



# **on Fundamentals of Electronics, Communications and Computer Sciences**

**VOL. E103-A NO. 9  
SEPTEMBER 2020**

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

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

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

**A PUBLICATION OF THE ENGINEERING SCIENCES SOCIETY**



**The Institute of Electronics, Information and Communication Engineers**

**Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3chome, Minato-ku, TOKYO, 105-0011 JAPAN**

# Exploiting Configurable Approximations for Tolerating Aging-induced Timing Violations\*

Toshinori SATO<sup>†a)</sup> and Tomoaki UKEZONO<sup>†</sup>, *Members*

**SUMMARY** This paper proposes a technique that increases the lifetime of large scale integration (LSI) devices. As semiconductor technology improves at miniaturizing transistors, aging effects due to bias temperature instability (BTI) seriously affects their lifetime. BTI increases the threshold voltage of transistors thereby also increasing the delay of an electronics device, resulting in failures due to timing violations. To compensate for aging-induced timing violations, we exploit configurable approximate computing. Assuming that target circuits have exact and approximate modes, they are configured for the approximate mode if an aging sensor predicts violations. Experiments using an example circuit revealed an increase in its lifetime to > 10 years.

**key words:** approximate computing, timing violation, bias temperature instability, configurability, canary flip-flop

## 1. Introduction

Bias temperature instability (BTI) is a major mechanism that determines device lifetime [2] by causing threshold-voltage shift in a transistor, thereby increasing in its delay. After 10 years, circuit delay usually increases by ~15% [2], [3], however, if the delay becomes longer than the clock period, which is determined at design time, a timing violation occurs and can result in system failure. Aging-induced timing violations have become serious issue along with improvements in semiconductor technology, which has increased transistor miniaturization. Because device failure is expected in <10 years in the absence of compensations, techniques to improve their lifetime are required. Additionally, it is difficult to repair failures or replace degraded components, such as internet-of-things (IoT) devices, making methods to automatically increase their lifetime desirable.

Guardbanding is a conventional technique that enables toleration of increases in aging-induced delays. Supply voltages, which are larger than those determined by typical working conditions, are provided in order to compensate for timing violations. However, although the worst-case scenarios rarely occur, overestimated supply voltages are still consistently provided, resulting in wasteful power consumption and a circuit area becoming larger than necessary to satisfy the pessimistic timing constraints. To address these problems, typical-case design methods that focus on typical

working conditions rather than worst-case scenarios have been proposed. When a worst-case scenario occurs, error-tolerant mechanisms manage the timing violations. Canary flip-flop (FF) [4] is a technique used to predict violations on the fly, where guardband voltage is only supplied upon Canary FF prediction of a violation. Unfortunately, Canary FF occasionally fails to predict timing violations [5]. To adopt Canary FF to practical use, its error-tolerance needs to be guaranteed.

Approximate computing [6] has garnered increased interest based on its lower power consumption, smaller circuit area, and smaller circuit-delay characteristics relative to always-exact computing. This study focuses on delay improvement via approximation, with approximate computing applied to compensate for aging-induced timing violations. We implemented a target circuit having two modes (exact and approximate), with changes in mode occurring from the exact to approximate, upon increases in aging-induced delays that result in timing violations. Notably, the approximate mode was implemented based on an expectation of its application under worst case scenarios. This differs from typical-case design methods in that the proposed technique relies on pre-designed diagnosis to predict timing errors in order to eliminate their inaccurate prediction.

This paper is organized as follows. Section 2 surveys related work. Section 3 explains mechanism of BTI degradation and describes the proposed error-tolerant technique in Sect. 4. Section 5 presents the experimental results, and conclusions are presented in Sect. 6.

## 2. Related Work

First in this section, configurable adders, to which the proposed technique will be applied as examples, are described. Next, similar studies where approximate circuits are used to tolerate timing violation are introduced. And last, sensors, which detect increase in delay, are explained.

### 2.1 Configurable Approximate Circuits

Approximate computing allows trade-offs between performance, power, and accuracy. Through slightly diminishing accuracy, circuit delay is improved. This method exploits observations that some contemporary applications are inherently error tolerant and do not always require exact computing results. Applications, such as image processing, machine learning, and recognition, are included in this domain,

Manuscript received November 28, 2019.

Manuscript revised March 27, 2020.

<sup>†</sup>The authors are with Department of Electronics Engineering and Computer Science, Fukuoka University, Fukuoka-shi, 814-0180 Japan.

\*This is an extended version of the paper [1] presented at 8th Global Conference on Consumer Electronics, October 2019.

a) E-mail: tsato@fukuoka-u.ac.jp

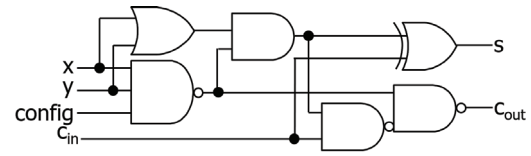
DOI: 10.1587/transfun.2019KEP0009

because they process noisy and redundant data to output an acceptable range of results rather than a unique exact result [16]. Examples of approximate circuits include multiple-bit adders, which are divided into two sections: the upper area comprises full adders (FAs) and is accurate, whereas the lower area is approximated by replacing FAs with simpler gates, such as OR gates, or through truncation. Another example is an approximate multiplier in which the less-significant bits are truncated based on their decreased importance in most cases.

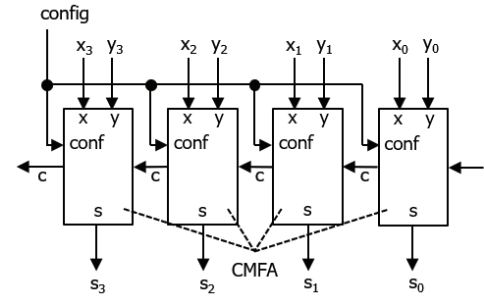
Some approximate adders are configurable and have both exact and approximate modes. The gracefully degrading adder (GDA) proposed by Ye et al. [10] uses exact and approximate modes along with a dynamically configurable carry generator to improve delay in the approximate mode. Unfortunately, GDA exhibits serious area and power overheads. Angizi et al. [11] proposed a low-power adder with both modes and that consumes 20.4% less power in approximate mode relative to exact mode. Unfortunately, its delay does not improve in approximate mode, thereby making it unusable for tolerating timing violations. Hassani et al. [12] proposed an ultralow-power adder with exact and approximate modes and a reduced power consumption relative to that of a conventional exact adder, even in the exact mode. Unfortunately, its delay is severely larger than that of the previously studied approximate adder.

Carry-maskable adder (CMA) proposed by Yang et al. [17] solves the aforementioned problems in area, delay, or power. It does not rely on redundant circuits or multiplexers, which differs from GDAs [10]. The CMA combines conventional exact and approximate adders into a simple structure, that results in minimal area overhead. Figure 1(a) shows a carry-maskable full adder (CMFA) [17]. In exact mode, where the signal *config* equals 1, the CMA works as an FA; otherwise, in approximate mode, where *config* equals 0, the CMFA becomes an OR gate (note that it is assumed that  $C_{in}$  is 0 in approximate mode). Figure 1(c) shows an example of a 16-bit CMA comprising a exact 4-bit adder and three 4-bit CMAs.  $C_{in}$  in the least-significant bit (LSB) is assumed to be 0. As shown in Fig. 1(b), every 4-bit CMA comprises four CMFAs that share a single configuration signal; therefore, a 16-bit CMA has four configurations. When a set of configuration signals  $\{C_2, C_1, C_0\}$  is  $\{1, 1, 1\}$ , the circuit works in exact mode as a 16-bit ripple carry adder (RCA); otherwise it works in approximate mode. When the configuration set  $\{C_2, C_1, C_0\}$  is  $\{1, 1, 0\}$ ,  $\{1, 0, 0\}$ , or  $\{0, 0, 0\}$ , the least-significant four, eight, and 12 bits are approximate. As described, the length of carry chain is reduced in the approximate mode, with this structure selected by intending a small number of configuration signals. It is also possible to choose other structures.

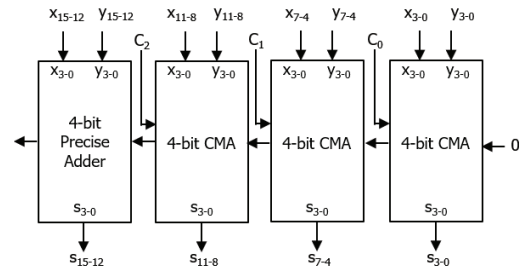
Carry predicting adder (CPredA) [18] is another example of a dynamically configurable approximate circuit, which solves the above problems. It is constructed with a CMFA and a carry predicting full adder (CPFA) [18] [Fig. 2(a)]. In exact mode, where the signal *config* equals 1, CPredA works as a conventional FA; otherwise, in ap-



(a) Carry-Maskable Full Adder



(b) 4-bit CMA



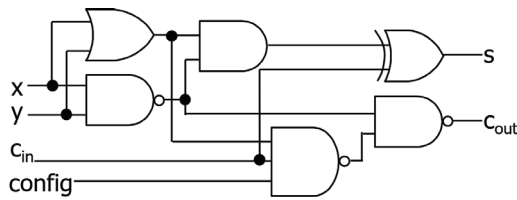
(c) 16-bit CMA

**Fig. 1** Carry-maskable adder (CMA).

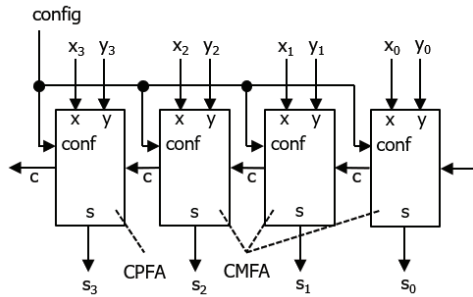
proximate mode, where *config* equals 0, the carry propagation from  $C_{in}$  to  $C_{out}$  is cut off. A 4-bit CPredA comprises one CPFA and three CMFAs from the most-significant bit (MSB) to the LSB [Fig. 2(b)]. Similarly, a 16-bit CPredA comprises a exact 4-bit adder and three 4-bit CPredAs from the MSB to the LSB [Fig. 2(c)]. The configuration set  $\{C_2, C_1, C_0\}$  configures 16-bit CPredAs similar to 16-bit CMAs. As shown, the length of the carry chain is reduced in the approximate modes, and similar to CMAs, it is possible to choose other structures.

## 2.2 Timing-Violation-Aware Approximate Circuits

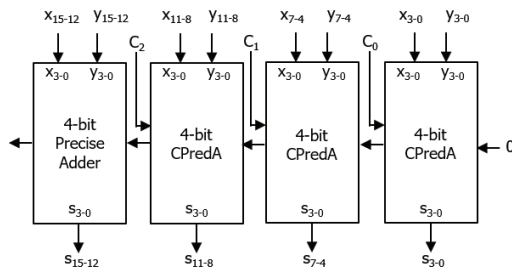
Amrouch et al. [7] promoted exploitation of approximation in order to compensate for aging-induced timing violations, with a target circuit implemented as its approximated version, which prevented aging-induced timing violation due to the smaller circuit delay in the approximated version relative to the original circuit. In this method, approximation was adopted statically at design time, with quality loss consistently occurring regardless of aging-induced effects on delays. Kim et al. [8] utilized a replica of the critical path to monitor increases in aging-induced delays. In this method, the lower bits of the target arithmetic circuit are truncated by an appropriate length upon detection, resulting in appli-



(a) Carry-Predicting Full Adder



(b) 4-bit CPredA



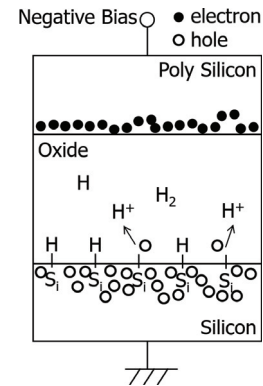
(c) 16-bit CPredA

**Fig. 2** Carry-predicting adder (CPredA).

cation of short carry propagation in order to avoid aging-induced timing violations. A concern of this method is that aging does not always equally affect the target circuit and its replication. Boroujerdian et al. [9] applied adaptive approximation to trade-off temperature guardbands, where increase in temperature causes increased delays, resulting in timing violations. When temperature increases beyond the guard-band, an exact circuit switches to an approximate circuit, and low of quality is avoided due to its dynamic adaptiveness. Although they do not address prediction of timing violations, their method employs a thermal sensor. Therefore, application of this method to aging-induced timing violations requires aging sensor.

### 2.3 Aging Sensors

Nakura et al. [13] proposed defect prediction flip-flop (DPFF). To tolerate latent defects undetectable during the fabrication process, the target circuit is duplicated, and the failed copy switches to its spare when a defect is predicted by DPFF. This technique is applicable to aging-induced timing violations but requires at least twice as large a circuit area. Canary FF [4] predicts timing violations due to variations, including those associated with processes, voltage,

**Fig. 3** NBTI inside PMOS transistor.

temperature, and aging. Upon a prediction of timing variation, the supply voltage is increased to prevent actual errors. Canary FF comprises duplicated conventional FFs and exploits scan FFs for duplication in order to avoid increases in circuit area. Moreover, the total power consumption is reduced along with the circuit area due to removal of the guardbands [14]. Unfortunately, Canary FF does not accurately predict all timing violations, especially when the supply voltage decreases to a nearly critical level and activated paths suddenly change [5].

### 3. The BTI Mechanism

BTI aging negatively affects circuit reliability and potentially increases the threshold voltage of a transistor, which slows transistor activity and results in possible timing violations. This section describes the BTI mechanism within metal-oxide-semiconductor (MOS) transistors.

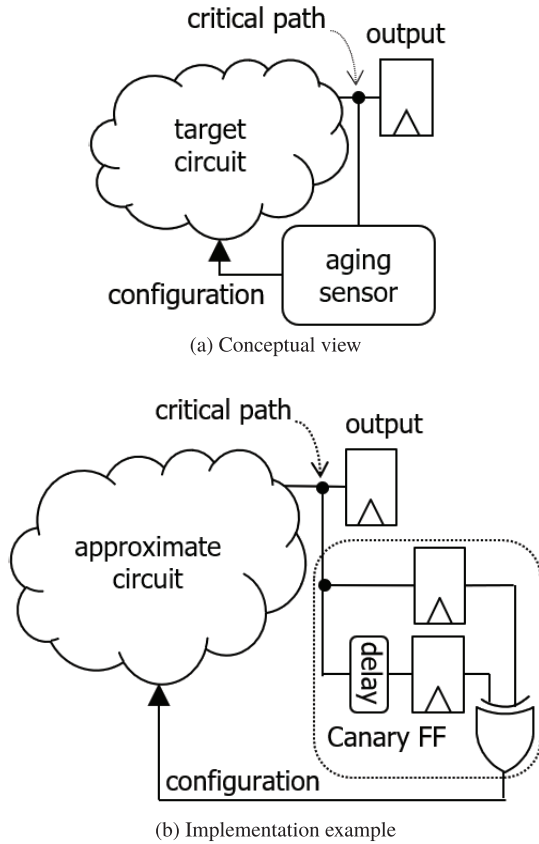
The BTI process is described by the reaction-diffusion model [15]. At the silicon-silicon dioxide ( $S_i - S_iO_2$ ) interface, some silicon atoms bond with hydrogen to form weak silicon-hydrogen bonds ( $S_i - H$ ). Under stress [negative bias for p-type (PMOS) and positive bias for n-type (NMOS) transistors], breakage of  $S_i - H$  bonds occurs at the  $S_i - S_iO_2$  interface and generates broken silicon bonds ( $S_i^-$ ), that represent interface traps, resulting in released hydrogen atoms/molecules ( $H/H_2$ ) that diffuse toward the gate. Figure 3 shows the BTI process inside a PMOS transistor [a negative BTI (NBTI)]. Due to the interface traps, the threshold voltage of the transistor increases. However, when stress is removed, recovery from BTI degradation occurs, and  $H/H_2$  diffuses toward the  $S_i - S_iO_2$  interface to anneal the  $S_i^-$  bonds and reduce the number of interface traps at the  $S_i - S_iO_2$  interface, resulting in recovery of the threshold voltage.

Aging 10 years usually results in a  $\sim 15\%$  increase in circuit delay [2], [3] that can cause timing violations and serious system failures.

### 4. Proposed Technique

An effective method for tolerating aging-induced timing vi-





**Fig. 4** Proposed technique.

olations is to slow the clock frequency of the circuit; however, this diminishes circuit performance. Another method is to increase the supply voltage; however, this causes increased power consumption. Therefore, a technique that tolerates aging-induced timing violations without negatively affecting performance and power consumption is required. To this end, here, we describe a method that exploits approximate computing.

Aging-induced increases in delay can be compensated by approximation; however, losses in accuracy are undesirable in the absence of aging effects. It is expected that approximation will be exploited only after the occurrence of aging-induced timing violations. One way to enforce this is to redundantly combine an exact circuit with its approximated copy; however, this requires almost twice as large a circuit area. By contrast, dynamic configurable approximate circuits create little overhead in circuit area [17], [18].

The proposed method tolerates aging-induced timing violations by utilizing configurable approximations, which were described in Sect. 2.1. In contrast to the redundant combinations previously described, this technique does not suffer severe area overhead. A target circuit should have an exact mode and at least one approximate mode and works in the exact mode before violations occur, after which it works in the approximate mode. CMA and CPredA were candidate targets for the proposed technique, with more complex circuits requiring further investigation in future studies.

Utilization of configurable approximate circuits requires determining when to transfer to approximate mode. An aging sensor is utilized to predict BTI aging, and as shown in Fig. 4(a), the aging sensor is attached to the output of the critical path of the circuit in order to predict aging-induced timing violations. Once the aging sensor predicts a timing error, the target circuit is configured to approximate mode and remains there lifetime of the circuit. The proposed method has two phases: testing phase and operating phase, and the phases switch periodically. During the testing phase, the aging sensor verifies whether operation of the target circuit is safe. Otherwise during the operating phase, the sensor is turned off and the configuration never changes. For example, each time the power is turned on (start up), a simple logic diagnosis is performed, with injection of the diagnosis that activates the critical path. If the aging sensor predicts timing violations, it configures the target circuit to approximate mode. It depends on a design decision how the configuration signals are generated. The generator may be implemented as a dedicated circuit or as a part of a diagnostic program. Other schemes are also acceptable. Because the aging sensor only works intermittently (for example, only at start up), it is expected to exhibit negligible power overhead.

Although several aging sensors [4], [13], [19], [20] could be adopted for this method, we used Canary FF. As shown in Fig. 4(b), each FF (called main FF) is augmented with a delay element and a redundant FF (called shadow FF), with the shadow FF working as a “canary in a coal mine” to predict timing errors. The shadow FF recognizes a timing error before the main FF, with this error predicted by an XOR gate that compares values held in the main and shadow FFs. Therefore, Canary FF works as the aging sensor and configures the target circuit. However, the open issue with Canary FF is that it can inaccurately predict errors. Although it is expected that the main FF consistently harbors correct values, in rare instances, it can still inaccurately predict a timing error. If a sudden change in critical-path delay occurs; thereby resulting in a violation, it is possible that a timing error will occur in both the main and shadow FFs, which could result in disappearance of the timing error.

As mentioned, the BTI aging is observed as a timing violation, which can be predicted by Canary FF. Notably, Canary FF consistently predicts these situations correctly, because in the proposed method the described unique situations never occur. First, both approximate and exact modes are implemented considering the worst scenarios. The target is designed based on worst-case design method. Any relaxed design methods such as better-than-worst-case design method [24] and typical-case design method are not chosen. Hence, a single critical path is deterministically identified. Second, the diagnosis is designed to not cause sudden changes in active paths. By relying on this technique, the circuit works correctly with only a slight loss in quality, even after aging affects the circuit delay, resulting in increases in the lifetime of LSI devices.

Figure 5 shows a proof-of-concept of the technique.

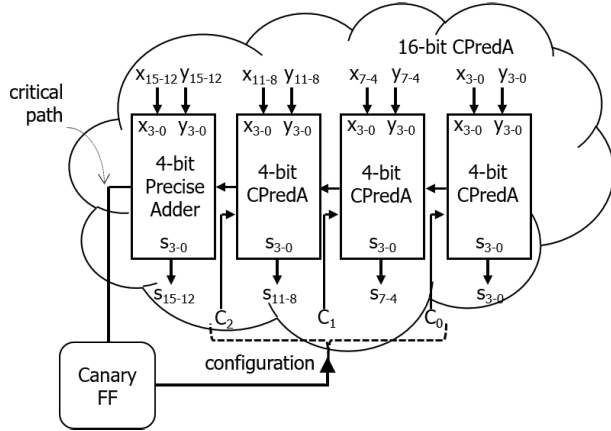


Fig. 5 CPredA and canary FF.

Canary FF is attached to a 16-bit CPredA. When logic diagnosis is conducted on CPredA and Canary FF predicts an aging-induced timing violation, CPredA is configured to one of approximate modes:  $\{C_2, C_1, C_0\}$  is  $\{1, 1, 0\}$ ,  $\{1, 0, 0\}$ , or  $\{0, 0, 0\}$ . When a timing violation is predicted, it is enough to choose next level for approximate configuration. For example, upon detected in the configuration set of  $\{1, 1, 0\}$ , the configuration set should change into  $\{1, 0, 0\}$ . Because its critical-path delay improves as a result, the timing violation is avoided.

## 5. Experiments

### 5.1 Methodology

To demonstrate the effectiveness of the proposed technique, we evaluated the effects of timing violations on an image-sharpening circuit [21] (called SHARP in this paper). An input image,  $I(i, j)$ , is smoothed by Gaussian kernel,  $G$ , and its smoothed image,  $S_m(i, j)$ , is obtained, as follows:

$$S_m(i, j) = \frac{1}{256} \sum_{k=-2}^2 \sum_{l=-2}^2 G(k+2, l+2) \cdot I(i+k, j+l)$$

$$G = \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

The sharpened image,  $S_h(i, j)$ , is generated as  $S_h(i, j) = \{I(i, j) - S_m(i, j)\} + I(i, j) = 2 \cdot I(i, j) - S_m(i, j)$ . Every multiplication emerging in the equation is replaced by shift operations and additions, with division also replaced with a shift operation. Every addition is then processed by a 16-bit RCA, CMA, or CPredA, and the smoothing module in SHARP is implemented using a tree structure. SHARPs utilizing RCA, CMA, and CPredA are called SHARP-rca, SHARP-cma, and SHARP-cpreda, respectively, in this paper.

SHARP was implemented by Verilog HDL and syn-

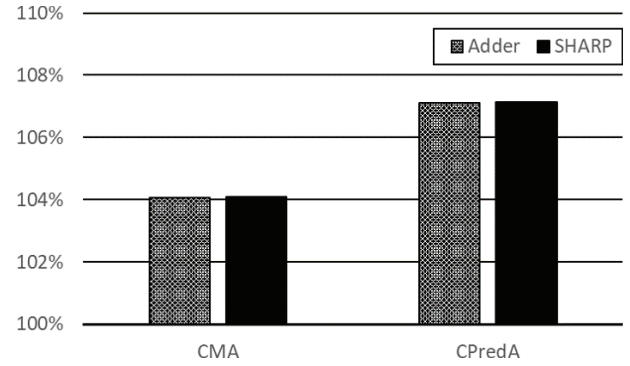


Fig. 6 Normalized circuit area.

thesized by Synopsys Design Compiler with Nangate 45-nm open-cell library [22]. The default options with no special optimizations were applied on compilation. Each cell in the generated netlist was annotated by its associated standard delay format (SDF) file and then simulated by Synopsys VCS. Increase in aging-induced delays were mimicked by tuning *scale\_factors* to the annotation. Six gray (boat, f16, goldhill, house, lena, and pepper) and three color (f16, lena, and pepper) images (512×512 bitmaps, 8-bit pixels) were used as inputs. The metric used to evaluate the qualities of the processed images was peak signal-to-noise ratio (PSNR), which was calculated, as follows:

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

$$MSE = \frac{1}{x \cdot y} \sum_{i=0}^{x-1} \sum_{j=0}^{y-1} [P(i, j) - A(i, j)]^2$$

where  $MAX_I$ ,  $P(i, j)$ , and  $A(i, j)$  are the maximum, accurate, and approximate values of each pixel, respectively, and  $x$  and  $y$  are the image dimensions.

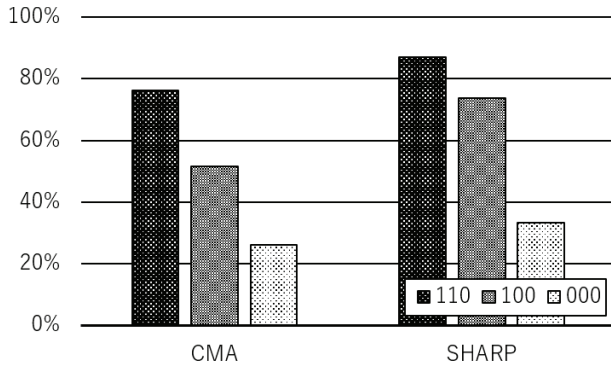
### 5.2 Results

Figure 6 shows the circuit areas of the 16-bit approximate adders and the SHARPs, normalized to those of the accurate ones (RCA and SHARP-rca), respectively. Two bars on the left in the figures show the areas of the 16-bit CMA and the SHARP-cma, with the two bars on the right showing the areas of the 16-bit CPredA and the SHARP-cpreda. For each group of two bars, the left one describes the area of the adder, with the right one describing the area of the SHARP. Overheads on circuit area were <8%. The dedicated circuit that generates the configuration set was also implemented and synthesized to evaluate its area overhead. As expected, it is very small and 0.57% and 0.55% for SHARP-cma and SHARP-cpreda, respectively.

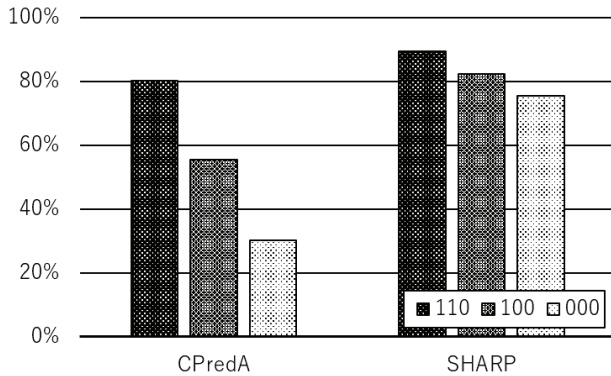
Figure 7(a) shows the critical-path delays of a single CMA and SHARP-cma for different configurations. Similarly, Fig. 7(b) shows the critical-path delays of a single 16-bit CPredA and SHARP-cpreda for different configurations. In both cases, three bars on the left in the figures show the

**Table 1** PSNR (dB).

+ $\Delta$ delay	boat	f16	goldhill	houses	lena	pepper	f16	lena	pepper
SHARP-rca									
10%	48.5	47.5	43.8	46.6	41.0	45.6	50.4	59.6	45.7
15%	40.6	40.2	36.9	37.6	35.7	38.8	41.6	45.8	39.2
20%	34.7	34.1	31.5	31.9	30.7	33.4	35.2	38.4	33.5
25%	30.2	29.3	27.6	27.6	26.8	29.1	14.3	32.7	12.2
30%	26.7	26.6	25.4	25.0	24.8	26.7	14.4	12.3	12.2
SHARP-cma									
0~15%	50.6	50.7	50.7	50.9	50.7	50.7	50.7	51.8	51.0
15~36%	23.8	24.0	23.8	24.3	23.9	24.0	24.2	24.1	24.4
SHARP-cpreda									
0~12%	55.4	55.5	55.4	55.6	55.4	55.5	55.5	56.0	55.7
12~22%	30.8	30.4	30.5	30.6	30.7	30.6	30.6	30.8	31.0
22~33%	12.3	13.2	12.4	13.1	12.4	12.5	13.3	12.8	13.4



(a) CMA and SHARP-cma



(b) CPredA and SHARP-cpreda

**Fig. 7** Normalized circuit delay.

delays of the 16-bit approximate adders, with the three bars on the right showing the delays of the SHARPs. For each group of three bars, those from left to right describe delays in the configuration sets  $\{1,1,0\}$ ,  $\{1,0,0\}$ , and  $\{0,0,0\}$ , respectively, normalized to that of the configuration set  $\{1,1,1\}$ . As expected, the approximate circuits showed smaller delays, and the scalability of both SHARPs was smaller than that of the approximate adders, because the tree structure was not reconfigured. Additionally, the signal propagation in the tree did not change, even when the carry propagation in each adder was reduced. In the case of SHARP-cma, margins of 15%, 36%, and 201% in delay were obtained for the three configurations, respectively, whereas for SHARP-

cpreda, margins of 12%, 22%, and 33% in delay were obtained for the three configurations, respectively. The reason why the scalability is lower in SHARP-cpreda than in SHARP-cma is that the critical path delay is larger in the former than in the latter [18]. This suggests that both SHARP-cma and SHARP-cpreda would tolerate 10-year aging for the configuration  $\{1,0,0\}$ .

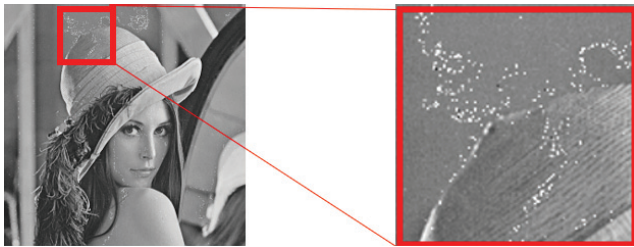
Table 1 summarizes how the PSNR is diminished as the aging-induced delay increased. The first column shows delay increase due to aging ( $\Delta$ delay). Because the approximate adders choose their appropriate configuration according to  $\Delta$ delay, PSNRs of SHARP-cma and SHARP-cpreda change stepwise. By contrast, the PSNR of SHARP-rca changes continuously. The remaining nine columns indicate PSNRs of the images. Notably, the accurate SHARP-rca consistently caused timing violations upon increases in delay. By contrast, the approximate SHARP-cma and SHARP-cpreda never caused violations following selection of the appropriate configurations. SHARP-cma selected the configuration sets  $\{1,1,0\}$  and  $\{1,0,0\}$  when  $\Delta$ delay was 0~15% and 15~36%, respectively. SHARP-cpreda selected the configuration sets  $\{1,1,0\}$ ,  $\{1,0,0\}$  and  $\{0,0,0\}$  when  $\Delta$ delay was 0~12%, 12~22%, and 22~33%, respectively. It is observed that SHARP-cma or SHARP-cpreda generated images, which had better quality in PSNR than SHARP-rca did, when  $\Delta$ delay was  $\leq 15\%$ . It is also observed that images generated by SHARP-cpreda had comparable quality with those generated by SHARP-rca does when  $\Delta$ delay was  $\leq 22\%$ . The PSNR values were  $>30$  dB, which is regarded as a typical value in lossy image compression [23]. These results suggest that both SHARP-cma and SHARP-cpreda would tolerate 10-year aging for a configuration set of  $\{1,0,0\}$ . The difference in the dependence of PSNR on  $\Delta$ delay between SHARP-cma and SHARP-cpreda is due to their characteristics in critical-path delay and accuracy. While SHARP-cma has smaller critical-path delay than SHARP-cpreda does, the former is less accurate than the latter [18].

Figure 8 shows the sharpened images. From left to right, the first and second columns show the original and accurately sharpened images, respectively. The remaining columns show the images processed under a condition where the aging-induced  $\Delta$ delay is 20%. The third col-





**Fig. 8** Original, accurately processed, SHARP-rca, -cma, and -cpreda Images.



**Fig. 9** Enlarged view of missing pixels.

umn shows images erroneously processed by the accurate SHARP-rca. Note that the sharpened images generated by the accurate SHARP-rca were inaccurate due to the occurrence of violations. The fourth and fifth columns show images correctly processed by the approximate SHARP-cma and SHARP-cpreda, respectively. The images in the second, fourth, and fifth columns are nearly visually indistinguishable, indicating that the configurable approximation tolerated aging-induced timing violations. Regarding the third column, it is difficult for human eyes to recognize significant quality loss in the small images. Figure 9 shows an enlarged view of a part of lena, revealing numerous missing pixels (white dots), which are not missing from the approximated image. This is the reason why the PSNRs of SHARP-rca are not very good.

## 6. Conclusions

This paper described exploitation of configurable approximate computing to compensate for aging-induced timing violations. A target circuit was implemented to harbor both exact and approximate modes, with the delay of the latter smaller than that in the former. We used Canary FF as an aging sensor to predict violations and configure the target to approximate mode in order to allow toleration of violations. Simulation results showed that the proposed technique tolerated 10-year aging.

Future research will include designing approximate complex circuits and adopting configurable approximations to other reliability problems.

## Acknowledgments

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

## References

- [1] T. Sato and T. Ukezono, "Tolerating aging-induced timing violations via configurable approximations," *Global Conference on Consumer Electronics*, 2019. doi: 10.1109/GCCE46687.2019.9015592
- [2] H. Amrouch, V.M. van Santen, and J. Henkel, "Estimating and optimizing BTI aging effects: From physics to CAD," *International Conference on Computer-Aided Design*, 2018. doi: 10.1145/3240765.3243475
- [3] M. Yabuuchi and K. Kobayashi, "Circuit characteristic analysis considering NBTI and PBTI-induced delay degradation," *International Meeting for Future of Electron Devices*, Kansai, 2012. doi: 10.1109/IMFEDK.2012.6218587
- [4] T. Sato and Y. Kunitake, "Canary: A variation resilient FF to eliminate design margin for energy reduction," *IPSS J.*, vol.49, no.6, 2008 (in Japanese). <http://id.nii.ac.jp/1001/00009556/>
- [5] Y. Kunitake, T. Sato, H. Yasuura, and T. Hayashida, "Possibilities to miss predicting timing errors in canary flip-flops," *54th International Midwest Symposium on Circuits and Systems*, 2011. doi: 10.1109/MWSCAS.2011.6026656
- [6] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," *18th European Test Symposium*, 2013. doi: 10.1109/ETS.2013.6569370
- [7] H. Amrouch, B. Khaleghi, A. Gerstlauer, and J. Henkel, "Towards aging-induced approximations," *54th Design Automation Conference*, 2017. doi: 10.1145/3061639.3062331
- [8] J. Kim, H. Kim, H. Amrouch, J. Henkel, A. Gerstlauer, and K. Choi, "Aging gracefully with approximation," *International Symposium on Circuits and Systems*, 2019. doi: 10.1109/ISCAS.2019.8702120
- [9] B. Boroujerdian, H. Amrouch, J. Henkel, and A. Gerstlauer, "Trading off temperature guardbands via adaptive approximations," *36th International Conference on Computer Design*, 2018. doi: 10.1109/ICCD.2018.00039
- [10] R. Ye, T. Wang, F. Yuan, R. Kumar, and Q. Xu, "On reconfiguration-oriented approximate adder design and its application," *International Conference on Computer-Aided Design*, 2013. doi: 10.1109/ICCAD.2013.6691096
- [11] S. Angizi, Z. He, R.F. DeMara, and D. Fan, "Composite spintronic accuracy-configurable adder for low power digital signal processing," *18th International Symposium on Quality Electronic Design*, 2017. doi: 10.1109/ISQED.2017.7918347
- [12] A.M. Hassani, M. Rezaalipour, and M. Dehyadegari, "A novel ultra low power accuracy configurable adder at transistor level," *8th International Conference on Computer and Knowledge Engineering*, 2018. doi: 10.1109/ICCKE.2018.8566643
- [13] T. Nakura, K. Nose, and M. Mizuno, "Fine-grain redundant logic using defect-prediction flip-flops," *International Solid-State Circuits Conference*, 2007. doi: 10.1109/ISSCC.2007.373464
- [14] K. Yano, T. Hayashida, and T. Sato, "Improving timing error tolerance without impact on chip area and power consumption," *15th International Symposium on Quality Electronic Design*, 2013. doi: 10.1109/ISQED.2013.6523638
- [15] Y. Kunitake, T. Sato, and H. Yasuura, "Short term cell-flipping technique for mitigating SNM degradation due to NBTI," *IEICE Trans. Electron.*, vol.E94-C, no.4, pp.520–529, 2011. doi: 10.1587/transele.E94.C.520
- [16] V.K. Chippa, S.T. Chakradhar, K. Roy, and A. Raghunathan, "Analysis and characterization of Inherent application resilience for approximate computing," *50th Design Automation Conference*, 2013. doi: 10.1145/2463209.2488873
- [17] T. Yang, T. Ukezono, and T. Sato, "A low-power configurable adder for approximate applications," *19th International Symposium on Quality Electronic Design*, 2018. doi: 10.1109/ISQED.2018.8357311
- [18] T. Sato, T. Yang, and T. Ukezono, "Trading accuracy for power with a configurable approximate adder," *IEICE Trans. Electron.*, vol.E102-C, no.4, pp.260–268, 2019. doi: 10.1587/transele.2018CDP0001
- [19] J. Abella, X. Vera, O. Unsal, O. Ergin, and A. Gonzalez, "Fuse: A technique to anticipate failures due to degradation in ALUs," *13th International On-Line Testing Symposium*, 2007. doi: 10.1109/IOLTS.2007.34
- [20] M. Agarwal, B.C. Paul, M. Zhang, and S. Mitra, "Circuit failure prediction and its application to transistor aging," *25th VLSI Test Symposium*, 2007. doi: 10.1109/VTS.2007.22



- [21] R. Szeliski, *Computer Vision - Algorithms and Applications*, Springer 2011. doi: 10.1007/978-1-84882-935-0
- [22] Silvaco Inc., “PDK 45 nm Open Cell Library,” [https://www.silvaco.com/products/nangate/FreePDK45\\_Open\\_Cell\\_Library/](https://www.silvaco.com/products/nangate/FreePDK45_Open_Cell_Library/) [Accessed on Nov. 18, 2019]
- [23] D. Bull, *Communicating Pictures - A Course in Image and Video Coding*, Academic Press, 2014.
- [24] T. Austin, V. Bertacco, D. Blaauw, and T. Mudge, “Opportunities and challenges for better than worst-case design,” 10th Asia and South Pacific Design Automation Conference, 2005. doi: 10.1109/ASPDAC.2005.1466113



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



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