMA4	0.00	0.00	0.00
GDA1	2022.95	388.69	83.08
GDA2	116.81	20.53	5.58
GDA3	6.59	1.62	0.16
GDA4	0.00	0.00	0.00

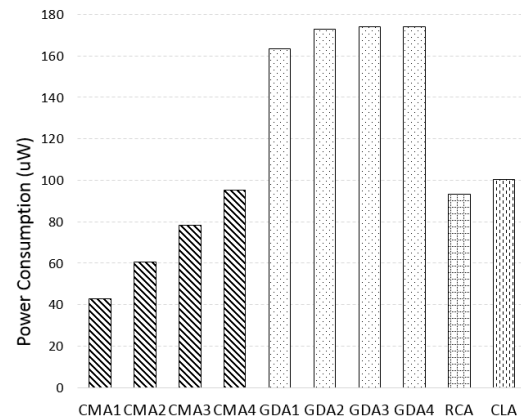


Figure 8: Power consumption results.

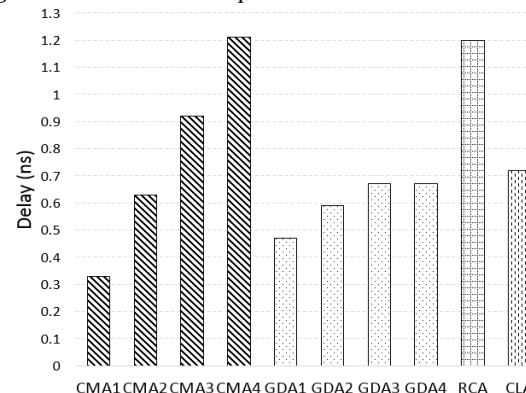


Figure 9: Critical path delay results.

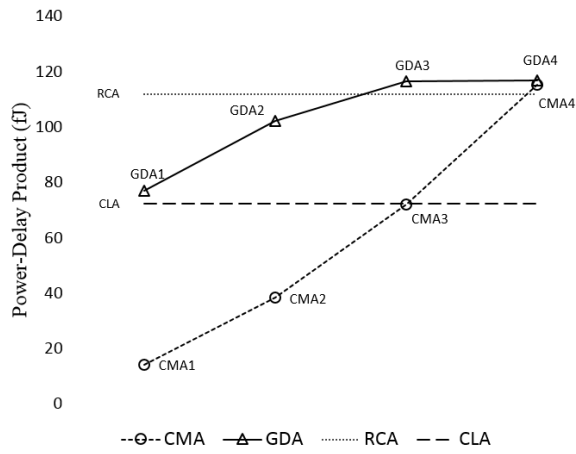


Figure 10: PDP results.

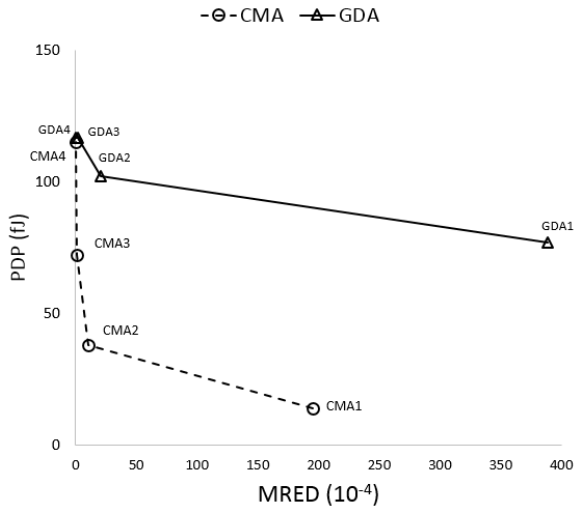


Figure 11: PDP results relative to MRED.

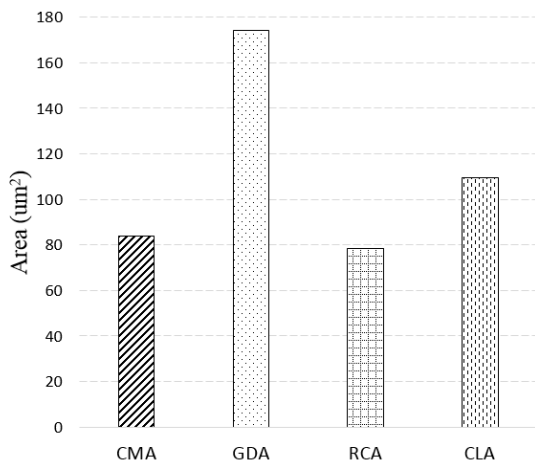


Figure 12: Area results.

A comparison of the design area results is shown in Fig. 12. Note that the accuracy configuration setting does not have any effect on the design areas of CMA and GDA. Although CMA is slightly larger than RCA, its area is 76.5% of CLA and 48.1% of GDA. As expected, the design area of RCA is the smallest among the adders.



(a) Original (b) accurate



(c) CMA1 (d) CMA2 (e) CMA3



(f) GDA1 (g) GDA2 (h) GDA3

Figure 13: Images processed by sharpening algorithm (a), (b), (c), (d), (e), (f), (g), and (h).

5. Image Processing

In this section, an image processing application of the proposed adder was also evaluated. An image sharpening algorithm [13], which 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 adders, whereas all of the other operations (multiplication, subtraction, and division) were accurate.

Fig. 13 shows the images processed by image sharpening algorithm. To achieve a clear comparison, the images processed by CMA and GDA at the same configuration setting are placed in the same column. Fig. 13 (a) is the original image (No. 1) and the accurately processed image is Fig. 13 (b). The images processed by CMA1, CMA2, CMA3, GDA1, GDA2, and GDA3 are Figs. 13 (c), (d), (e), (f), (g), and (h), respectively. As can be seen, the images processed by CMA3 (e) and GDA3 (h) are visually indistinguishable images from the accurately processed image (b). Obviously, the image processed by CMA2 (d) is sharper than that processed by GDA2 (g). The difference between the images processed by CMA2 (d) and CMA3 (e) is imperceptible because the bit-lengths of the sub adder units in GDA and CMA were set to four bits. For more controllability of accuracy (quality) to achieve smaller difference in the accuracy between two different configuration settings, the bit-lengths of the sub adder units can be reduced.

Similar to [5], the processed image quality was measured using the peak signal-to-noise ratio (PSNR), which is usually used to measure the quality of reconstructive processes that involve information loss.

Table 3: PSNR results of 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

Table 3 shows the PSNR results of CMA and GDA in dB. Larger values represent better quality images. Excepted for the results of image No. 2 for CMA2 and GDA2, all of the PSNR values of CMA are larger than those of GDA at the same configuration settings, demonstrating that our proposed CMA delivers better quality images than GDA. CMA4 and GDA4 are accurate, with no PSNR results.

6. Conclusion

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

Our ongoing work seeks to implement accuracy-configurable designs for arithmetic components from the aspect of power consumption. The achievement of the selectivity from energy and performance at runtime of accuracy-configurable systems and applications is an interesting avenue for future exploration.

7. Acknowledgment

This work was supported by JSPS KAKENHI Grant Number JP17K00088 and by funds (No.175007 and No.177005) from the Central Research Institute of Fukuoka University. This work is supported by VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Synopsys, Inc.

8. References

- [1] S. Cotofana, C. Lageweg, and S. Vassiliadis, "Addition related arithmetic operations via controlled transport of charge", *IEEE Transactions on Computers*, vol. 54, no. 3, pp. 243-256, Mar. 2005.
- [2] V. Beiu, S. Aunet, J. Nyathi, R. R. Rydberg, and W. Ibrahim, "Serial Addition: Locally Connected Architectures", *IEEE Transactions on Circuits and Systems-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, Jun. 2010.
- [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] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-Power digital signal processing using approximate adders", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 1, pp. 124-137, Jan. 2013.
- [7] 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 Transactions on Circuits and Systems I: Regular papers*, vol. 57, no. 4, pp. 850-862, Apr. 2010.
- [8] 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.
- [9] J. Miao, K. He, A. Gerstlauer, and M. Orshansky, "Modeling and Synthesis of Quality-Energy Optimal for Approximate Adder", *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 728-735, Nov. 2012.
- [10] 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, Mar. 2012.
- [11] NanGate, Inc. NanGate FreePDK45 Open Cell Library, http://www.nangate.com/?page_id=2325, 2008
- [12] J. Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate and probabilistic adders", *IEEE Transactions on computers*, vol. 62, no. 9, pp. 1760-1771, Sep. 2013.
- [13] M. S. Lau, K. V. Ling, and Y. C. Chu, "Energy-Aware probabilistic multiplier: Design and Analysis", *International Conference on Compilers, architecture, and synthesis for embedded systems*, pp. 281-290, Oct. 2009.