# A Low-Power Yet High-Speed Configurable Adder for Approximate Computing

Tongxin Yang[1] Tomoaki Ukezono[2] Toshinori Sato[3]
[1] Graduate School of Information and Control Systems, Fukuoka University, Japan
[2,3]Department of Electronics Engineering and Computer Science, Fukuoka University, Japan
[1]Email: td166502@cis.fukuoka-u.ac.jp [2]Email: tukezo@fukuoka-u.ac.jp [3]Email: toshinori.sato@computer.org

*Abstract*—**Approximate computing is an efficient approach for error-tolerant applications because it can trade off accuracy for power. Addition is a key fundamental function for these applications. In this paper, we proposed a low-power yet high-speed accuracy-configurable adder that also maintains a small design area. The proposed adder is based on the conventional carry look-ahead adder, and its configurability of accuracy is realized by masking the carry propagation at runtime. Compared with the conventional carry look-ahead adder, with only 14.5% area overhead, the proposed 16-bit adder reduced power consumption by 42.7%, and critical path delay by 56.9% most according to the accuracy configuration settings, respectively. Furthermore, compared with other previously studied adders, the experimental results demonstrate that the proposed adder achieved the original purpose of optimizing both power and speed simultaneously without reducing the accuracy.**

***Keywords—Approximate computing; accuracy-configurable adder; high-speed adder; low-power adder***

## I. Introduction

Applications that have recently emerged (such as image recognition and synthesis, digital signal processing, which is computationally demanding, and wearable devices, which require battery power) have created challenges relative to power consumption. Addition is a fundamental arithmetic function for these applications [1] [2]. Most of these applications have an inherent tolerance for insignificant inaccuracies. By exploiting the inherent tolerance feature, approximate computing can be adopted for a tradeoff between accuracy and power. At present, this tradeoff plays a significant role in such application domains [3]. As computation quality requirements of an application may vary significantly at runtime, it is preferable to design quality-configurable systems that are able to tradeoff computation quality and computational effort according to application requirements [4] [5]. The previous proposals for configurability suffer the cost of the increase in power [5] or in delay [12]. In order to benefit such application, a low-power and high-speed adder for configurable approximation is strongly required.

In this paper, we propose a configurable approximate adder, which consumes lesser power than [5] does with a comparable delay and area. In addition, the delay observed with the proposed adder is much smaller than that of [12] with a comparable power consumption. Our primary contribution is that, to achieve accuracy configurability the proposed adder achieved the optimization of power and delay simultaneously and with no bias toward either. We implemented the proposed adder, the conventional carry look-ahead adder (CLA), and the ripple carry adder (RCA) in Verilog HDL using a 45-nm library. Then we evaluated the power consumption, critical path delay, and design area for each of these implementations. Compared with the conventional CLA, with 1.95% mean relative error distance (MRED), the proposed adder reduced power consumption and critical path delay by 42.7% and 56.9%, respectively. We provided a crosswise comparison to demonstrate the superiority of the proposed adder. Moreover, we implemented two previously studied configurable adders to evaluate power consumption, critical path delay, design area, and accuracy. We also evaluated the quality of these accuracy-configurable adders in a real image processing application.

## II. 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 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. The above static approximate designs [6-8] with fixed accuracy may fail to meet the quality requirements of applications or result in wastage of power when high quality is not required.

Kahng et al. [4] proposed an accuracy-configurable adder (ACA), which is based on a pipeline structure. The correction scheme of the ACA proceeds from stage 1 to stage 4, if the most significant bits of the results are required to be correct, all the four stages should be performed. Motivated by the above, Ye et al. [5] proposed an accuracy gracefully-degrading adder (GDA) which allows the accurate and approximate sums of its sub adders to be selected at any time. Similar to [5], our adder proposed in this paper does not consider a pipeline structure either. 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]. However, the additional logic blocks require more area. Furthermore, these blocks will cause power wastage when their outputs are not used to generate a sum. This problem was addressed by [12] based on a low-power configurable adder that generates an approximate sum by using OR gates. Similar to [12], the proposed adder also uses OR gates to generate an approximate sum, but [12] focuses on only power consumption and its delay is large. Thus, it may fail to meet the speed requirement of an application.

## III. Proposed Accuracy-configurable Adder

Typically, a CLA consists of three parts: (1) half adders for carry generation (G) and propagation (P) signals preparation, (2) carry look-ahead units for carry generation, and (3) XOR gates for sum generation. We focus on the half adders for G and

978-1-5386-4881-0/18/$31.00 ©2018 IEEE

P signals preparation in part 1. Consider an n-bit CLA; each part of it can be obtained as follows:

$$\mathrm{P}_i = \mathrm{A}_i \oplus \mathrm{B}_i,\ \mathrm{G}_i = \mathrm{A}_i \cdot \mathrm{B}_i, \qquad (1)$$

$$\mathrm{C}_i = \mathrm{G}_i + \mathrm{P}_i \cdot \mathrm{C}_{i-1}, \qquad (2)$$

$$\mathrm{S}_i = \mathrm{P}_i \oplus \mathrm{C}_{i-1}. \qquad (3)$$

where *i* is denoted the bit position from the least significant bit. Note that owing to reuse of the circuit of $A_i$ *XOR* $B_i$ for $S_i$ generation, here $P_i$ is defined as $A_i$ *XOR* $B_i$ instead of $A_i$ *OR* $B_i$. Because $C_0$ is equal to $G_0$ , if $G_0$ is 0, $C_0$ will be 0. From (2), we find that $C_1$ is equal to $G_1$ when $C_0$ is 0. In other words, if $G_0$ and $G_1$ are equal to 0, $C_0$ and $C_1$ will be 0. By expanding the above to *i* , $C_i$ will be 0 when $G_0$, $G_1$, … , $G_i$ are all 0. This means that the carry propagation from $C_0$ to $C_i$ is masked. From (3), we can obtain that $S_i$ is equal to $P_i$ when $C_{i-1}$ is 0.

From the perspective of approximate computing, if G is controllable and can be controlled to be 0, the carry propagation will be masked and S (=P) can be considered as an approximate sum. In other words, we can obtain the selectivity of S between the accurate and approximate sum if we can control G to be A *AND* B or 0. Evidently, we can achieve selectivity by adding a select signal. Figure 1(a) is a conventional half adder and Fig. 1(b) is a half adder to which the select signal has been added. Compared with the conventional half adder, we add a signal named "M_X" as the select signal and use a 3-input AND gate to replace the 2-input one. When M_X = 1, the function of G is the same as that of a conventional half adder; when M_X = 0, G is equal to 0.

Consider the condition when the inputs $A_i$ and $B_i$ are both 1, when $M\_X_i = 1$, the accurate sum $S_i$ and carry $C_i$ will be 0 and 1 ($\{C_i , S_i\} = \{1,0\}$); when $M\_X_0$ , $M\_X_1$ , … , $M\_X_i$ are all 0, $S_i$ is equal to $P_i$ (= $A_i$ *XOR* $B_i$ = 0) as an approximate sum and $C_i$ is equal to 0 ($\{C_i , S_i\} = \{0, 0\}$) as discussed above. Here {,} denotes concatenation. This means that the difference between the accurate and approximate sum is 2. Toward better accuracy results for the approximate sum, we use an OR function instead of an XOR function for P generation when M_X = 0. Thus, the difference will be reduced to 1. A 2-input XOR gate can be implemented by using a 2-input NAND gate, a 2-input OR gate, and a 2-input AND gate. An equivalent circuit of the conventional half adder is shown in Fig. 2. This is called a carry-maskable half adder (CMHA). The dashed frame represents the equivalent circuit of a 2-input XOR (M_X = 1). We can obtain the following: P is equal to A *XOR* B, and G is equal to A *AND* B when M_X = 1; when M_X = 0, P is equal to A *OR* B and G is 0. Thus, M_X can be considered as a carry mask signal.

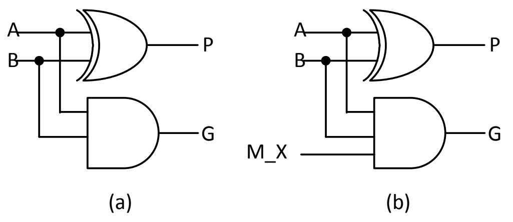

Fig. 1. (a) An accurate half adder, and (b) a half adder with a select signal.

Consider an n-bit CLA, whose half adders for G and P signals preparation are replaced by CMHAs. In this case, an n-bit carry mask signal for each CMHA is required. To simplify the structure for masking carry propagation, we group four CMHAs and use a 1-bit mask signal to mask the carry propagation of the CMHAs in each group. The structure of a group with four CMHAs is shown in Fig. 3 as an example. $A_{3\text{-}0}$, $B_{3\text{-}0}$, $P_{3\text{-}0}$, and $G_{3\text{-}0}$ are 4-bit-length signals and represent {$A_3$, $A_2$, $A_1$, $A_0$}, {$B_3$, $B_2$, $B_1$, $B_0$}, {$P_3$, $P_2$, $P_1$, $P_0$}, and {$G_3$, $G_2$, $G_1$, $G_0$,}, respectively. $M\_X_0$ is a 1-bit signal and is connected to the four CMHAs to mask the carry propagation simultaneously. When $M\_X_0 = 1$, $P_{3\text{-}0} = A_{3\text{-}0}$ *XOR* $B_{3\text{-}0}$, and $G_{3\text{-}0} = A_{3\text{-}0}$ *AND* $B_{3\text{-}0}$; when $M\_X_0 = 0$, $P_{3\text{-}0} = A_{3\text{-}0}$ *OR* $B_{3\text{-}0}$, and $G_{3\text{-}0} = 0$. We proposed an accuracy-configurable adder by using CMHAs to mask the carry propagation.

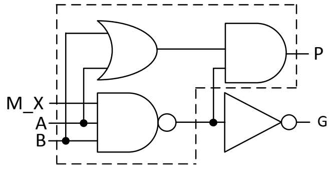

Fig. 2. A carry-maskable half adder.

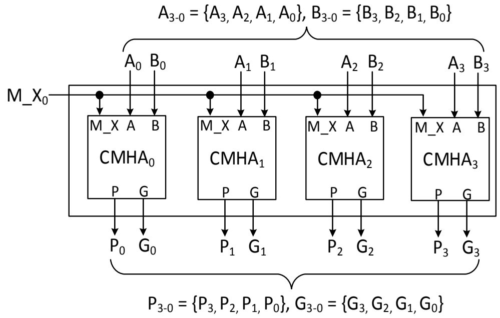
A3-0 = {A3, A2, A1, A0}, B3-0 = {B3, B2, B1, B0}
A0 B0
A1 B1
A2 B2
A3 B3
M_X0
M_X A B
CMHA0
CMHA1
CMHA2
CMHA3
P G
P0 G0
P1 G1
P2 G2
P3 G3
P3-0 = {P3, P2, P1, P0}, G3-0 = {G3, G2, G1, G0}

Fig. 3. Structure of a group with four CMHAs.

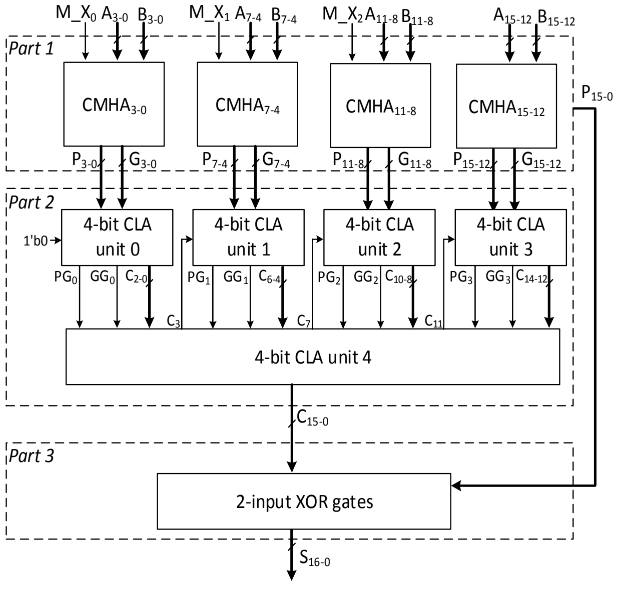

Fig. 4. Structure of the proposed 16-bit adder.

The structure of the proposed 16-bit adder is shown in Fig. 4 as an example. Four groups (CMHA$_{3\text{-}0}$, CMHA$_{7\text{-}4}$, CMHA$_{11\text{-}8}$, and CMHA$_{15\text{-}12}$) are used to prepare the P and G signals. Each group comprises four CMHAs There is no mask signal for CMHA$_{15\text{-}12}$ in this example; therefore, accurate P$_{15\text{-}12}$ (= A$_{15\text{-}12}$ *XOR* B$_{15\text{-}12}$) and G$_{15\text{-}12}$ (= A$_{15\text{-}12}$ *AND* B$_{15\text{-}12}$) are always obtained. P$_{15\text{-}0}$ and G$_{15\text{-}0}$ are the outputs from Part 1 and are connected to Part 2. Note that P$_{15\text{-}0}$ is also connected to Part 3 for sum generation. In Part 2, four 4-bit carry look-ahead units (unit 0, 1, 2, 3) generate four PGs (PG$_0$, PG$_1$, PG$_2$, and PG$_3$), four GGs (GG$_0$, GG$_1$, GG$_2$, and GG$_3$), and 12 carries (C$_{2\text{-}0}$, C$_{6\text{-}4}$, C$_{10\text{-}8}$, and C$_{14\text{-}12}$) first, and then the carry look-ahead unit 4 generates the remaining four carries (C$_3$, C$_7$, C$_{11}$, and C$_{15}$) by using the PGs and GGs. C$_{15\text{-}0}$ is the output of Part 2 and is connected to Part 3. The fifteen 2-input XOR gates in Part 3 generate the sum.

## IV. Experimental Results

### A. Experimental Setup

To clarify the contributions to the power and delay of the proposed adder, we implement and evaluate the proposed adder, the conventional CLA and RCA, the previously studied configurable adder GDA [5], and [12]. All these adders are 16-bit. The bit-length of the sub adder units in GDA is set to be four bits. The number of carry-in prediction bits in GDA is set to be 0, 4, 8, and 12 bits. The proposed adder and adder [12] also separate 16 bits to four 4-bit-length groups; therefore, the configuration settings of the proposed adder, adder [12], and GDA are the same. The different configuration settings of these adders are referred to as A_1, A_2, A_3, A_4 (for the proposed adder), B_1, B_2, B_3, B_4 (for adder [12]), and GDA1, GDA2, GDA3, GDA4 (for GDA), respectively. For example, A_1 means that the carry propagations of CMHA$_{3\text{-}0}$, CMHA$_{7\text{-}4}$, and CMHA$_{11\text{-}8}$ are masked (M_X$_0$ = M_X$_1$ = M_X$_2$ = 0). Similar considerations apply to adder [12] (B_1, B_2, B_3, and B_4).

All adders were coded using Verilog HDL. The Synopsys VCS was used to evaluate the numerical outputs of all the adders with one million randomly-generated input patterns to these adders, 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 [9]. The power consumption was evaluated at a frequency of 0.5 GHz. All the designs were synthesized and optimized with default compile options using the typical conditions of libraries. The Synopsys Power Compiler was used to estimate the power consumption from the switching activity interchange format files that were generated from the VCD files.

### B. Power, Delay, and Area Results

The comparison results for the adders are shown in Fig. 5, where the y-axis is the percentage of each adder from the conventional CLA. Smaller values represent better results. The values of the conventional CLA are considered to be 100%. The x-axis is the names of the evaluated adders. The two bars (left to right) in each adder show the percentages of power and delay.

Among the configurable adders, adder [12] delivers the best results for power consumption The power consumption with B_1 is the lowest among all the adders. The delays observed with B_3 and B_4 are much larger than those of the proposed adder and GDA in the same configurable setting. This demonstrates that adder [12] is very effective with regard to power consumption, but it does not work well to reduce delay. Although the delay of GDA1 is larger than those of the proposed adder and adder [12], GDA delivers unmistakably good results with regard to delay as a whole. Unfortunately, GDA involves the largest power consumption among all the adders.

As can be seen, the heights of the two bars in Fig. 5 are close; this means that the proposed adder delivers a balance between power consumption and delay. Compared with GDA in each configuration setting, the power consumption of the proposed adder is much smaller than that of GDA. Furthermore, the delays of A_1 and A_2 are also smaller than those of GDA1 and GDA2, respectively. The power consumption of A_4 is 41.2% smaller than that of GDA4, and the delay of A_4 is 12.9% larger than that of GDA4. Compared with adder [12] in each configuration setting, although the power consumption of the proposed adder is slightly larger than that of adder [12], its delay is much smaller than that of adder [12]. For example, the delays of A_3 and A_4 are 35.1% and 60.0% smaller than those of B_3 and B_4, respectively.

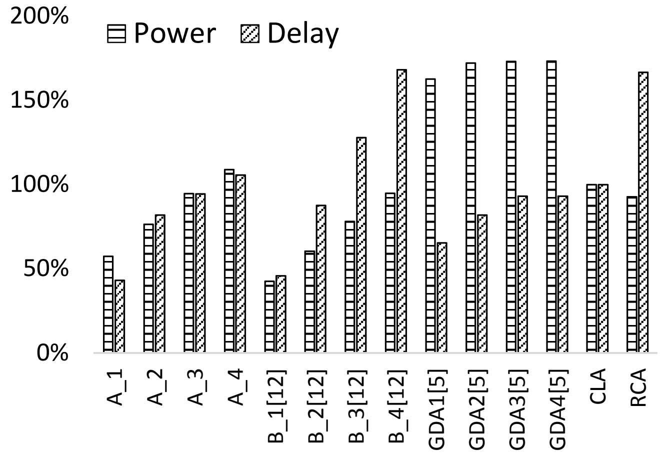

Fig. 5. Percentages (%) compared with the CLA in power and delay.

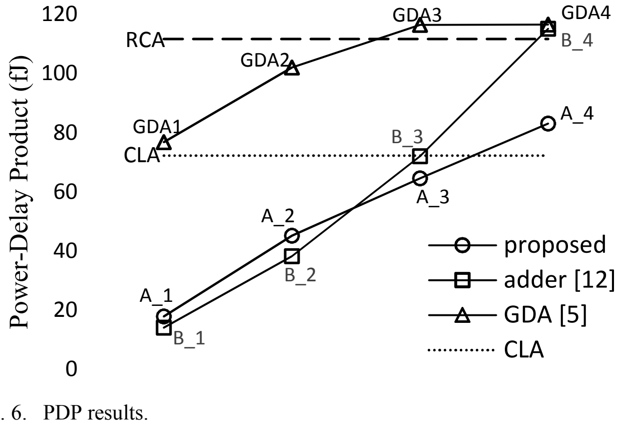

Fig. 6. PDP results.

TABLE I. Area results.

| Proposed | Adder [12] | GDA [5] | CLA | RCA |
|---|---|---|---|---|
| 125.55 | 83.79 | 174.23 | 109.59 | 78.47 |

The power-delay product (PDP) is proposed to evaluate approximate arithmetic circuits [2]. The results of PDP for these adders are shown in Fig. 6 for a better overview of the circuit characteristics. The circles, rectangles, and triangles represent the proposed adder, adder [12], and GDA, respectively. Smaller values represent better results in energy savings. A_3 and A_4 deliver the best results among the configurable adders in their configuration settings, especially the PDP of A_4, which is about 72% of those of B_4 and GDA4. Although the PDP values of A_1 and A_2 are slightly larger than those of B_1 and B_2, respectively, they are much smaller than those of GDA1 and GDA2. This also demonstrates that our purpose of optimizing both power and delay has been achieved. Obviously, the proposed adder delivers great improvements in energy, particularly when an accurate sum is required.

Table I shows a comparison of the design area results, in square microns. Note that the accuracy configuration setting does not have any effect on the design area of the configurable adders. The area of the proposed adder is 72% of that of GDA. Adder [12] is the smallest among the configurable adders.

### C. Accuracy Results

The error distance (ED) and mean ED (MED) are proposed to evaluate the performance of approximate arithmetic circuits [10]. 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 the EDs for a set of outputs. The relative error distance (RED) is the ED divided by the accurate output, that is, RED = |S - S′|/S, and MRED is the average of the REDs which can be obtained in a manner similar to how MED is calculated. 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 II compares the accuracy results. Because both the proposed adder and adder [12] mask the carry propagation to achieve an approximate sum by an OR function, all the three metrics of the proposed adder (A_1, A_2, A_3, and A_4) and adder [12] (B_1, B_2, B_3, and B_4) in the configuration settings are the same, respectively. Both MED and MRED of the proposed adder and adder [12] are smaller than those of GDA in each setting of the configuration. Note that there are no errors in A_4, B_4 and GDA4; therefore, the results of all the metrics are **0.00**. Although the ER values of the proposed adder and adder [12] are larger than that of GDA in each accuracy configuration setting, their MED and MRED values are about 50% of GDA. The accuracy comparison results of the adders demonstrate that the proposed adder achieved an unbiased, optimized result between power and delay without reducing the accuracy.

TABLE II. ACCURACY COMPARISON.

| | MED | MRED ($10^{-4}$) | ER (%) |
|---|---|---|---|
| A_1, B_1 | 1012.62 | 195.14 | 95.95 |
| A_2, B_2 | 58.38 | 10.38 | 88.36 |
| A_3, B_3 | 3.72 | 0.79 | 68.15 |
| GDA1 | 2022.95 | 388.69 | 83.08 |
| GDA2 | 116.81 | 20.53 | 5.58 |
| GDA3 | 6.59 | 1.62 | 0.16 |

TABLE III. PSNR RESULTS OF CONFIGURABLE ADDERS, IN DB.

| No. | A_1, B_1 | A_2, B_2 | A_3, B_3 | GDA1 | GDA2 | GDA3 |
|---|---|---|---|---|---|---|
| 1 | 7.79 | 27.01 | 49.60 | 7.45 | 25.86 | 40.58 |
| 2 | 8.88 | 27.83 | 51.49 | 8.32 | 27.90 | 41.86 |
| 3 | 12.11 | 28.44 | 51.38 | 11.05 | 25.44 | 42.15 |
| 4 | 9.62 | 27.79 | 51.68 | 8.79 | 26.61 | 40.18 |
| 5 | 11.20 | 27.35 | 49.67 | 10.34 | 26.12 | 40.09 |
| 6 | 8.88 | 27.24 | 49.60 | 8.43 | 24.93 | 39.45 |

## V. Image Processing

In this section, the evaluation of an image processing application with the proposed adder is described. An image sharpening algorithm [11] that is popular for the evaluation of approximate adders was used. Six images collected from the Internet were utilized. These were 512 × 512, 8-bit grayscale bitmap images. Only the additions were replaced by the adders; all the other operations (multiplication, subtraction, and division) were accurate. The processed image quality was measured using the peak signal-to-noise ratio (PSNR). This method is usually used to measure the quality of reconstructive processes that involve information loss.

Table III gives the PSNR results of these configurable adders in dB. Larger values represent better quality images. As can be seen, all the PSNR values of the proposed adder as well as adder [12] are larger than those of GDA in the same configuration setting. The PSNR results demonstrate that our proposed adder and adder [12] deliver better quality images than GDA. This shows that the proposed adder and adder [12] deliver better accuracy than GDA again. A_4, B_4 and GDA4 are accurate, and there are no PSNR results corresponding to these configurations.

## VI. Conclusion

In this paper, an accuracy-configurable adder without suffering the cost of the increase in power or in delay for configurability was proposed. The proposed adder is based on the conventional CLA, and its configurability of accuracy is realized by masking the carry propagation at runtime. The experimental results demonstrate that the proposed adder delivers significant power savings and speedup with a small area overhead than those of the conventional CLA. Furthermore, compared with previously studied configurable adders, the experimental results demonstrate that the proposed adder achieves the original purpose of delivering an unbiased optimized result between power and delay without sacrificing accuracy. It was also found that the quality requirements of the evaluated application were not compromised.

## 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, and by a grant from the Marubun Research Promotion Foundation. This work is 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 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 Comptuer-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] NanGate, Inc. NanGate FreePDK45 Open Cell Library, http://www.nangate.com/?page_id=2325, 2008

[10] 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.

[11] M. S. 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.

[12] T. Yang, T. Ukezono, and T. Sato, “A Low-Power Configurable Adder for Approximate Applications”, *19th International Symposium on Quality Electronic Design*, Mar. 2018.

978-1-5386-4881-0/18/$31.00 ©2018 IEEE