International Journal of Networking and Computing – www.ijnc.org, ISSN 2185-2847
Volume 9, Number 2, pages 301–317, July 2019

Evaluations of CMA with Error Corrector in Image Processing Circuit

Tomoaki Ukezono
Department of Electronics Engineering and Computer Science
Fukuoka University
Fukuoka City, Japan
tukezo@fukuoka-u.ac.jp

Received: February 15, 2019
Revised: May 6, 2019
Accepted: June 11, 2019
Communicated by Takeshi Ohkawa

Abstract

To reduce power consumption, approximate computing is an efficient approach for error-tolerant applications such as image processing. Approximate arithmetic adders can be used for the approximate computing, and can trade off accuracy for power. CMA, a dynamically accuracy-configurable approximate adder, had been proposed. CMA can sharply reduce power consumption compared with other accuracy-configurable approximate adders, while allowing it to change accuracy-setting at run-time. In this paper, we evaluate CMA with error corrector that needs only two gates for each digit in actual image processing circuit. By increasing slight extra power, the proposed value corrector can improve PSNR quality of output images by up to 73.71%.

Keywords: Approximate Computing, Arithmetic Adder, Error Corrector, Low-Power

1 Introduction

There are cases in which computational resources that can control significance of miscalculation are used to computer design. The computer design can be applied for error-tolerant applications such as image processing. The reason is that because it is difficult to perceive the small arithmetic error in each pixel data by human eyes. The inaccurate data computing is called as approximate computing. Approximate circuits such as adders, multipliers and other logical circuits are one of important components constructing approximate computing since the approximate circuits can reduce hardware size, delay and power consumption compared with accurate circuits.

This paper focuses on arithmetic adder circuits especially in approximate circuits. There are several previous works for approximate adder circuits [8][5][3]. In the previous works, accuracy that is assumed by the approximation cannot be changed after the circuit design since they are required for error acceptable systems or applications with a configuration for desired accuracy. Carry Maskable Adder (CMA) that can be dynamically switched accuracy of approximation had been proposed[9]. CMAs can dynamically select optimal balance point between energy consumption and accuracy at run-time based on accuracy requirements on each error-tolerant application.

The CMA has two factors of approximate calculations. One is that CMAs avoid generating and calculating carries. Therefore, value related to carries is lost from the result of result. The other is that CMAs calculate approximate result of addition regardless of the carries by using logical ORs instead of logical XORs. The truth tables that are given by two inputs ORs and XORs are deferent in only one row where signal value 1 is assigned to both inputs. The similarity can be exploited to approximate addition. The OR Adder is already used in the previous work[8].

Ignoring carries means that the approximate result of addition leads to under estimation. On the other hand, logical ORs produce over estimated result compared with logical XORs. For example, when considering the calculation of $1+1$, if only ignoring the carry to the higher digit of $(10)_2$ that is obtained as correct calculation result, the approximate calculation result will be $(00)_2$, and the approximation error is 2 in the negative direction. CMA has a characteristic that can reduce the approximation error to 1 in the negative direction by not only preventing the carry but also setting the approximation result of $1+1$ to $(01)_2$ by using logical OR. The two conflicting direction for the estimation might act to compensate with each other, and have preferable influence upon error distance for result of addition.

The each factor of approximate calculations contributes reducing energy consumption. Since the circuits related to the carries are inactivated, and XOR gates related to result of addition are partially inactivated by that the XOR gates behave OR gates in CMAs. Details of the low power techniques by CMA will be explained in the Section 2.

This paper propose an error corrector for CMA. The error corrector can be implemented by minor modification for conventional CMA. Approximate error from CMA is corrected by biasing output result. The bias is defined based on statistical analysis for error distances of pre-designed approximate adder. Proposed error corrector requires only two gates for implementation for each digit. The additional circuits will no consume much power. Therefore, performance of low-power consumption for CMA cannot be degraded by proposed mechanism.

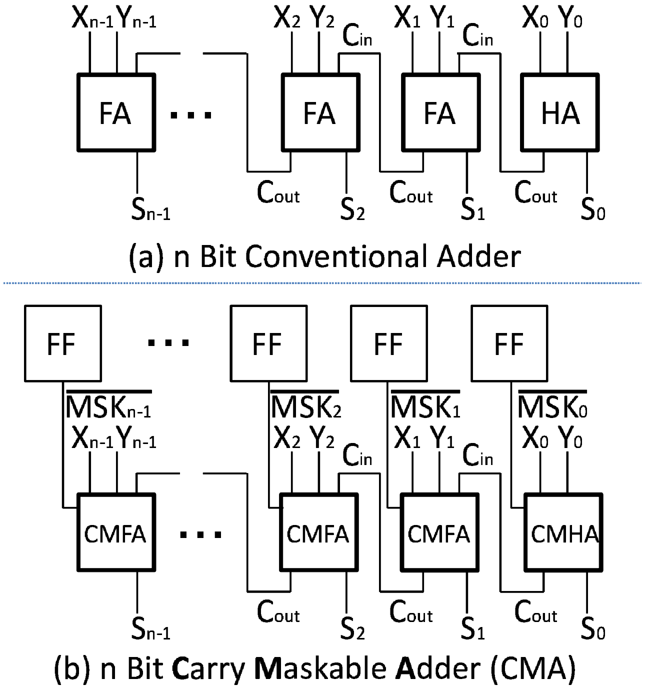
2 CMA:Carry Maskable Adder

2.1 Overview of CMA

Figure 1(a) shows block diagrams for n bit conventional adders. X, Y and C_{in} are inputted as augend, addend and carry-in respectively. The result of addition and carry-out are outputted to S and C_{out} respectively. We assume that C_{out} for each full adder (FA) is C_{in} of the succeeding next significant FA. The adder is called as the ripple carry adder (RCA) generally. In addition, in Figure 1(a), half adder (HA) is used for the least significant bit of the adder instead of FA since the carry-in from lower digit assumes zero. Figure 1(b) shows block diagram of inside CMAs. CMAs wire the carries as with RCAs. There are differences compared conventional adders with CMAs. FAs and HAs are replaced with carry maskable full adders (CMFA) and carry maskable half adders (CMHA) respectively. The details of internal logic for CMFA and CMHA are described in next subsection.

CMFA and CMHA have additional input that is represented as $\overline{MSK}$ with in the figure. $\overline{MSK}$ is used to control accuracy of approximation, inputted from storages such as a flip flop (FF) that can be updated by system users at run-time. If $\overline{MSK}$ is negated, corresponding CMFA or CMHA is immediately switched to approximate calculation. Otherwise, CMFA and CMHA calculate accurate result of addition same as FA and HA. The storages are only used for not calculation but controlling accuracy. Therefore the storages are ignored by evaluation in this paper.

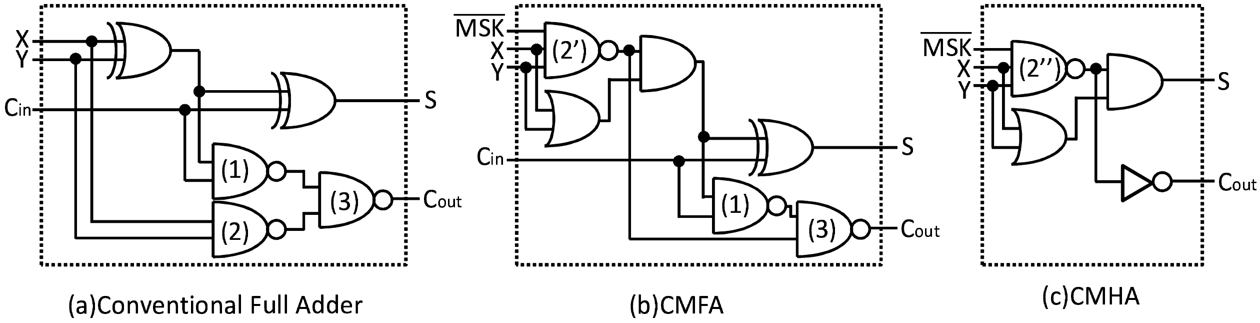
Stepwise accuracy can be defined by considering a bit pattern for $\overline{MSK}$s. A value included all digits is less most affected by incorrect value for the lowest one digit of them. Oppositely, a value included all digits is the most affected by uncertain value for the highest one digit of them. Therefore, for example, if a four bits CMA is designed, we can define the stepwise accuracy in five steps by extracting effective bit patterns from the complete accuracy to the lowest accuracy. The bit patterns are {1,1,1,1}, {1,1,1,0}, {1,1,0,0}, {1,0,0,0} and {0,0,0,0}, stepwisely degrade the accuracy in the listed order.



(a) n Bit Conventional Adder

(b) n Bit **C**arry **M**askable **A**dder (CMA)

Figure 1: Conventional Adders and CMAs.



(a)Conventional Full Adder

(b)CMFA

(c)CMHA

Figure 2: Gate Leve Diagram of Carry Maskable Full Adder (CMFA) and Carry Maskable Half Adder (CMHA)

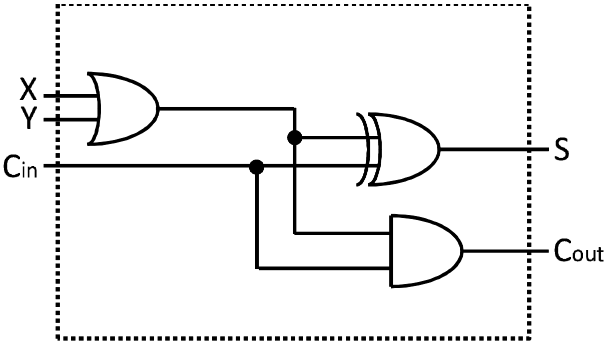


Figure 3: Equivalent circuit where $\overline{MSK} = 0$ in CMFA.

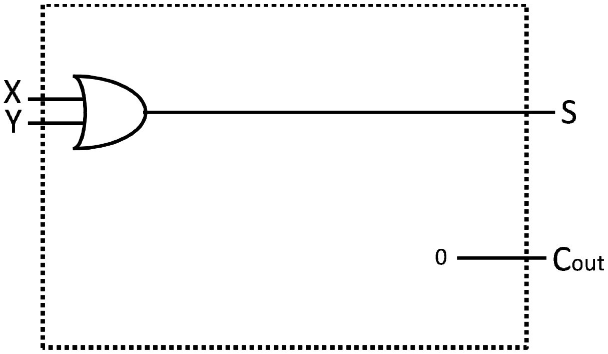


Figure 4: Equivalent circuit where $\overline{MSK} = 0, C_in = 0$ in CMFA.

2.2 CMFA and CMHA

Figure 2 shows gate level diagrams for conventional full adder, CMFA, and CMHA. Figure 2(a) shows conventional full adders. In general, full adders have two half adders and an OR gate, and the half adder have a XOR gate and an AND gate. Consequently, a full adder has two XOR gates, two AND gates and an OR gate. The two XOR gates are required to generate result of addition. The two AND gates and an OR gate are required to generate carry-out. In Figure 2(a), conventional full adders have two XOR gates and three NAND gates. The three NAND gates are numbered form (1) to (3). The three NAND gates behave as the same function by two AND gates and an OR gates described above.

Figure 2(b) shows CMFA. XOR gates can be assembled as composite gates using the other logic gates such as AND, OR and NOT gate. The left side XOR gate in Figure 2(a) is represented as an AND gate, an OR gate and a NAND gate in Figure 2(b). In addition, the NAND gate that is numbered by (2) can be eliminated in Figure 2(b) since the NAND gate (2) can be unified as the NAND gate (2').

Figure 3 shows the equivalent circuit where signal value 0 is assigned to $\overline{MSK}$ in CMFA. The NAND gate (2') outputs signal value 1 where at least one or more signal value 0 is inputted to it. By fixing the other side input of the next AND gate to signal value 1, the output of the OR gate connected in parallel with the NAND gate (2') passes through the next AND gate to the input of the next XOR gate with no change. Therefore, if $\overline{MSK}$ is assigned by signal value 0, the left side composite XOR gates will behave as an OR gate.

Figure 4 shows the equivalent circuit where signal value 0 is assigned to $\overline{MSK}$ and Cin in CMFA. If the signal value 0 is assigned to Cin in Figure 3, the output of the OR gate passes through the right side XOR gate to the S with no change by fixing the other side input of the XOR gate to signal value 0. In addition, Cout is fixed to signal value 0 since one or more signal value 0 is inputted to the AND gate.

They assume the behavior that is shown by Figure 4 where approximation is validated in CMFA at run-time. If the Cin from least significant bit of the adders is reliably determined to signal value 0, the Cout of CMFA that are assigned to second or later bit of the adder are fixed to signal value 0.

Figure 2(c) shows CMHA that can be assigned to least significant bit of CMA. The conventional half adder is constructed with a XOR gate and an AND gate. CMHA have composite XOR gates with $\overline{MSK}$ same as CMFAs in Figure 2(b). If signal value 0 is assigned to $\overline{MSK}$, CMHA produce approximate result of addition by logical OR. CMHA has a NOT gate instead of an AND gate since the inverted carry-out signal is already produced by the NAND gate represented as (2") inside the composite XOR gates. Furthermore, the Cout will be fixed to signal value 0, if $\overline{MSK}$ is assigned to signal value 0, since the NAND gate (2") outputs signal value 1 regardless of X and Y. Finally, where $\overline{MSK}$ is assigned to signal value 0, CMHA works exactly like equivalent circuit of CMFAs shown by Figure 4.

2.3 Reducing Power and Delay by CMA

The most important contribution of CMA is shown by Figure 4. If signal value 0 is assigned to $\overline{MSK}$ and Cin in CMFA or CMHA, only an OR gate is validated. Therefore, CMFA or CMHA can reduce dynamic energy for carry propagation by changing input values of X and Y. In other words, during approximate calculation, t transistors that are mapped to the carry-chain cannot be switched since all input parameters of the gates are fixed.

In addition, CMA can reduce delay beyond critical path delay. If signal value 0 is assigned to $\overline{MSK}$ and Cin in CMFA or CMHA, the switching delay is accumulated only on the unmasked carry chain path. Therefore, the delay of CMA can be determined by the number of adjacent CMFA or CMHA that are configured as accurate.

3 Proposed Error Correction

3.1 Statistical Analysys for Error by CMA

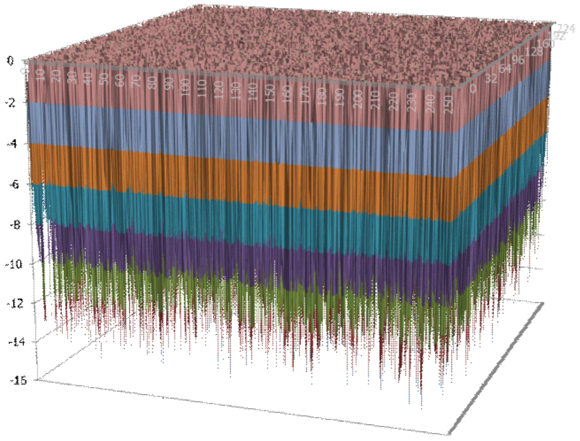


Figure 5: Results of Approximation for Lower 4 bits masked 8 bit CMA.

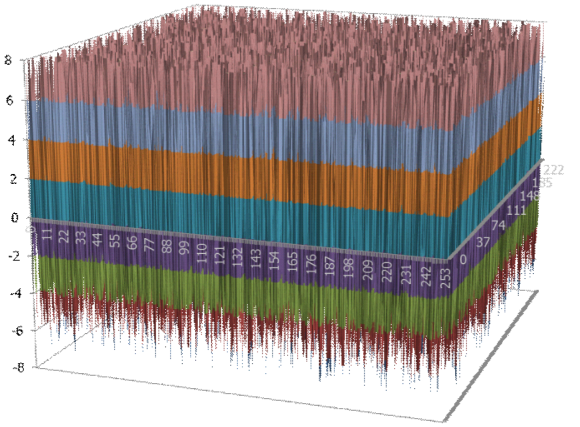


Figure 6: Results of Approximation for Lower 4 bits masked 8 bit CMA with Proposed Error Corrector(EC).

In this section, we statistically analyze output value of CMA, show the aim of our proposed error corrector and how to correct error for CMAs. Figure 5 shows results of approximation for lower 4 bits masked 8 bit CMA. X (from 0 to 255) and depth Z (from 0 to 255) axis means addend and augend, Y axis (from 0 to -16) means error. CMAs calculate $1 + 1 = 1$ as approximate addition using logical OR. Moreover, carry propagation is prevented in the range of a masked bit field. Therefore, the error direction of CMAs is always negative. In the figure, the maximum error distance is represented as -16, because the carry-out signal of fourth CMFA which means the value of 2^4 is discarded by its mask input. When the addend or augend is repeatedly incremented by one, the error will increase in order until the carry-out to fifth digit is required. After that, the error is rapidly decreased. From this reason, Figure 5 shapes like a flower flog pin that turns upside down.

We can obtain two important results from the analysis. One is that the direction of error for

CMA is always negative. Therefore, value that is used for correction must be positive value. If the positive value is added to output of CMA, the error distances might be decreased. Two is that spikes, the maximum error, can be found regularly in the figure. This means that discarding carry-out signal in boundary between masked bits and unmasked bits is dominant for maximum error distance. Therefore, we can detect the maximum error distance only focusing on the boundary of mask bit field.

3.2 Methodology of Proposed Error Correction in CMA

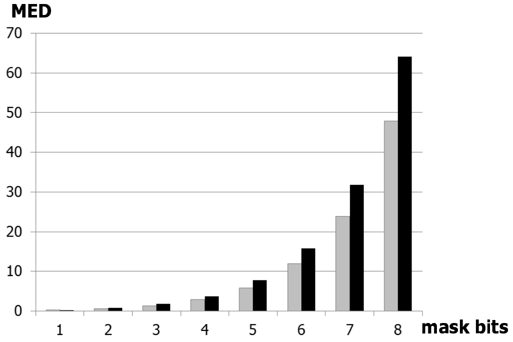


Figure 7: MED of proposed error corrector compared with conventional 8 bit CMA.

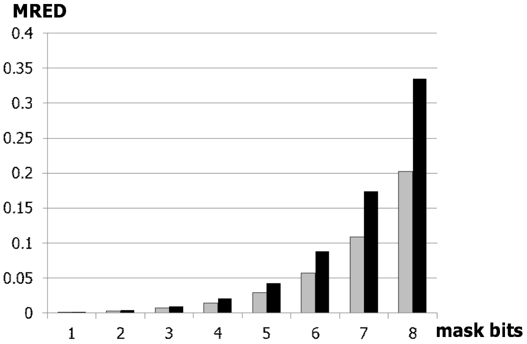


Figure 8: MRED of proposed error corrector compared with conventional 8 bit CMA.

Figure 6 shows results of approximation for lower 4 bits masked 8 bit CMA with proposed error corrector. In this paper, the error corrector is called as EC in convenience. X, Y and Z axis in Figure 6 is the same as Figure 5. The proposed EC corrects error using positive value. Therefore, maximum error distance can be decreased. In the figure, maximum error distance is changed from 16 to 8. Our proposed EC shifts error range. In the figure 6, error range is from 0 to -16. In the figure 8, error range is from 8 to -8. Using the error range of positive direction, proposed EC can improve accuracy of CMA.

For example, the CMA with EC of 8-bit mask and CMA of 7-bit mask have similar accuracy since the maximum error distance is the same. However, in CMA with 7-bit mask, carry propagation from 7th bit to 8th bit occur. On the other hand, CMA with EC ignores the carry propagation. Therefore, CMA consume more power in the 8th bit position than CMA with EC.

In our EC, the carry-out signal is used for the correction. Where the error exceeds the half of maximum error distance (2^3), if the carry-out signal across the mask boundary of CMA forcedly be

1, corrected maximum error distance will be decreased to less than half since the correcting value of 2^4 is added to result of original approximate addition. To simplify the EC circuit, our EC only detect a partial condition that both addend and augend that is pointed by the highest bit of mask bits are 1. That is because if the large circuit is required for the detection, the most important advantage of CMA, low-power, will spoil.

In this section, statistical accuracy of CMA with EC is evaluated to validate our proposal. MED and MRED [2] are widely used for evaluation for approximate arithmetic circuits.

Error distance (ED) is defined as the difference between an accurate sum (M) and its approximate sum (M'), i.e., $ED = |M' - M|$. MED is the average of EDs. Relative ED (RED) is defined as the ED divided by M, i.e., $RED = ED/M = |M' - M|/M$. Mean RED (MRED) is the average of REDs.

Figure 7 and 8 show MED and MRED respectively. MED and MRED is calculated by assuming all patterns for inputs are assigned to addend and augend. Therefore, the number of tests is $2^8 \times 2^8$.The black bar means conventional CMA. The gray bar means CMA with proposed EC. X axis means mask bits. For example, 4 of X axis mean that the lower 4 bits in CMA is masked. In all mask setting, accuracy is improved for both criteria. These result shows that our EC can contribute improving accuracy for CMA regardless of accuracy configuration for conventional CMA. In our experimental result 16 bit CMA with EC is evaluated. The bit-scalability of our proposal will show in Section 6.

4 Circuit Desgin for Proposed Error Corrector

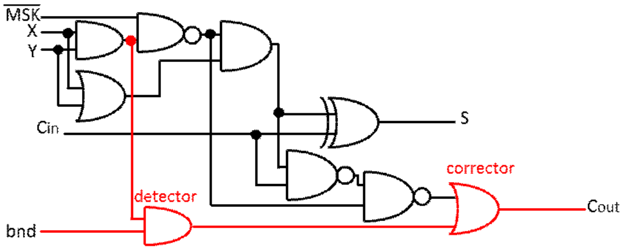


Figure 9: CMFA with proposed error corrector (CMFA with EC).

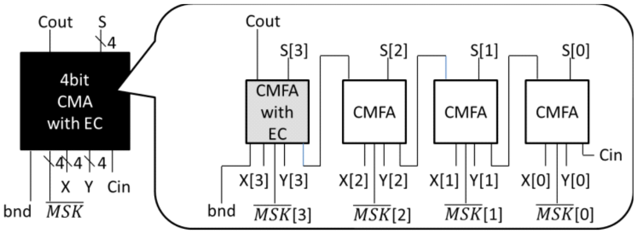


Figure 10: 4 bit sub-adder with proposed error corrector (4 bit CMA with EC).

This section shows circuit design for proposed error corrector. Figure 9 shows CMFA with EC that proposed error correction in Section 3 is embedded into CMFA. Two gates, a detector and a corrector, are added to conventional CMFA logic. The input signal bnd is used to inform where is the boundary position. If the bnd is input as 1 the CMFA is configured as the highest bit of mask bits. The corrector outputs forced carry-out signal when the bnd and X, Y are input as 1. The two gates consume extra dynamic energy by increasing the total amount of switching activities. Moreover,

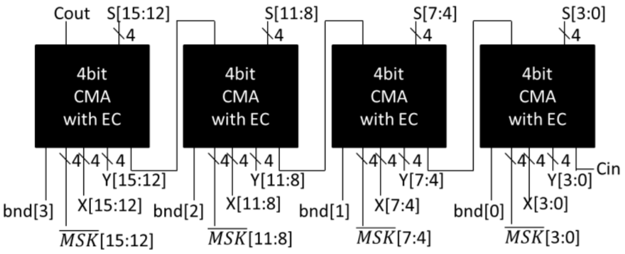


Figure 11: 16 bit CMA with EC.

unmasked CMFA might consume extra dynamic energy due to the forced carry signal. This paper will evaluate power consumption for proposed EC, and carefully discuss trade-off between accuracy and power reduction in section 6 since the main contributions of CMA is power reduction.

Figure 10 shows 4 bit CMA with EC. The CMFA with EC is only located in the highest bit of the 4bit CMA. In our evaluations, 16 bit CMA with EC is designed. Therefore, the 4 bit CMA with EC is treated as a sub-adder, placed into the adder design as four instances.

Figure 11 shows 16 bit CMA with EC. The four 4 bit CMA with ECs are placed and connected each other like a ripple carry adder. In 16 bit CMA with EC, granuarity of accuracy configuration is 4 bitwise. In other words, four level accuracy can be can be configured. Therefore, $\overline{MSK}$ bits in a 4 bit CMA with EC must be {0,0,0,0} or {1,1,1,1}. In addition, the bnd input that is connected with the highest level of masked 4 bit CMA with ECs must be 1.

5 Experimental Setup

5.1 Experimental Setup for adder units

To evaluate power consumption, conventional 16 bit CMA and proposed CMA with EC are designed by using Verilog HDL. Synopsys Design Compiler and NanGate 45nm Open Cell Library [1] are used for logic synthesis. The default compiler option is used. Area and delay required to ASIC implementation can be obtained from the report of Design Compiler. The designed RTL and test bench are used to generate the value change dump (VCD) files. The RTL and test bench is run on Synopsys VCS simulator. The switching activity interchange format (SAIF) file that is converted from the VCD file is used as input to Synopsys Power Compiler. Dynamic power consumption for designed circuits can be obtained from the report of Synopsys Power Compiler. In the VCS simulation, input signals for target circuits are changed by 0.5 GHz frequency. Using one million set of input signal that is generated by random function in VCS, switching activity for target circuits are simulated. Static (leakage) power consumption for designed circuits is also obtained the report of Synopsys Power Compiler.

MED and MRED are calculated same as calculation method for Figure 7 and 8. Therefore, in this case, the number of tests is $2^{16} \times 2^{16}$. Accuracy evaluation is evaluated by using C++ program.

5.2 Experimental Setup for actual image processing circuits

$$G = \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix} \tag{1}$$

$$R(x,y) = \frac{1}{273} \sum_{i=-2}^{2} \sum_{j=-2}^{2} G(i+3, j+3) I(x+i, y+i) \tag{2}$$

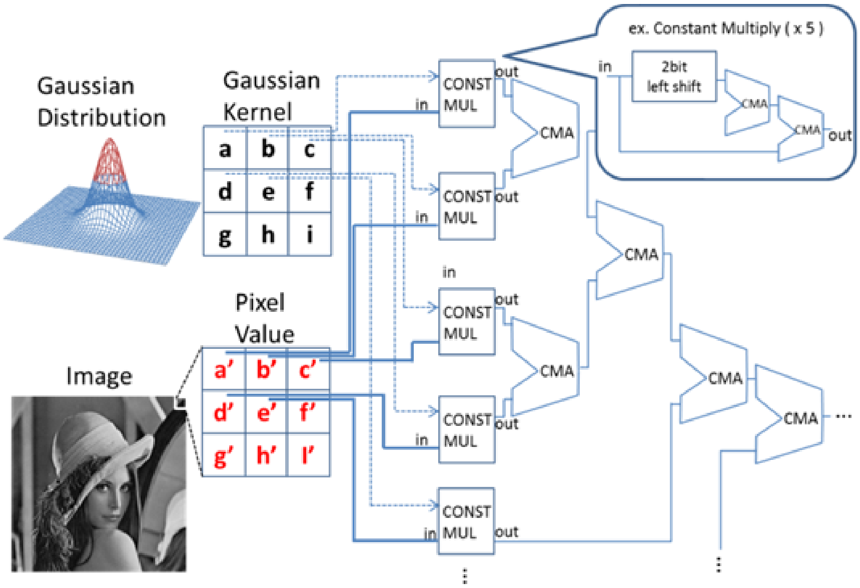


Figure 12: An Example of image sharpening circuit for monochrome images.

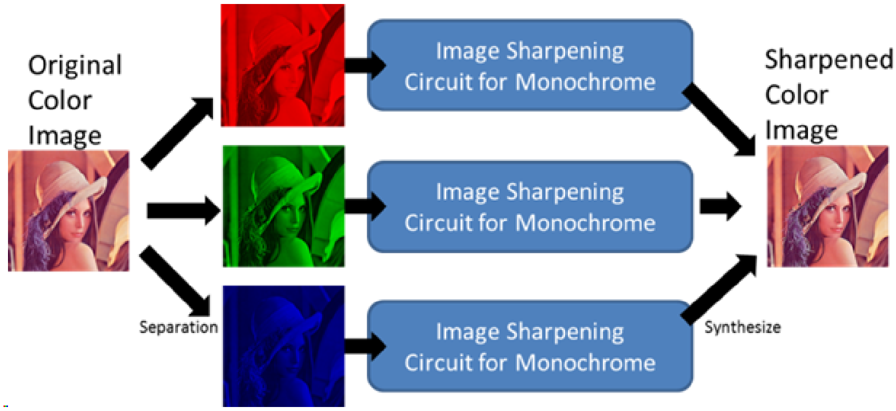

Original Color Image
Separation
Image Sharpening Circuit for Monochrome
Image Sharpening Circuit for Monochrome
Image Sharpening Circuit for Monochrome
Synthesize
Sharpened Color Image

Figure 13: Colorization for image sharpening circuits.

$$S(x,y) = |I(x,y) - R(x,y)| + I(x,y) \tag{3}$$

To evaluate proposed EC in image processing, Image sharpening algorithm [4] is used. Calculations of the algorithm represented as in formulas (1), (2) and (3). G is Gaussian kernel that is approximately represented as discrete values of two dimensional Gaussian distribution. In our evaluation, 5×5 Gaussian kernel is used. I is pixel value of original image, and R is pixel value of smoothed image. Coordinates of the I and R in images are specified by x and y in the formulas. R is calculated as discrete convolution of Gaussian kernel and pixel value for original images, and then is divided by a total sum of elements of Gaussian kernel (273). S is pixel value of sharpened image. S is generated by adding absolute value of the deference between original and smoothed image to original image.

The image sharpening algorithm is implemented as hard-wired logic in our evaluation. Figure 12 shows an example of our implementation of image sharpening algorithm for monochrome images. All calculations of (2) and (3) can be implemented as addition since the multiplier and divider is given as constant value from the formulas. For example, $5 \times a$ can be transformed to $4 \times a + a$. Any power of two multiplications can be implemented as constant left shift which is not required any transistors. That is, only one adder are required to implement calculation of $5 \times a$. In this way, any constant multiplication can be implemented as combination of constant left shift and addition. Therefore, the sharpening algorithm only requires adder unit, whole circuit would be adder unit array. In our implementation, conventional 16 bit CMA or 16 bit CMA with EC is applied to all additions of (2) and (3) as adder unit, a monochrome image sharpening circuit requires 34 adder units.

In our evaluation, we assume that color images are provided to our circuit. However, the sharpening algorithm can process only monochrome images. Figure 13 shows how to colorize the image sharpening circuits. Color images are constructed by information of three primary colors which are red, green, and blue. In the true color representation, 24 bit that are 8 bit for red, 8 bit for green, and 8 bit for blue is required to represent for each pixel. To colorize the image sharpening algorithm, we must process each 8 bit of pixel data as monochrome pixel data. Therefore, in our implementation, three image sharpening circuit are configured in parallel. Naturally, 102 adder units are required to implement color image sharpening circuit.

The color image sharpening circuit for one pixel is designed by Verilog HDL and synthesized by Synopsys Design Compiler same as evaluation for adder units. Test bench that run on Synopsys VCS read a color image file, separate each pixel data to three primary colors, and repeatedly provide the separated pixel data to each monochrome image sharpening circuit. If the 256 x 256 pixel image is given, data input is repeated at 65536 times. We assumed that the input frequency is 100MHz in VCS. Switching activity file (SWIF) is also obtained through the VCS simulation. At the end of VCS simulation, buffered outputs from the three sharpening circuits are joined. Finally, sharpened color image file is generated by test bench. The generated image files can be used to evaluate accuracy for approximation by CMA and CMA with EC.

Peak signal noise ratio (PSNR) is widely used as criteria for quality of images [6]. PSNR can be used for relative comparison between generated images from the color image sharpening circuit. The input images for the circuit are obtained from SIDBA web site [7]. In this evaluation, 256×256 pixels “ Lenna” and “ Mandrill” are used from the images.

In this paper, to evaluate accuracy, five step accuracy configuration names are defined for conventional 16 bit CMA and proposed CMA with EC. ACCURATE means that the all sub-adders are not masked. MSK1 means that only the least significant sub-adder unit is masked. MSK2 means that the least significant sub-adder unit and its next higher level sub-adder are masked. MSK3 and MSK4 are masked the same manner of these. In CMA with EC, notice that appropriate bnd signal is provided at the same time of configuration. For example, when MSK2 is configured, only one bnd signal of the second level sub-adder is asserted.

6 Experimental Results

6.1 Area, Delay and Power for An Adder Unit

Table 1: Area and Delay Results for An Adder Unit

	Area(um)	Delay(ns)
CMA	90.972	1.06
CMA with EC	96.292(5.8%up)	1.24(16.9%up)

Table 2: Leakage Power Results for An Adder Unit

	Leakage Power(uW)
CMA	2.4668
CMA with EC	2.6043(5.5%up)

Table 3: Accuracy Results of MED for An Adder Unit

	CMA	CMA with EC	Improvement(%)
MSK1	3.7	2.8	23.3
MSK2	63.7	47.8	24.9
MSK3	1023.75	767.8	24.9
MSK4	16385.6	12287.8	24.9

Table 1 shows area and delay results for an adder unit. The CMA with EC increases by 5.8% of implementation area, and by 16.9% of critical path delay. Those overhead is due to additional two gates represented by Figure 9. Especially, in delay results, EC negatively affect circuit performance since the additional OR gate is located in the path that calculates ripple carry-chain. However, CMA does not aim to reduce critical path delay because of ripple carry connection. Moreover, conventional CMA has area overhead against ripple carry adder (RCA). CMA can achieve low power consumption instead of delay and area overhead. Therefore, delay and area overhead by our proposal might be accepted as the same uses for CMAs.

Table 2 shows static power consumption results for an adder unit. The CMA with EC increases by 5.5% of leakage power because of the area overhead. However, it is nescessary to pay attention to that the increase of leakage power is reratively small compared with dynamic power consunption. Figure 14 shows dynamic power consumption. In all configurations, on average, 0.3% of power consumption is decreased by proposed CMA with EC. However, this reduction ratio is in error range of Synopsys VCS simulation and Power Compiler. In principle, extra power must be required in additional two gates. In fact, 1.4% extra power is consumed from MSK1 to MSK3 on average. From the result, we can conclude that the EC only consume ignorable power. About 4.5uW can be saved for each mask bit. On the other hand, EC consume extra power by about 1uW on average. This conclusion is important because the EC can provide non-negligible improvement for accuracy.

6.2 Evaluation of Accuracy for An Adder Unit

Table 3 and 4 show MED and MRED respectively. In all configurations, on average, EC can improve accuracy for MED by 24.5%, for MRED by 29.3%. As same as figure 7 and 8, in all mask setting, accuracy is improved for both criteria. Especially, MSK4 achieves maximum improvement by 39.1%. The most important characteristic from the table is that over 20% of minimum improvement is observed for both MED and MRED. Therefore, we can conclude that proposed EC can achieve stable error correction capability regardless of accuracy configurations.

Table 4: Accuracy Results of MRED for An Adder Unit

	CMA($\times e^{-5}$)	CMA with EC($\times e^{-5}$)	Improvement(%)
MSK1	7.9	6.0	23.4
MSK2	131.1	100.7	25.5
MSK3	2198.0	1551.3	29.4
MSK4	33332.9	20306.7	39.1

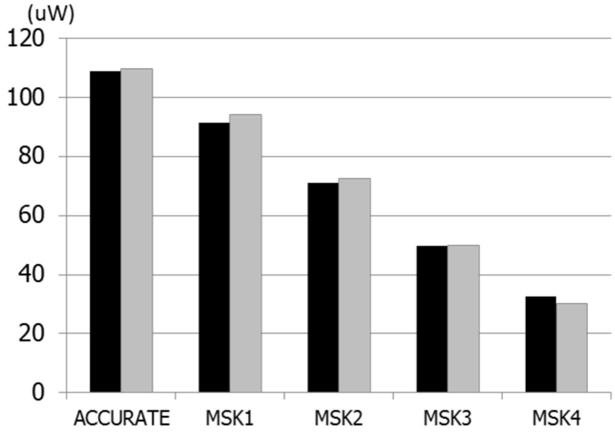


Figure 14: Dynamic Power Consumption for An Adder Unit.

6.3 Discussion of Trade-off between Accuracy and Power Reduction

Proposed EC can improve accuracy while increasing power consumption. In general, accuracy and power reduction are in the relationship of trade-off. Figure 15 and 16 show Power vs. MED and Power vs. MRED respectively. In the figures, X axis means MED or MRED, Y axis means Power. In both figure, a polygonal line of CMA with EC is always located under CMA.

6.4 Evaluations in Actual Image Processing Circuts

Table 5: Area and Delay Results for Actual Implementation of Image Sharpening Circuit

	Area(um)	Delay(ns)
CMA	11614.091	9.13
CMA with EC	12156.732(4.6%up)	9.22(9.22%up)

Table 5 shows area and delay results for a color image sharpening circuit that is illustrated as Figure 12. By using CMA with EC, implementation area increases by 5.8%, and critical path delay increases by 0.9%. The percentage of area overhead is almost same as results for an adder unit since the circuit is constructed as adder unit array. On the other hand, delay is relatively small compared with results for an adder unit. It is because the delay of connection between adder units might be dominant.

Table 6 shows static power consumption results for an adder unit. The leakage power increases by 4.7% by adding the error corrector to conventional circuit. The leakage power overhead is reasonable for increase of area overhead.

Figure 17 shows dynamic power consumption while each image processing. Y axis means dynamic power consumption, X axis means mask configurations. Results of " Lenna" image represents as

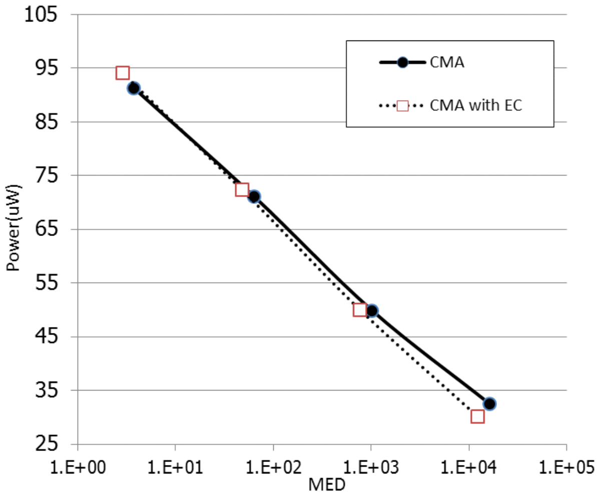


Figure 15: Power vs. MED for An Adder Unit.

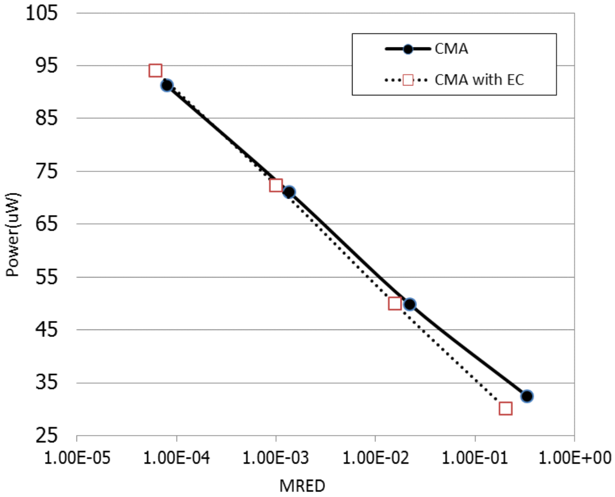


Figure 16: Power vs. MRED for An Adder Unit.

Table 6: Leakage Power Results for Actual Implementation of Image Sharpening Circuit

	Leakage Power(uW)
CMA	306.9021
CMA with EC	321.6187(4.7%up)

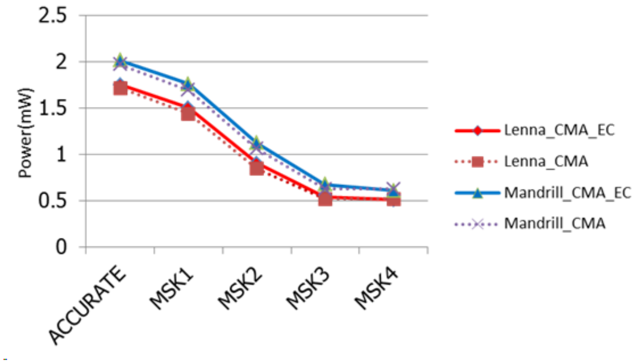


Figure 17: Dynamic Power Results from Implementation of Color Image Sharpening Circuit.

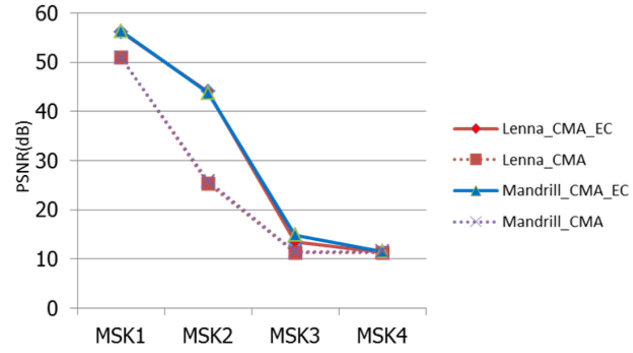


Figure 18: PSNR Results from Implementation of Color Image Sharpening Circuit.

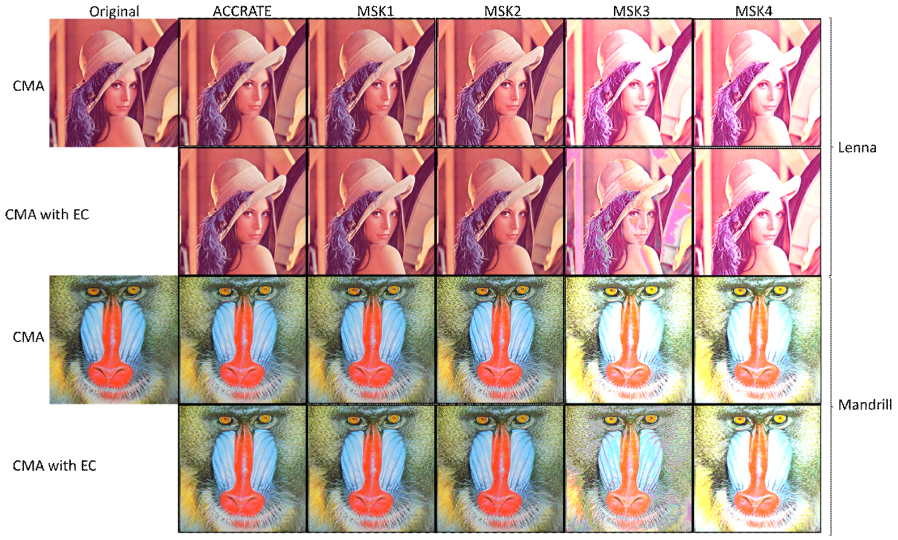


Figure 19: Test Picture Outputted from implementation of image sharpening circuit.

red lines. Results of “ Mandrill” image represents as blue lines. Solid lines mean results of CMA with EC. Dashed lines mean results of conventional CMA. Results of Conventional CMA are always located under CMA with EC. However, the distance between CMA and CMA with EC is quite close, and following each other without leaving. This result means that the power overhead from error corrector is almost constant regardless of mask and bnd configuration. Therefore, we can conclude that the proposed EC can be used to any circuits without hesitation since estimation of power consumption would be predictable.

Figure 18 shows PSNR results for the color image sharpening circuit. 256 × 256 pixels “Lenna” and “ Mandrill”color image are processed by the circuit. Y axis means PSNR, X axis means mask configurations. Red or blue and solid or dashed lines mean the same manner of Figure 17. Higher value for PSNR means that the output image keeps higher quality. In each image, results of CMA and CMA with EC are almost overlapped respectively. This result means that the quality of output images is not related to contents of images. On the other hand, CMA with ED is always located upper CMA. That is, CMA with ED can always produce even or higher quality images than CMA. Especially in configuration of MSK2, the distance between CMA and CMA with EC is long. PSNR is improved by 73.71%. This result means that the proposed EC works better than the other mask configurations since the each pixel data is provided as 8 bit data.

Figure 19 shows images of “ Lenna” and “ Mandrill” that have been processed by CMA and CMA with EC on each mask configurations. The far left image is original image. Images in the second column are sharpened images by using accurate adder. Images in from third to sixth column are sharpened images by using CMA or CMA with EC. The first row represents conventional CMA. The second row represents CMA with EC. In color image processing, error in each pixel data reflect as color fading. From human eyes, in “ Lenna” image, obvious deference between CMA and CMA with EC can be recognized in MSK2. In “ Mandrill” image, obvious deference between CMA and CMA with EC can be recognized in MSK3. From these images, we can intuitively confirm validity of proposed error correctionmechanism.

7 Conclusion

This paper proposed error correction mechanism for approximate adder, CMA that accuracy can be dynamically configured. Proposed error corrector focuses on negative error direction that is produced by CMA, and corrects toward positive direction. Proposed error corrector only requires additional two gates for each digit. In our evaluations, we represented that proposed error corrector does not consume much power. In addition, this paper discussed the relationship of trade-off between accuracy and power reduction, showed the higher power efficiency for accuracy compared with conventional CMA. To confirm the validity of our proposal in actual image processing applications, example algorithm, image sharpening, was designed as a dedicated circuit, was evaluated in performance index of area, delay, and power consumption. From the view point of the index, we showed that the proposed error corrector caused around 5% of overhead on each performance index, and conclude that the overhead is reasonable for contribution that the error corrector increases accuracy. In addition, from the view point of PSNR quality and eyesight, this paper also showed that proposed error corrector can produce high quality images in the both criteria.

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..

References

[1] NanGate Inc. Freepdk45 open cell ibrary. http://www.nangate.com/. Last accessed on Sep 6 2018, 2018.

[2] C.Liu J.Han and F.Lombardi. A low-power, high-performance approximate multiplier with configurable partial error recovery. In *Proc. of Design, Automation & Test in Europe (DATE 2014)*, 2014.

[3] N.Zhu W.L.Goh W.Zhang K.S.Yeo and Z.H.Kong. Design of low-power high-speed truncation-error tolerant adder and its application in digital signal processing. *IEEE Transaction on VLSI Systems*, 18(8):1225–1229, 2010.

[4] M.S.Lau K.V.Ling and Y.C.Chu. Energy-aware probabilistic multiplier: Design and analysis. In *Proc. of International Conference on Compliers, architecture, and synthesis for embedded systems (CASES 2018)*, pages 281–290, 2009.

[5] Sbib-Lien Lu. Design of low-power high-speed truncation-error tolerant adder and its application in digital signal processing. *IEEE Computer*, 37(3):67–73, 2004.

[6] R.Ye T.Wang F.Yuan R.Kumar and Q.Xu. On reconfiguration-oriented approximate adder design and its application. In *Proc. of IEEE/ACM International Conference on Computer-Aided Design (ICCAD 2013)*, pages 48–54, 2013.

[7] SIDBA. Standard image data-base. http://vision.kuee.kyoto-u.ac.jp/IUE/IMAGE_DATABASE/STD_IMAGES/. Last accessed on Sep 6 2018, 2018.

[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 Transactions on Circuits and Systems I: Regular Papers*, 57(4):850–862, 2010.

[9] T.Yang T.Ukezono and T.Sato. A low-power configurable adder for approximate applications. In *Proc. of 19th International Symposium on Quality Electronic Design (ISQED 2018)*, pages 347–352, 2018.