IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 8, 2019, accepted August 29, 2019, date of publication September 9, 2019, date of current version September 30, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2940083

Correction of Bit-Aliasing in Memristor-Based Physically Unclonable Functions With Timing Variability

HA-PHUONG NGUYEN[1], THE-NGHIA NGUYEN[2], YEONG-SEOK SEO[1], DOSAM HWANG[1], AND DONGHWA SHIN[2], (Senior Member, IEEE)

[1]Department of Computer Engineering, Yeungnam University, Gyeongsan 38541, South Korea
[2]Department of Smart Systems Software, Soongsil University, Dongjak-Gu 06978, South Korea

Corresponding author: Donghwa Shin (donghwashin@soongsil.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) under Grant NRF-2018R1D1A1B07049445.

ABSTRACT The dependence between the write time and process variation of a memristor was investigated as a candidate physically unclonable function (PUF). Such write-time-based approach requires exact timing control of the programming pulse to achieve well-balanced results as the source of randomness. However, exact timing control requires precise hardware, such as a high-frequency timer. Furthermore, the effects from other variability sources, such as the voltage and temperature, may degrade the quality of the PUF device operation. In this study, we introduce a method to enhance the bit-aliasing of a memristor-based PUF with write time variability considering the timing error. By exploiting the non-volatility and bidirectional operation of the memristor, the proposed method attempts to correct the timing error using extra programming pulses. The experiment results reveal that the proposed method reduces the bit-aliasing error by 75.44%.

INDEX TERMS Physically unclonable function, bit-aliasing, memristor, performance metrics, hamming distance.

I. INTRODUCTION

Memristor have become a potential technology for use in memory devices owing to their high density, low power, non-volatility, and nanoscale capability [1], [2]. A memristor can be regarded as an electrical element that is able to retain the internal resistance according to the history of its applied voltage and current even when power is removed [3]. The resistance states can be switched between them by applying a voltage with a proper magnitude and duration. Owing to this unique property, a memristor is a promising candidate for many up-to-date applications such as non-volatile memory.

In general, a memristor has been utilized in nanoscale systems, and thus, a small normal variation is likely to have a considerable impact on the parameters and behaviors of a device. When the device is utilized as a memory cell, the variations in the read and write times of the memristor are affected by the variability in thickness and area. Hence, these variations affect the resistance related to the supplied voltage and current across the device [4]. Furthermore, the supply voltage for each device is also variable according to the physical connection through the power distribution network. The dependence between the domain-wall mobility and temperature is also a well-known source of variability.

For ordinary memory applications, the effects of the variability should be suppressed to lower than a certain level to guarantee the timing requirement. However, at the same time, such effects of the variability can be utilized for specific applications. Several previous studies have attempted to use memristor as a temperature sensor based on the temperature-dependent device characteristics. After a constant-voltage pulse is applied to the memristor cell for a certain period, the resistance is measured using a constant voltage/current source and a voltage-to-digital converter circuit [5]. Another approach attempted to measure the variation of the write time to obtain the temperature [6].

The associate editor coordinating the review of this manuscript and approving it for publication was Ho Ching Iu.

A physically unclonable function (PUF) is a physical entity with features that are practically impossible to duplicate. Such features are usually originated from random physical factors introduced during the manufacturing process of the individual device. These factors are expected to be *unpredictable and uncontrollable*, making the function *unclonable*.

The dependence between the write time and process variation of the memristor has also been investigated as a candidate of the PUF [7]–[9], [14], [20]. In [7], the minimum duration of the write pulse was applied to each memristor cell to achieve a 50% rate of change in the cell value. By contrast, in [20] the write pulse current was modulated to obtain the statistical variation. In [9] the geometrically distributed resistance over the nano crossbar architecture was exploited to obtain randomness. Finally, a new logic cell based on the memristors for PUF applications was proposed in [8].

Among these approaches, a write-time-based approach is a simple and straightforward solution that can be implemented without much overhead being integrated into an ordinary memory structure. The challenging process of a write-time-based approach can be overcome through a digital method without specialized logic or analog circuits. However, it still requires exact control of the write current pulse timing or programming current to achieves well-balanced results as the source of randomness. The exact timing control requires a high-frequency timer, and, furthermore, the effect from other variability sources, such as the voltage and temperature, may degrade the quality of the PUF device operation.

In this study, we investigated a correction method of the results with a timing error. We aimed at reducing the effect of timing error on the write time variation-based random number generation for PUF. Specifically, the proposed method obtains multiple samples when meeting biased results owing to the timing error by exploiting the non-volatility of the memristors. Note that this work is an attempt to reduce the timing error on bit-aliasing of the generated random number in the existing PUF structure by using the features of the memristor cells in the array. The generation process of final PUF with extra helper data and its other quality measures are beyond this work.

We evaluated the effectiveness of the sampling method with respect to the control of the voltage and current with a physical model of the memristor in a crossbar structure. The experiment results show that the proposed value achieves enhanced bit-aliasing even with the much lower operating frequency of the sampling unit. Because the proposed method can basically be applied under any circumstances, it is also expected to be able to mitigate the effects of other variabilities such as variations in temperature and supply voltage.

The rest of this paper is organized as follows. Section III shows the memristor-based PUF using write-time control. Section IV details the effect of the timing error correction on bit-aliasing. Section V describes the experiment results of our proposed method. Section VI provides some concluding remarks regarding this research. Finally, section Appendix describes the details of models and metrics.

II. RELATED WORK

Memristors have become an emerging research topic owing to its numerous advantages as an intrinsic high density, fast access speed, and good energy efficiency [1]. A memristor can be regarded as an electrical switch that is able to retain the internal resistance according to its history of applied voltage and current even when the power is removed [3]. The variability in thickness and area are translated into variations in the read and write times of the memristor when using the device as a memory cell. The effects of the variation in thickness and area depend on the history of the voltage applied across the device within a certain time period [4].

A PUF is applied to certain devices to increase their security, for example, an arbiter PUF, ring oscillator PUF, and a CMOS-memristor hybrid ring oscillator PUF [10]. A PUF can be used to mitigate or limit piracy, counterfeiting, and side-channel attacks providing a unique hardware signature or identification [7]. A PUF has a relation with a hardware-based security primitive, and a unique fingerprint for each PUF is available, which can be used for security purposes [11].

PUF devices are often regarded as a desirable source of randomness for cryptography and security applications thanks to their uniqueness when connected to a device. Such applications can use the PUF enrollment process as the source of entropy. PUFs based on a process variation of an integrated circuit, such as SRAM and non-volatile memory (NVM), have been studied thus far [21]. NVM-PUFs are often regarded as vulnerable for higher security applications owing to their non-volatility. However, the smaller footprint and higher reliability of NVM non-volatile memory make it attractive for smaller applications with lower security requirements [2].

In [2], an approach is proposed in which the entropy sources in the PUF enrollment process, i.e., the magnitude of the analog variations, are simulated and digitized as multibit digital information.

Such secret information supports the regeneration processes and is stored in non-volatile memory, and as such a memristor represents a certain vulnerability and prevents the NVM-PUFs from achieving a high-security application. However, the small footprint and high reliability in storing such information in non-volatile memory are more attractive for lower-security and small form-factor applications.

Researchers have utilized bit-aliasing, uniformity, and uniqueness to evaluate PUFs [7], [12], [13], [15], [16]. Bit-aliasing is a statistical performance measure used to analyze a memristor, and estimates the rate of bit values across the responses. The ideal value of averaged bit-aliasing is 50%. If the value of the bit aliasing is close to 50%, then we can expect that the responses will be well-distributed within a statistical space. We can determine whether a PUF has been easily counterfeited through bit-aliasing.

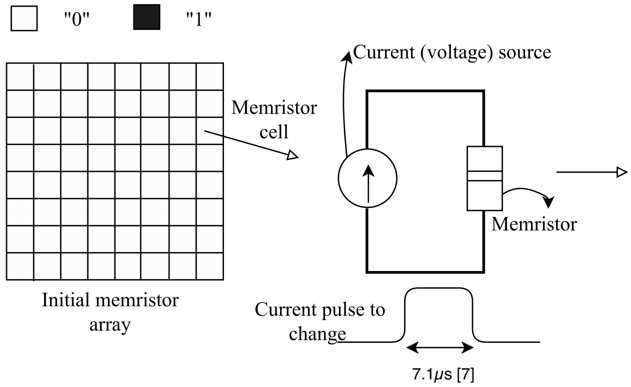

FIGURE 1. Enrollment process of the memristor-based PUF.

TABLE 1. Comparison of bit-aliasing over different models.

Author	PUF type	Bit-aliasing
Maiti et al. (2013)	APUF RO PUF	19.57% [24], [25] 50.56% [24], [25]
Loong et al. (2016)	RO PUF	51.43% [13]
Chatterjee et al. (2016)	APUF	50.60% [28]
Sehwag et al. (2016)	TV-PUF	49.96% [26]
Uddin et al. (2018)	XbarPUF	50.25% [29]
Rose et al. (2013)	**M-PUF**	**49.99%** [7]

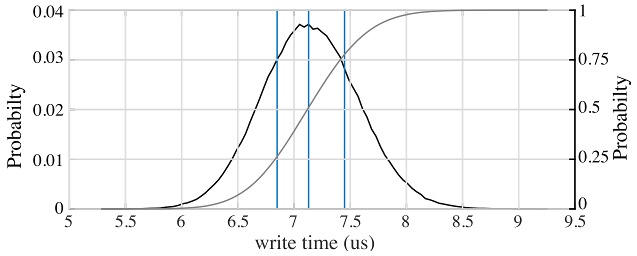

FIGURE 2. Normally distributed t_W and timing error sensitivity.

III. MEMRISTOR-BASED PUF USING WRITE-TIME CONTROL

Fig. 1 illustrates the challenge-response process of the write-time-based memristor PUF introduced in [7]. First, the values of the target cell array are set to the same initial value (initial memristor array). The current (voltage) source affects the memristor cells, and the current pulse will change the state of the memristor cell from 0 to 1. A dedicated controller stops the programming process within a certain timing to achieve the required randomness.

In [7], to obtain the exact duration of a write pulse, a Monte Carlo simulation has been conducted in advance to produce a histogram of the minimum SET time for a TiOx memristor with a fixed voltage and a thickness varying by 10%. The expected minimum SET time was approximately $7.1\mu s$ to obtain an even distribution of the states and maximize the randomness from the responses (PUF response with balanced bit-aliasing). Table 1 shows the bit-aliasing of the memristor-based PUFs in the literature. In [7], the proposed M-PUF shows the highest bit-aliasing value of 49.99%. However, it assumes exact control of timing that is difficult to be achieved in reality. Note that we are focusing on the mitigation of timing error in memristor-based PUF in this work.

For the conventional approaches, it is important to obtain the exact duration of the write pulse and control the pulse during the enrollment to achieve better bit-aliasing values. We need a resolution of at least 0.1 μs for the measurement and control with the 7.1 μs programming pulse length used in [7]. However, the timing measurement has various sources of errors. The major sources of a measurement error for an electronic counter are generally classified into the following four categories: a ± 1 count error, time base error, trigger error, and systematic error [22]. Even if we ignore the effects of the latter three error sources, the ±1 count error cannot be avoidable. The implied timing error for a 10 MHz counter is $\pm 1\mu s$. This may result in a serious degradation of bit-aliasing in the write-time-based memristor PUF because the total write time is less than 10 μs.

Furthermore, the statistical characteristics of the cells are likely to be unfriendly for such exact control of timing. The statistical variation of the write time in the memristor cells originates from the variation in the cell thickness according to the model presented in Appendix. If the thickness of the cell, and in turn, the write time, are normally distributed, the 50% point is the most vulnerable point for the timing error.

Fig. 2 depicts the cumulative density function (CDF) of the cell change and the probability density function (PDF) as the derivative of the CDF, which is derived from the statistics depicted in Fig. 12. The same amount of error in the timing results in the largest error the CDF point of 0.5.

IV. CORRECTION OF TIMING ERROR EFFECT ON BIT-ALIASING

A. EFFECT OF TIMING ERROR ON BIT-ALIASING

Bit-aliasing can be defined as the proportion between bit 0 and bit 1. If its value is 0% or 100%, all values of the responses are 0 or 1, respectively. However, these cases do not result in high security owing to little information obtained from such responses, and because it is easy to create a fake

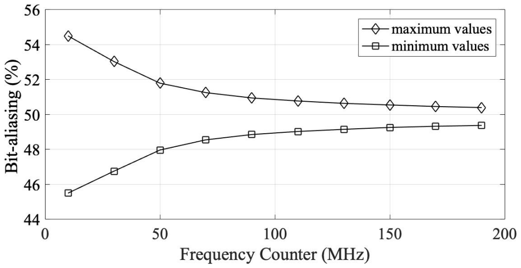

FIGURE 3. Minimum and maximum values of bit-aliasing with ±1 counter cycle timing error at different frequencies.

PUF. For a bit-aliasing of 75%, such responses are created from different PUFs, indicating that the performance of bit 1 is 75%, which implies that the performance of bit 0 is 25%. Detection of 25% bits is easier than 50%. Thus, bit-aliasing reaches as close to 50% as possible.

The following figure simulates bit-aliasing using a frequency counter. The time at which a bit-aliasing of 50% is obtained in this experiment is t = 7.137 μs at a temperature of 300 K. When the frequency counter increases, the time is reduced, which is a deviation of time t. With the maximum line, the time is added to the deviation, with the minimum line, the time is subtracted from the deviation. From this, we determine the bit-aliasing based on frequency. The deviation is the timing error. The timing error changes the bit-aliasing according to a small or large value of the timing error.

We considered a case in which the frequency counter is 10 MHz, which is equivalent to 0.1 μs; in addition, the maximum value of the bit-aliasing value is 55.6370% (at 7.2 μs) and the minimum value is 46.3410% (at 7.1 μs). The ability for different PUF instances to produce nearly the same responses is extremely low if the bit-aliasing is close to 50%. According to Fig. 3, when the frequency counter is increased (the timing error decreases), the bit-aliasing is more close to the ideal value. The frequency counter is in inverse proportion with the write time of the memristor cells. To enhance the accuracy of the bit-aliasing, the bit-aliasing error is reduced using multiple samples (as presented in Section IV-B).

Note that the variability in the operating conditions, such as the temperature and supply voltage, may result in s similar error under bit-aliasing when we use a fixed timing control. When the temperature increases, the write time decreases, and thus, the time required to obtain 50% of the number of memristor cells also declines. In this study, we are focused on the management of the timing error during the challenging process for a memristor-based PUF. However, it is generally expected that the proposed method is basically applicable to any source of error including the temperature and supply voltage because the method attempts to correct an error after it occurs.

B. TIMING ERROR AND BIT-ALIASING

As we previously discussed, the value of the bit-aliasing for a PUF is expected to be 50%. To achieve 50% bit-aliasing, the conventional method attempts to stop the programming process at time t_i, which obtains 50% the bit-aliasing of the response r_i. This implicitly assumes that we have a sufficiently accurate method to control the programming process with the exact timing.

The variation in thickness during the semiconductor manufacturing process is often modelled through a normal (Gaussian) distribution [23]. In a normal distribution, the point required to obtain 50% bit-aliasing is located at the center of the CDF. Unfortunately, this is the most vulnerable point for the timing error because it is along the largest slope in the CDF. In other words, a conventional method attempts to control the process at the most difficult point structurally.

To achieve an accurate control, we can adopt a dedicated cell value recognition circuitry dedicated to generating the exact control signal of a programming pulse (or perhaps a combination of a comparator and an accumulator). Otherwise, we can attempt to stop the programming process at the pre-collected timing to achieve sufficient fairness in the bit-aliasing value. We need to equip a sufficiently higher frequency counter to enable an accurate timing control.

Both approaches incur cost overhead. The former is a costly solution in term of the area because it needs to deal with all different output patterns from each challenge using combinational logic, whereas the latter is power consuming as the counter frequency increases.

C. BIT-ALIASING ERROR CORRECTION CONSIDERING MEMRISTOR CHARACTERISTICS

It is expected that we can pre-determine the length of the programming pulse to obtain exactly 50% bit-aliasing with a certain statistical variability in thickness. In reality, however this requires fine-grain control of the timing and operating conditions such as the temperature. The logical value of a real device might differ from the expectation with the same length of pulse owing to the operating conditions. furthermore, the pulse itself may not be the same as expected.

There are two intuitive ways to deal with this problem: i) obtaining the value at the lower timing vulnerability or ii) fixing the value after the timing error occurs. For i), we investigate the possibility of reducing the bit-aliasing error by generating the final value as a combination of multiple samples in a given CDF. We can generate a 50% result by subtracting x% from x + 50%. We then evaluate the effectiveness of ii) through a statistical analysis. Fig. 11 shows the relation among the thickness, timing, and logical values. Considering ±1 count error, a negative timing error may result in a 1 to 0 error, and vice versa. Because the range of control is extremely close to the critical region, even a small timing error may result in a totally different logical value.

1) COMBINATION OF MULTIPLE SAMPLES

Unfortunately, owing to the intrinsic statistical features presented in Fig. 2, the combined errors are always expected to be larger than those of a single pulse. The differences

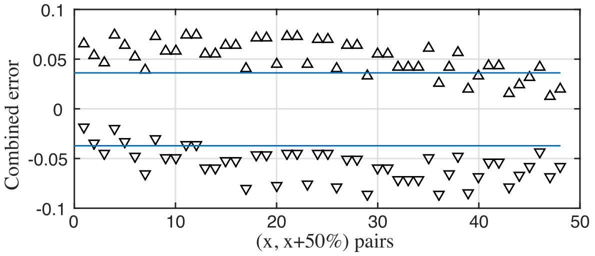

FIGURE 4. The combined error of sample pairs in normally distributed t_W.

in bit-aliasing originating from a timing error will be larger when we add values with an error. The values of the PDF, as a gradient of the CDF, can be regarded as a sensitivity of the timing error in a bit-aliasing error, and the shape of the PDF demonstrates why we cannot achieve a smaller error by manipulating multiple values. For instance, we can see that the values of the PDF, as a gradient of the CDF, at 25% and 75% (the outer blue vertical lines in Fig. 2) are higher than for half of 50% (the center blue vertical line in Fig. 2). We can reduce the timing error sensitivity of a bit-aliasing error by sampling the values in the left and right tail values. However, as we sample the values far from the average, we need to obtain another value in the pair at the point closer to the average. This effect countervails the benefits from the lower timing error sensitivity at the tail points.

Fig. 4 shows the expected maximum error of the sample pairs combined as (x%, x + 50%) with a counter frequency of 20 MHz. The blue lines indicate the minimum and maximum errors of a single pulse program with a ±1 counter error. As presented in Fig. 12, the timing error sensitivity in other values may be smaller than that of the 50% point, whereas the errors are combined and are finally larger than a single pulse case. A combined error tends to move the lower points as the value of x increases in the pair (x, x + 50%). As x increases, the timing error sensitivity of x increases, and that of x + 50% decreases. Therefore, the upper bound of the error decreases and the lower bound of the error increases.

2) CORRECTION AFTER SAMPLING USING INCREMENTAL AND REVERSE PROGRAMMING PULSE

The observation in the previous section shows that the combination of multiple samples likely extends the range of the error. It is not preferable to utilize multiple samples for the preservation of bit-aliasing. In this section, we attempt to correct a single sample considering the bit-aliasing result after the programming pulse. Differing from the other non-volatile memories, a memristor has symmetric characteristics in both programming directions, namely, 0 to 1 and 1 to 0. Thus, we can expect that a correction after the programming will not incur a significant effort during an operation. We first check the tendency of the error by using multiple consecutive pulses.

Fig. 5 shows the position of the domain wall by applying programming pulses with different sequences: i) a single pulse (Fig. 5(a)), ii) consecutive forward pulses (Fig. 5(b)), and iii) a forward pulse followed by a reverse pulse (Fig. 5(c)). Note that we use a constant voltage pulse. We then check the relation between their domain wall positions with different programming pulses.

We consider two cells with different thicknesses to consider the effects of the variability: a thinner cell with a faster programming time and a other thicker cell with slower programming time and a thicker cell with a slower programming time. We can create an intermediate logical state with two different cells. Specifically, when we apply the same length of the programming pulse in Fig. 5(a), the thicker cell may remain in its logical state of zero whereas the thinner cell will have already changed its state from 0 to 1.

Consequently, we can apply a programming pulse of 69 to 100 *ns* to obtain 50% bit-aliasing. This can be regarded as the timing control margin of the programming pulse. This timing margin is related to the relative domain wall positions. The difference in the domain wall positions continues increasing within this timing margin by the single pulse in Fig. 5(a).

Next, we apply consecutive multiple pulses and check the relative domain wall position, as shown in Fig. 5(b). We apply a write pulse with an amplitude of 1V and a duration of 69 *ns*, and then we apply several shorter pulses. The relative domain position shows the same aspect with some delays inserted

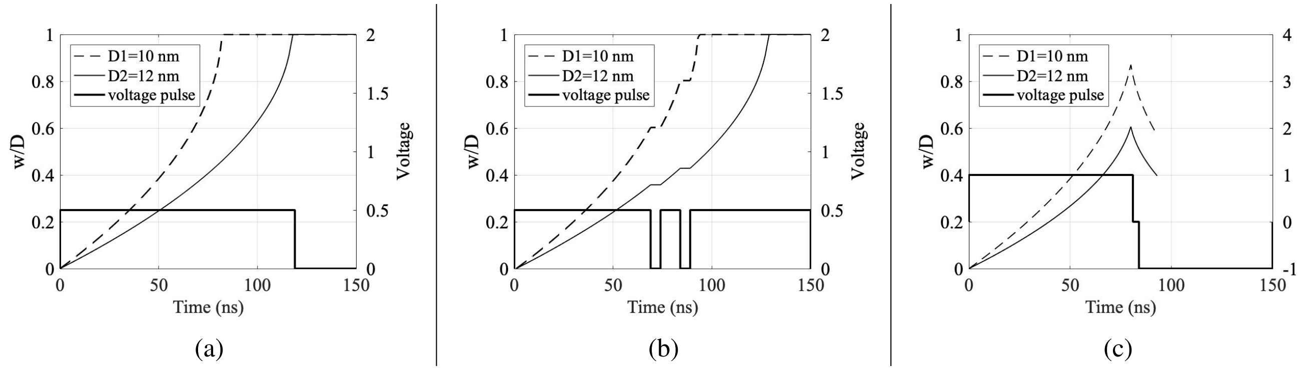

FIGURE 5. (a) Programmable pulse of memristor and correction after programming by (b) incremental programming in the same direction and (c) roll-back with reverse programming.

between the pulses. This is a theoretically reasonable result as long as the resistance maintains its characteristics even with an idle period of insertion.

Finally, we check the relative domain wall position with the programming pulse followed by a reverse pulse. As seen, the changing rate of the domain wall positions by the reverse pulse is symmetric to that by the forward pulse in Fig. 5(c). The result of the forward pulse followed by a reverse pulse can be equivalent to the shorter forward pulse. Therefore, we do not lose the timing margin even when we change the direction of the pulse as long as we maintain the equivalent pulse length at the same value.

Based on the observation, we propose a correction method that behaves as described in Fig. 6. When the value of the bit-aliasing is far less than expected (e.g., 50% with tolerance), the proposed correction method applies an incremental pulse to increase the number of written cells. By contrast, we apply the roll-back pulse to reduce the number of written cells to close to 50% when the value of the bit-aliasing is far greater than expected.

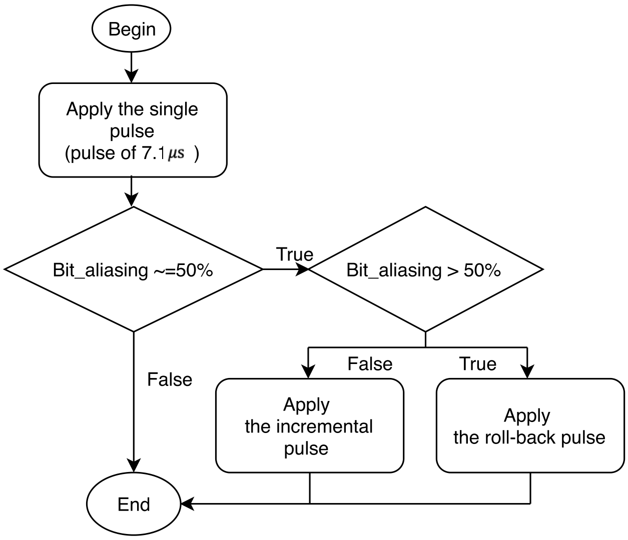
Begin
Apply the single pulse (pulse of 7.1μs)
Bit_aliasing ~=50%
True
Bit_aliasing > 50%
False
False
True
Apply the incremental pulse
Apply the roll-back pulse
End

FIGURE 6. Proposed bit-aliasing correction method by extra forward and reverse programming pulses.

V. EXPERIMENT

First, we examine the effects of the pulse order and kind. Fig. 7 shows the simulation results of different orders of forwarding and reverse pulse with a fixed voltage and current. We considered four cases in a simulation using two cells with a thickness of 10×10^{-9} and 12×10^{-9}.

In case 1, the input voltage is 1 V, the current is changed following the variability of the resistance, and we see that cell 1 has a greater thickness and will achieve a state of 1 earlier than cell 2 (the time at which cell 1 achieves $w/D \geq O_h$ (Fig. 11) is earlier than cell 2). After cell 2 is written, we apply a negative voltage, i.e., -1 V. Similar to case 1, case 2 also causes the same result when we apply an initial voltage of -1 V and change it to 1 V to change the state from 1 to 0. For cases 3 and 4, we apply a fixed current at 20 and -20 mA. Overall, the relative positions based on the voltage and current pulses in different directions are symmetric.

The results show that both of the forward-reverse and reverse-forward pulse orders reveal the same symmetric aspect of the relative domain wall position. The fixed

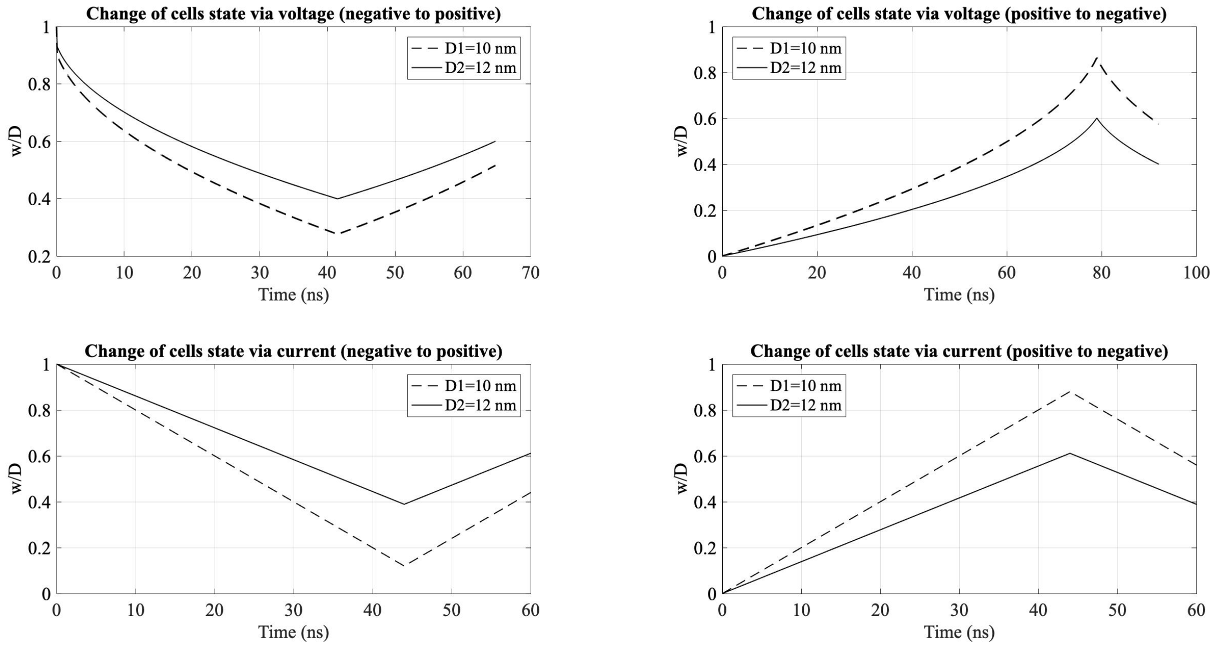

FIGURE 7. The state change of cells via the voltage and current pulses.

current pulse linearly changes the domain wall position overtime when the domain wall velocity is proportional to the applied current density. The fixed voltage pulse changes the domain wall position in an accelerated or decelerated manner when the current density induced domain wall velocity is inversely proportional to the resistance. Note that we chose the forward-inverse fixed voltage curve for the statistical evaluation of the PUF bit-aliasing in the following experiments.

Next, we statistically evaluated the proposed correction method. We populated 2,000 samples of a 16-bit PUF from the population presented in Fig. 12. We compared the bit-aliasing of the sample PUFs using a traditional method (single pulse) and the proposed method (using both an incremental pulse and a roll-back pulse individually). The pre-obtained timing value (7.1 μs for the population (Fig. 2)) followed by the incremental or roll-back pulses are applied using the correction method shown in Fig. 6.

Fig. 8 shows a detailed illustration when using a single pulse and the proposed pulse with a frequency counter of 10 MHz. In this experiment, 2,000 samples of 16-bit memristor PUFs were selected as the sample population to conduct the experiments. The bit-aliasing of a single pulse is plotted in grey in the figure, whereas that of the proposed pulse is plotted in black. Bit-aliasing of the proposed pulse is closer to the ideal value than that of a single pulse. The amount of bit-aliasing of the proposed pulse is distributed closer to the center region of 50% than the bit-aliasing of a single pulse. We conducted one more experiment on the effectiveness of the proposed pulse at the frequencies shown in Fig. 9. In this experiment, we used a sample of a 16-bit memristor PUF at different frequencies for both a single pulse and a proposed pulse. The proposed pulse clearly achieves better results. The proposed correction pulse decreases the bit-aliasing error by 75.44%. Besides, we consider the energy overhead of the model when the timing error occurs. Energy overhead is calculated as follows:

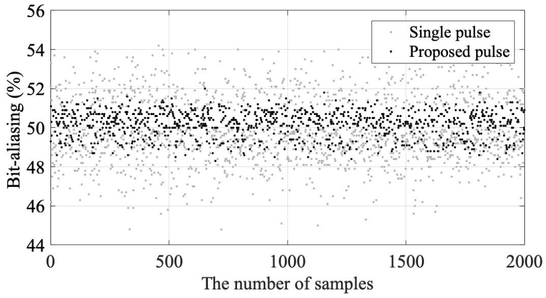

FIGURE 8. Effectiveness of additional pulses at the frequency of 10 MHz.

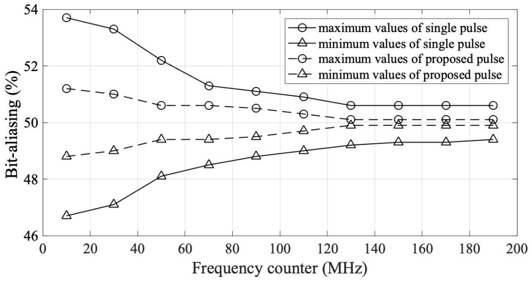

FIGURE 9. Effectiveness of additional pulses at different frequencies.

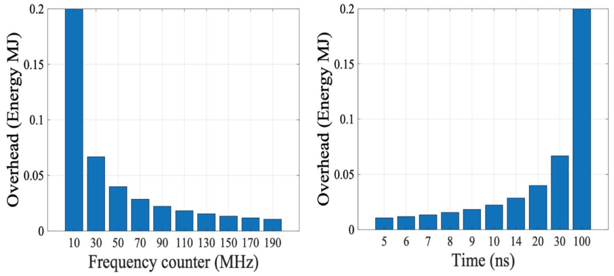

FIGURE 10. Overhead of the model according to frequency and time.

$$Energy\_overhead = U \times I \times T = U \times I \times \frac{1}{f} \qquad (1)$$

Fig. 10 shows the change of energy overhead according to frequency and time. Energy overhead decreases when the frequency counter increases, but energy overhead increases when the time increases. The maximum value of the energy overhead only achieved an extremely small value of 0.2 MJ at the time of 100 ns.

VI. CONCLUSION

A correction method of a PUF response for a write-time-based memristor PUF was proposed to reduce the bit-aliasing error originating from the timing error. In a previously introduced write-time-based PUF, we need precise control of the programming pulse to achieve a balanced value of the bit-aliasing, which is one of the major performance metrics of PUF reliability.

The proposed method attempts to correct the PUF response by utilizing a non-volatility and bidirectional operation of the memristors. We investigated the feasibility of the correction method, and checked its effectiveness through a statistical experiment. We generated 2,000 PUF samples with variable thicknesses for a Monte-Carlo simulation. The experiment results show that the proposed method can reduce the bit-aliasing error by 75.44%.

This study attempted to enhance the reliability of a memristor-based PUF based on the features of a memristor. In a future study, the practical aspect of the proposed method including the energy and cost will be analyzed and enhanced.

APPENDIX

MODELS AND METRICS

A. MEMRISTOR MODEL WITH VARIABILITY

In [6], [19], the relationship between programming time and the ambient temperature of crossbar structure memristors has been exploited in order to estimate temperature from the temperature-dependent variability of a memristor. In our work, we use memristor's physical model proposed in [6] to

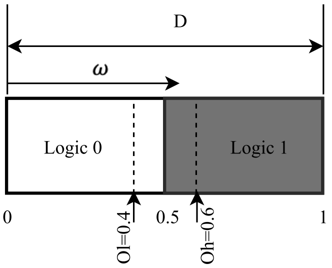

FIGURE 11. Memristor output level.

evaluate the effectiveness of our method. The overall behavior of a memristor can be described in the following equations:

$$\frac{dx(t)}{dt}) = \frac{\mu_I R_{on}}{D^2} i_m(t),$$
$$v_m(t) = [R_{on}x(t) + R_{off}(1 - x(t))]i_m(t), \tag{2}$$

where R_{on} and R_{off} are considered as resistances of doped and undoped regions respectively. $x = w/D$ is a state variable of the memristor and defined by the ratio of the doped region (w) and thickness (D) of the memristor. Ideally, a memristor is at logic 0 or 1 when $x = 0$ or $x = 1$, respectively. However, in reality, under the effect of noise injections, a safety margin was proposed in [30] for each logic output: $0 \le x \le O_l \le 0.5$ for logic 0, and $0.5 \le O_h \le x \le 1$ for logic 1. Fig. 11 illustrates the case where $O_l = 0.4$ and $O_h = 0.6$. $v_m(t)$ and $i_m(t)$ are the voltage and current applied to the memristor. The temperature-dependent mobility, $\mu_I(T)$, and $D_I(T)$ relations are denoted as follows [6]:

$$\mu_I(T) = \frac{q_I D_I(T)}{k_B T},$$
$$D_I(T) = fa^2 exp(\frac{-E_A}{k_B T}), \tag{3}$$

where k_B is a Boltzmann constant, q_I is the ion charge, f is the jump frequency, a is the crystal geometry, E_A is the ion activation energy, T is the temperature. From (2) and (3), we can derive an equation for the write time of the memristor cells, shown in (4) below. The write time is related to temperature, and if the temperature increases, the write time will decrease [6].

$$t_{write} = \frac{D^2}{\mu_I(T)v_w}(\frac{r_1 - 1}{2}(x_0^2 - x_f^2) + (r_1 + r_2)(x_f - x_0)), \tag{4}$$

where parameters r_1 and r_2 depend on R_{off}, R_{on}, and $R_{pulldown}$, and are calculated as follows: $r_1 = R_{off}/R_{on}$ and $r_2 = R_{pulldown}/R_{on}$.

The write time is usually modeled according to a Gaussian distribution [18], [19] similar to other physical parameters used in the semiconductor manufacturing process. Fig. 12 shows the variation in write time with respect to the normally distributed thickness about ±3% of variation in thickness results in ±6% of write time variation by the presented model.

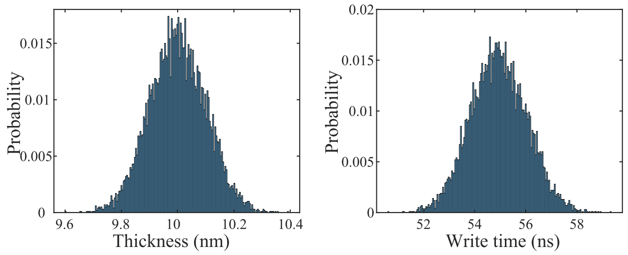

FIGURE 12. Variation of write time with thickness.

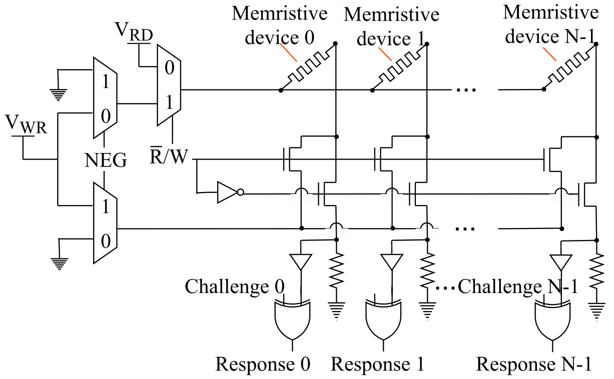

FIGURE 13. An N-bit write-time based M-PUF.

TABLE 2. Constants and parameters of the proposed model [6].

Parameters	Values	Description (Unit)
a	0.15	Ion jump distance (nm)
E_A	0.18	Ion activation energy (eV)
k_B	8.6173303 $\times 10^{-5}$	Boltzmann constant (eV K^{-1})
f	10	Ion jump frequency (THz)
q_I	2	Ion charge
V_w	10	Voltage for write (V)
Variables	**Values**	**Description (Unit)**
T	300 – 400	Temperature (K)
D	10	Thickness of memristor (nm)
R_{on}	100	Turn on resistance (Ω)
R_{off}	16000	Turn off resistance (Ω)
$R_{pulldown}$	1000	Pull-down resistance (Ω)
x_0	0	Initial state of memristor
x_f	1	Final state of memristor

We will use the sample set with similar characteristics presented in Fig. 12 for the statistical analysis of the PUF behavior. The parameters used in this simulation are summarized in Table 2.

B. MEMRISTOR MEMORY IN CROSSBAR ARRAY STRUCTURE

This study, we use the N-bit memristive PUF (M-PUF) circuit in [7] with N memristive devices, N challenge bits, N response bits. In Fig. 13, the combination between the challenge for memristive PUF and the random output of the memristive device cell by XOR gate cause the response bit of the PUF cell. Because the challenge-response pair is unique, it identifies the integrated circuit that contains M-PUF.

C. BIT-ALIASING IN PUFS

When the same challenge affects different PUFs, different responses should be generated. However, in a case where two of those responses are identical, the corresponding PUFs will be the same. This problem affects the uniqueness of a PUF. Bit-aliasing is one of the PUF performance metrics in [15], [16]. In general, bit-aliasing at the j-th bit position is estimated as the percentage of the bit values at j-th across x responses. Its equation is defined in (5).

$$Bit - aliasing = \frac{1}{x}\sum_{i=1}^{x} r_{i,j} \times 100\% \qquad (5)$$

where x is the number of responses, and $r_{i,j}$ is the value of a bit (0 or 1) at the j-th bit position of the i-th response.

The ideal value of bit-aliasing is 50%. However, the timing error occurs and make the bit-aliasing not to achieve the expected values. Then, we proposed the methods to reduce the error of bit-aliasing and make the bit-aliasing value close to the ideal value.

REFERENCES

[1] J. J. Yang, D. B. Strukov, and D. R. Stewart, "Memristive devices for computing," *Nature Nanotechnol.*, vol. 8, pp. 13–24, Dec. 2012.

[2] W. Che, J. Plusquellic, and S. Bhunia, "A non-volatile memory based physically unclonable function without helper data," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Design*, Nov. 2014, pp. 148–153.

[3] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, pp. 80–83, May 2008.

[4] J. Rajendran, H. Maenm, R. Karri, and G. S. Rose, "An approach to tolerate process related variations in memristor-based applications," in *Proc. 24th Int. Conf. VLSI Design*, Jan. 2011, pp. 18–23.

[5] X. Bi, C. Zhang, H. Li, Y. Chen, and R. E. Pino, "Spintronic memristor based temperature sensor design with CMOS current reference," in *Proc. IEEE Design, Automat. Test Eur. Conf. Exhib. (DATE)*, Mar. 2012, pp. 1301–1306.

[6] C. Merkel, "Thermal profiling in CMOS/memristor hybrid architectures," M.S. thesis, Dept. Comput. Eng., Rochester Inst. Technol., Rochester, NY, USA, 2011. [Online]. Available: https://scholarworks.rit.edu/theses/

[7] G. S. Rose, N. McDonald, L.-K. Yan, and B. Wysocki, "A write-time based memristive PUF for hardware security applications," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Design (ICCAD)*, Nov. 2013, pp. 830–833.

[8] J. Mathew, R. S. Chakraborty, D. P. Yang, Y. Yang, and D. K. Pradhan, "A novel memristor based physically unclonable function," *Integr. VLSI J.*, vol. 51, pp. 37–45, Sep. 2015.

[9] O. Kavehei, C. Hosung, D. Ranasinghe, and S. Skafidas, "mrPUF: A memristive device based physical unclonable function," 2013, *arXiv:1302.2191*. [Online]. Available: https://arxiv.org/abs/1302.2191

[10] R. Maes and I. Verbauwhede, "Physically unclonable functions: A study on the state of the art and future research directions," in *Towards Hardware-Intrinsic Security*, Berlin, Germany: Springer, 2010, pp. 3–37.

[11] J. T. H. Loong, N. A. N. Hashim, M. S. Hamid, and F. A. Hamid, "Performance analysis of CMOS-memristor hybrid ring oscillator physically unclonable function (RO-PUF)," in *Proc. IEEE Int. Conf. Semiconductor Electron. (ICSE)*, Aug. 2016, pp. 304–307.

[12] J. T. H. Loong, N. A. N. Hashim, and F. A. Hamid, "Memristor-based arbiter physically unclonable function (APUF) with multiple response bits," in *Proc. IEEE Student Conf. Res. Develop. (SCOReD)*, Dec. 2016, pp. 1–5.

[13] J. T. H. Loong, K. A. S. C. Ismail, and F. A. Hamid, "Effect of different memristor window function with variable random resistance on the performance of memristor-based RO-PUF," in *Proc. Int. Conf. Adv. Elect., Electron. Syst. Eng. (ICAEES)*, Nov. 2016, pp. 445–450.

[14] P. Koeberl, Ü. Kocabas, and A.-R. Sadeghi, "Memristor PUFs: A new generation of memory-based physically unclonable functions," in *Proc. Conf. Design, Automat. Test Eur.*, Mar. 2013, pp. 428–431.

[15] A. Maiti, "A systematic approach to design an efficient physical unclonable function," Ph.D. dissertation, Dept. Comput. Eng., Virginia Polytech. Inst. State Univ., Blacksburg, VA, USA, 2012. [Online]. Available: https://vtechworks.lib.vt.edu/handle/10919/51257

[16] J. Rajendran, R. Karri, J. B. Wendt, M. Potkonjak, N. McDonald, G. S. Rose, and B. Wysocki, "Nano meets security: Exploring nanoelectronic devices for security applications," *Proc. IEEE*, vol. 103, no. 5, pp. 829–849, May 2015.

[17] S. Eiroa, J. Castro, M. C. Martínez-Rodríguez, E. Tena, P. Brox, and I. Baturone, "Reducing bit flipping problems in SRAM physical unclonable functions for chip identification," in *Proc. 19th IEEE Int. Conf. Electron., Circuits, Syst. (ICECS)*, Dec. 2012, pp. 392–395.

[18] H. Li and M. Hu, "Compact model of memristors and its application in computing systems," in *Proc. Conf. Design, Autom. Test Eur.*, Mar. 2010, pp. 673–678.

[19] T.-N. Nguyen and D. Shin, "Statistical memristor-based temperature sensors without analog-to-digital conversion," in *Proc. IEEE 7th Non-Volatile Memory Syst. Appl. Symp. (NVMSA)*, Aug. 2018, pp. 99–104.

[20] E. I. Vatajelu, G. Di Natale, M. Indaco, and P. Prinetto, "Stt MRAM-based pufs," in *Proc. Design, Automat. Test Eur. Conf. Exhib. (DATE)*, Mar. 2015, pp. 872–875.

[21] M. A. E. S. Roel, "Physically unclonable functions: Constructions, properties and applications," Ph.D. dissertation, Dept. Elect. Eng., Katholieke Universiteit Leuven, Leuven, Belgium, Aug. 2012.

[22] Hewlett-Packard Software Company, "Application note 200 electronic counter series: Fundamentals of the electronic counters," Electron. Counter Ser., Englewood, CO, USA, Appl. Note 200, 1997.

[23] D. S. Boning, J. Stefani, and S. W. Butler, "Statistical methods for semiconductor manufacturing," in *Wiley Encyclopedia of Electrical and Electronics Engineering*. Hoboken, NJ, USA: Wiley 1999, pp. 463–479. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/047134608X.W7041

[24] A. Maiti and P. Schaumont, "Improved ring oscillator PUF: An FPGA-friendly secure primitive," *J. Cryptol.*, vol. 24, no. 2, pp. 375–397, 2011.

[25] A. Maiti, V. Gunreddy, and P. Schaumont, "A systematic method to evaluate and compare the performance of physical unclonable functions," in *Embedded Systems Design with FPGAs*. New York, NY, USA: Springer, 2013, pp. 245–267.

[26] V. Sehwag and T. Saha, "TV-PUF: A fast lightweight analog physical unclonable function," in *Proc. IEEE Int. Symp. Nanoelectron. Inf. Syst. (iNIS)*, Dec. 2016, pp. 182–186.

[27] D. Mukhopadhyay and R. S. Chakraborty, *Hardware Security: Design, Threats, and Safeguards*. Orange, CA, USA: Chapman Hall/CRC, 2014. [Online]. Available: https://www.taylorfrancis.com/books/9780429066900

[28] U. Chatterjee, R. S. Chakraborty, J. Mathew, and D. K. Pradhan, "Memristor based arbiter PUF: Cryptanalysis threat and its mitigation," in *Proc. 29th Int. Conf. VLSI Design 15th Int. Conf. Embedded Syst. (VLSID)*, Jan. 2016, pp. 535–540.

[29] M. Uddin, M. D. B. Majumder, K. Beckmann, H. Manem, Z. Alamgir, N. C. Cady, and G. S. Rose, "Design considerations for memristive crossbar physical unclonable functions," *ACM J. Emerg. Technol. Comput. Syst. (JETC)*, vol. 14, no. 1, p. 2, Mar. 2018.

[30] Y. Ho, G. M. Huang, and P. Li, "Dynamical properties and design analysis for nonvolatile memristor memories," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 58, no. 4, pp. 724–736, Apr. 2011.

HA-PHUONG NGUYEN received the M.S. degree in computer science from the University of Science and Technology - The University of Da Nang, Vietnam, in 2013. She is currently pursuing the Ph.D. degree with the Department of Computer Engineering, Yeungnam University, South Korea. Her research interests include low-power embedded system application, machine learning, and hardware security.

THE-NGHIA NGUYEN received the M.S. degree in computer engineering from Yeungnam University, South Korea, in 2018. He is currently pursuing the Ph.D. degree with the Department of Software Convergence, Soongsil University, South Korea. His research interests include low-power embedded system application, machine learning, and hardware security.

YEONG-SEOK SEO received the B.S. degree in computer science from Soongsil University, Seoul, South Korea, in 2006, and the M.S. and Ph.D. degrees in computer science from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2008 and 2012, respectively. From September 2012 to December 2013, he was a Postdoctoral Researcher with the KAIST Institute for Information and Electronics, Daejeon. From January 2014 to August 2016, he was a Senior Researcher with the Korea Testing Laboratory (KTL), Seoul. Since 2016, he has been an Assistant Professor with the Department of Computer Engineering, Yeungnam University, Gyeongsan, Gyeongbuk, South Korea. His research interests include software engineering, the Internet of Things, artificial intelligence, data mining, and big data analysis. He is a member of board of directors of software engineering society in Korea. He was a recipient of the 2nd JIPS Survey Paper Awards, in 2019. He has served as the Proceedings Co-Chair for APSEC 2018, the Publicity Chair for WITC 2019, and the Program Chair for HCIS 2019. He is an Associate Editor of *Journal of Information Processing Systems* (JIPS) and *Korea Information Processing Society* (SCOPUS/ESCI indexed), and a Guest Editor of *Journal of Systems and Software* (JSS), Elsevier (SCIE indexed). Also, he is involved in international standardization activities and is a member of the Korean National Body mirror committee to ISO on IT Service Management and IT Governance (ISO/IEC JTC1/SC40).

DOSAM HWANG received the Ph.D. degree from Kyoto University, Kyoto, Japan. He was the Head of the Yeungnam University's Computer Engineering Department, from 2005 to 2009. He was a Principal Researcher with the Korea Institute of Science and Technology (KIST) and an Invited Professor with the Korea Advanced Institute of Science and Technology (KAIST). He has been not only a chair of several international conferences but also a committee member of many international organizations as well. He has been the Assistant Secretary of ISO/TC37/SC4 for language resource management, since 2005, and also the Secretary of Korean TC for ISO/TC37/SC4. In 2006, he was the Director of the Korean Society for Cognitive Science (KSCS) and the Korean Information Science Society (KISS). He has been the Society's Director and the Mentor of a knowledge engineering study group, since 2007. He has also participated in several Korean national research projects, such as a project on machine translation system, from 1985 to 1990, and the national IT ontology infrastructure and technology development project called CoreOnto, from 2006 to 2009, and Exobrain, from 2013 to 2014, the project focused on the construction of deep knowledge base and question-answering platform. He has been in charge of an intelligent service integration based on IoT Big Data as part of Korea's another principal national research project BK, since 2014. His research interests mainly include natural language processing, ontology, knowledge engineering, information retrieval, and machine translation. He is currently a Full Professor with the Department of Computer Engineering, Yeungnam University, Korea. He had more than 50 publications. In recognition of his such great commitment and contribution to the relative fields of study, he has been honored as a Distinguished Researcher of KIST by Korea's Ministry of Science and Technology (MoST), in 1988, and awarded a prize for Good Conduct from Kyunghee High School, in 1973.

DONGHWA SHIN (S'05–M'12–SM'18) received the B.S. degree in computer engineering and the M.S. and Ph.D. degrees in computer science and electrical engineering from Seoul National University, Seoul, South Korea, in 2005, 2007, and 2012, respectively. He joined the Ming Hsieh Department of Electrical Engineering, University of Southern California, Los Angeles, CA, USA, as a Visiting Scholar, and the Dipartimento di Automatica e Informatica-EDA Group, Politecnico di Torino, Turin, Italy, as a Researcher. He was an Assistant Professor with the Department of Computer Engineering, Yeungnam University, Gyeongsan, South Korea, from 2014 to 2017. He is currently an Assistant Professor with the Department of Smart Systems Software, Soongsil University, Seoul. His research interests include system-level low-power techniques for embedded systems and hybrid power system design, and he is currently focusing on the next-generation computing and energy resources, including neuromorphic computing. He serves (and served) as a Reviewer of the IEEE TRANSACTIONS ON COMPUTERS, TCAD, TVLSI, ACM TODAES, TECS, JETC, *International Journal of Electrical Power and Energy Systems*, *Journal of Applied Electrochemistry*, and *Journal of Signal Processing Systems*, and the *International Symposium on Industrial Electronics*. He serves on the Technical Program Committee of IEEE and ACM technical conferences, including Design Automation and Test in Europe (DATE), International Symposium on Low-Power Electronics and Design (ISLPED), Asia South-Pacific Design Automation Conference (ASP-DAC), ACM Great Lakes Symposium on VLSI (GLSVLSI), International Green and Sustainable Computing Conference (IGSC), and IFIP/IEEE International Conference on Very Large Scale Integration.