In Situ Aging-Aware Error Monitoring Scheme for IMPLY-Based Memristive Computing-in-Memory Systems

Jiarui Xu, *Member, IEEE*, Yi Zhan, *Member, IEEE*, Yujie Li, *Member, IEEE*, Jiajun Wu, *Member, IEEE*, Xinglong Ji, Guoyi Yu, *Member, IEEE*, Wenyu Jiang, *Senior Member, IEEE*, Rong Zhao, *Member, IEEE*, and Chao Wang, *Senior Member, IEEE*

***Abstract*— Stateful logic through memristor is a promising technology to build Computing-in-Memory (CIM) systems. However, aging-induced degradation of memristors' threshold voltage imposes a major challenge to the reliability and guardbands estimation of memristive CIM systems, especially the Material Implication (IMPLY) logic based CIM systems. In this paper, a novel in-situ aging-aware error monitoring scheme for memristor-based IMPLY logic is proposed. The proposed in-situ error monitoring scheme can achieve faster error detection speed and higher detection accuracy than the straightforward program-verify monitoring scheme. Simulation results under Monte-Carlo simulation show that the proposed monitoring scheme can effectively detect the major operation failures existing in IMPLY logic operations with a detection accuracy up to 99.95%. Moreover, a case study of error monitoring design of 4-bit IMPLY-based adder is carried out. The analysis result exhibits that the proposed in-situ monitoring scheme can achieve 75.2% improvement on the detection speed against the program-verify scheme. Further analysis on a convolution filter in VGG-11 based Binarized Neural Network shows that 74% improvement on the detection speed can also be achieved by using the proposed monitoring scheme, which suggests that the proposed in-situ error monitoring scheme is an efficient solution to improve the reliability of IMPLY-based memristive CIM systems.**

***Index Terms*— Material implication (IMPLY) logic, memristor, in-situ error monitoring and detection, aging-induced degradation, computing-in-memory, logic-in-memory, edge computing.**

Manuscript received March 7, 2021; revised June 6, 2021; accepted July 5, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61974053, in part by the Singapore Government's Research, Innovation and Enterprise 2020 Plan (Advanced Manufacturing and Engineering domain) under Grant A1687b0033, and in part by the Fundamental Research Funds of the Central Universities under Grant 2019KFYXJJS049. This article was recommended by Associate Editor S.-B. Ko. *(Corresponding author: Chao Wang.)*

Jiarui Xu, Yi Zhan, Yujie Li, Jiajun Wu, and Guoyi Yu are with the School of Optical and Electronic Information, Huazhong University of Science and Technology, Wuhan 430074, China.

Xinglong Ji was with the Department of Engineering and Product Development, Singapore University of Technology and Design, Singapore 487372. He is now with the Center for Brain-Inspired Computing Research, Tsinghua University, Beijing 100084, China.

Wenyu Jiang is with the Institute for Infocomm Research, Agency for Science, Technology and Research (A∗STAR), Singapore 138632.

Rong Zhao was with the Department of Engineering and Product Development, Singapore University of Technology and Design, Singapore 487372. She is now with the Department of Precision Instruments, Tsinghua University, Beijing 100084, China, also with the Center for Brain-Inspired Computing Research, Tsinghua University, Beijing 100084, China, and also with the Innovation Center for Future Chip, Tsinghua University, Beijing 100084, China.

Chao Wang is with the School of Optical and Electronic Information, Huazhong University of Science and Technology, Wuhan 430074, China, and also with the Wuhan National Laboratory of Optoelectronics, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: chao.wang.1978@hotmail.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSI.2021.3095545.

Digital Object Identifier 10.1109/TCSI.2021.3095545

I. Introduction

INTERNET-OF-THINGS (IoT) applications require low-power intelligent edge computing devices significantly. This demand is posing a particular interest in the innovative research that enables the computation directly inside the memory, aiming to reduce the time and energy in the process of data movement between memory and computing units. The energy-hungry data movement deteriorates the memory wall issue lying in the traditional Von Neumann architecture [1]. Among the emerging memory technologies, thanks to the excellent properties of non-volatile storage, fast-switching, and ultra-low power consumption [1], [2], memristor is indeed a promising candidate for Computing-In-Memory (CIM) systems. A specific scheme based on the stateful IMPLY logic is preferred in the non-volatile CIM solutions especially for fast and low-power Logic-in-Memory (LIM) operations [2]–[6]. This IMPLY-based memristive LIM operations are particularly suitable for the simple logic operations and signal processing as well as classification tasks based on lightweight neural network models in intelligent edge-devices. However, memristors' non-ideality issues including device variabilities, resistance/logic state drift, and aging-induced degradation [1], [4]–[6], have posed a great challenge in the reliability of IMPLY-based memristive CIM systems.

Recently, addressing memristors' non-idealities for improvement of IMPLY operation reliability has been increasingly gaining attention from the research community [4]–[6]. Design methodology for memristive IMPLY logic has been reported to give a guidance on the design constraints and parameter determination, with the aim of addressing the device variability issues such as cycle-to-cycle and device-to-device variations [4]. Smart IMPLY structure

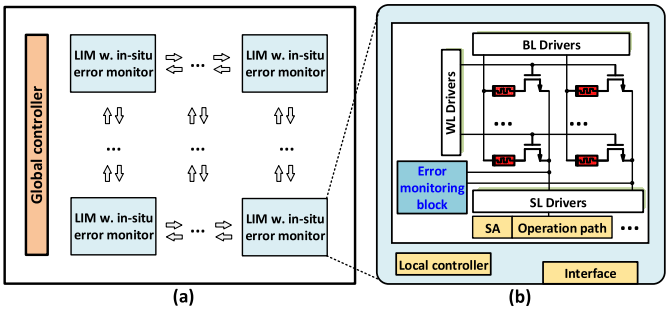

Fig. 1. (a) Diagram of IMPLY-based Computing-in-Memory systems; (b) 1T1R memristor array of an IMPLY-based Logic-In-Memory (LIM) block with built-in error monitoring block.

has been proposed to solve the reliability challenge on logic state drift and degradation of memristors in IMPLY logic [5]. As the threshold voltage is actually one of key metrics for memristor operation, researchers have started to study the aging-induced degradation issue of memristors' threshold voltage recently [7], [8]. Grossi *et al.* [7] report the endurance test results of HfO_2 based memristors. Robayo *et al.* [8] present a comprehensive and systematic analysis of endurance reliability of conductive bridge based memristors. The pioneer research of memristor's threshold voltage degradation is very significant to the memristor-based stateful logic, where the memristive devices are repeatedly written during the logic operation. The limited cycling endurance of memristors is a key drawback to the memristor-based stateful logic [1]. However, the study of addressing aging-induced degradation of memristors' threshold voltage in IMPLY logic is still missing in the literature, as far as we know.

Generally, the straightforward program-verify method in [9], [10] can be developed to ensure correctness of memristor programming process during the memristive CIM operations. However, this method incurs extra reading process after stepping incremental programming pulses to verify the memristors' resistance state. Moreover, the programming pulses will be constantly increased until the cell resistance reaches the expected value. Considering the overhead of extra operational steps and time, this program-verify method is inefficient and not suitable for error monitoring of IMPLY-based memristive CIM systems. The logic operations in memristive CIM systems have to be interrupted for allowing extra programming and reading process, so that the expected switching of memristive state occurred by programming pulses can be verified. This interruption seriously degrades the system efficiency of IMPLY-based memristive CIM systems.

To solve the aforementioned issues of the program-verify scheme, a novel in-situ error monitoring scheme for IMPLY-based memristive CIM systems is proposed in this paper, as illustrated in Fig. 1. To the best of our knowledge, this is the first work to systematically analyze and solve the reliability issues of aging-induced memristors' threshold voltage degradation in IMPLY logic by in-situ monitoring and detection. This paper extends our work in the conference paper [11], which only proposes the in-situ error monitoring concept for memristor-based IMPLY logic with preliminary simulation results. The major contributions of this paper include:

1) Operation failures in IMPLY logic caused by memristors' threshold voltage degradation are systematically analyzed. Two major operation failures that cannot be protected by conventional guardband scheme are identified and analyzed for designing an effective error detection and identification scheme, to enable the IMPLY-based memristive CIM system adaptively adjust the magnitudes of operational voltage pulses and improve the reliability of IMPLY logic operation.

2) An in-situ error monitoring circuit for IMPLY-based logic (covering both IMPLY and FALSE operations) is proposed to detect the two major operation failures, thereby mitigating the impact of aging-induced degradation of memristors' threshold voltage.

3) Based on the experimental results of memristor endurance test from the literature, a case study of the proposed error monitoring scheme is carried out for the evaluation of efficiency improvement of an IMPLY-based full adder.

4) Further analysis on a convolution filter in VGG-11 based Binarized Neural Network (BNN) is also performed. The potential reliability improvement of the BNN inference on the IMPLY-based memristive CIM architecture with the proposed in-situ monitoring scheme is evaluated.

The remainder of this paper is organized as follows: Section II describes aging-induced operation failures of memristor-based IMPLY logic gate, and the in-situ error monitoring scheme for IMPLY-based memristive CIM systems is presented. Section III presents a case study of the in-situ error monitoring design for an IMPLY-based full adder and an analysis of a convolution filter based on the memristive IMPLY logic for BNN applications. Section IV presents a conclusion of this paper.

II. Proposed In-Situ Error Monitoring Scheme for IMPLY-Based Computing-in-Memory Systems

A. Principle of IMPLY Logic and Memristive IMPLY Realization

The logic function P IMPLY Q is described in [4] and the truth table of IMPLY logic is listed in Table I, where the output Q' is the final state of input Q depending on the state of input P. The state of Q is set to logic "1" (i.e., Q' = "1") when $P = 0$, while the initial state of Q is maintained (i.e., Q' = Q) when $P = 1$. In general, each Boolean function can be realized by a sequence of core operations based on IMPLY and FALSE (denoted as FALSE Q, meaning it always resets Q to logic "0", i.e., Q' = "0"). For example, XOR operation can be realized by (P IMPLY Q) IMPLY ((Q IMPLY P) IMPLY 0), where logic "0" can come from the output of FALSE operation on some dummy memristor.

Fig. 2 (a) and (b) show the symbol of the memristor and logic correspondence of memristor, respectively. Memristor changes to low resistance state (R_{LRS}) when the voltage drop across the memristor is larger than the positive threshold voltage V_{close} ($V_{close} > 0$), which is called SET process. Conversely, the memristor switches to high resistance state (R_{HRS}) when the voltage drop across the memristor is lower than the negative threshold voltage V_{open} ($V_{open} < 0$), which

TABLE I
TRUTH TABLE OF MATERIAL IMPLICATION (IMPLY) LOGIC

Case	P	Q	Q'
1	0	0	1
2	0	1	1
3	1	0	0
4	1	1	1

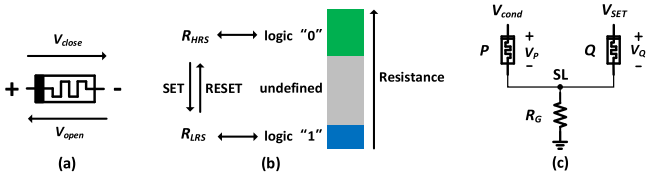

Fig. 2. (a) Symbol of memristor: the thick black line on the left terminal represents the positive polarity of memristor; (b) correspondence between memristors and logic values: R_{LRS} corresponds to logic "1", while R_{HRS} corresponds to logic "0"; (c) Memristive IMPLY gate realization: two parallel memristor P and Q with a serial resistor R_G, and voltage pulses V_{cond} and V_{SET} are simultaneously applied to P and Q, respectively.

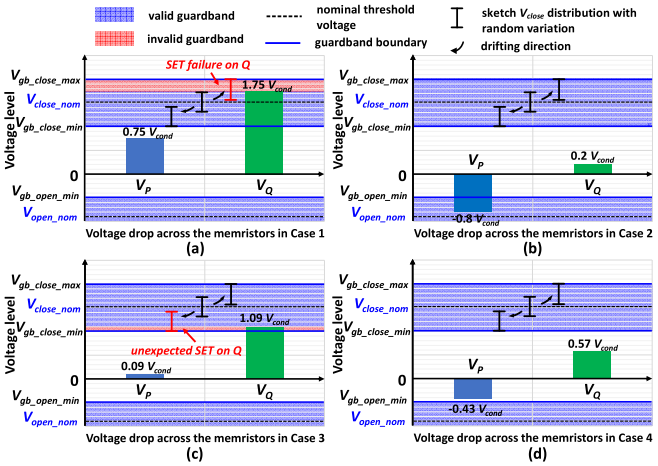

Fig. 3. Different voltage drop across memristor P and Q (i.e., V_P and V_Q) with V_{close} guardband under different input cases: (a) Case 1; (b) Case 2; (c) Case 3; (d) Case 4. Note: assume $R_{HRS} : R_G : R_{LRS} = 100 : 10 : 1$, and $V_{SET}/V_{cond} = 2$; The guardband preset by V_{gb_max} and V_{gb_min} (i.e., equal to V_{SET} and V_{cond} in our study, respectively) is only valid in a narrower range for input Case 1 and Case 3.

is called RESET process. In general, the IMPLY and FALSE functions involving set and reset operations are perfectly compatible with the memristor's characteristic of two-state switching.

IMPLY logic can be realized by utilizing conditional SET of memristors [3]. Fig. 1 (c) shows the IMPLY logic gate based on memristor. To perform an IMPLY operation only involving conditional SET process, the voltage pulses V_{cond} and V_{SET} are simultaneously applied to P and Q, respectively. V_P and V_Q are the voltage drop across the memristor P and Q, respectively. During the operation, the memristor P remains unchanged, while the memristor Q changes its state only when Case 1 happens, and Q' represents the final state of Q after the operation, as shown in Table I. That is, the memristor Q is conditionally SET when $P = 0$ and $Q = 0$ (i.e., Case 1), while maintaining its initial state in the other three input cases.

B. Analysis of Aging-Induced Operation Failure of Memristor Based IMPLY Gate

To implement the correct operations on P and Q in IMPLY logic that only involve SET process, V_Q must be above the V_{close} ($V_{close,Q}$) for Case 1 or below the $V_{close,Q}$ for Case 3, while V_P must be always below the V_{close} ($V_{close,P}$). Thus, V_{cond} and V_{SET} must follow the equations from (1) to (5):

$$V_{SET} > V_{close} + |\Delta v_{guardband}| \quad (1)$$

$$V_{cond} < V_{close} - |\Delta v_{guardband}| \quad (2)$$

$$V_{SET} - V_{cond} < V_{close} - |\Delta v_{guardband}| \quad (3)$$

$$R_G = \sqrt{R_{HRS} R_{LRS}} \quad (4)$$

$$\frac{1}{2}|V_{SET} - V_{cond}| < |V_{open}| - |\Delta v_{guardband}| \quad (5)$$

where V_{close} and V_{open} are the positive and negative threshold voltages, for the SET and RESET of memristor, respectively. The $\Delta v_{guardband}$ is a derived guardband variable for handling V_{close} variation. Notably, equation (5) is an additional constraint to ensure the $|V_P|$ is always lower than $|V_{open,P}|$ for Case 4. The guardband provides a certain margin for addressing random variation of the memristors' threshold voltage (e.g., cycle-to-cycle and device-to-device variations). The operation failure of the IMPLY operation induced by the random variation can be prevented as much as possible within the guardband. The constraint of R_G can be determined from V_P and V_Q under different input cases. R_G should be larger than R_{LRS} and at the same time lower than R_{HRS}. Equation (4) shows a reasonable value of R_G as the geometrical mean of R_{LRS} and R_{HRS} [4]. The boundaries of the preset guardband for V_{close} (i.e., V_{gbQ_max} and V_{gbQ_min}) correspond to the allowed minimum and maximum amplitudes for the operational voltage pulses V_{SET} and V_{cond}, respectively, according to the equation from (1) to (3). Similarly, the allowed lower boundary of the preset guardband for V_{open} (i.e., V_{gbP_min}) is determined by the equation (5).

Fig. 3 shows the initial V_P and V_Q under different input cases derived from (6) according to Kirchhoff current law during IMPLY operation (Refer to Fig. 2 (c)). Without loss of generality, typical setting of $R_{HRS} : R_G : R_{LRS} = 100 : 10 : 1$ and $V_{SET}/V_{cond} = 2$ are selected from [4]. In equation (6), R_P and R_Q are the resistances of memristor P and Q, respectively.

$$\frac{V_{cond} - V_{SL}}{R_P} + \frac{V_{SET} - V_{SL}}{R_Q} + \frac{0 - V_{SL}}{R_G} = 0,$$

$$V_{SL} = R_G \frac{V_{cond} R_Q + V_{SET} R_P}{R_P R_Q + R_P R_G + R_Q R_G} \quad (6)$$

There are only three possible failure types for the memristor-based IMPLY logic, i.e., unexpected SET, SET failure, and unexpected RESET, as the correct IMPLY logic only involves SET operation on Q. Considering additional drifting variation of threshold voltage induced by memristor's aging process, there could be four possible failures during IMPLY operations (Refer to Table II):

1) Type-I failure: unexpected SET on P can occur in Case 1, when the $V_{close,P}$ drops below the V_P;

TABLE II
POSSIBLE OPERATION FAILURES BY AGING-INDUCED THRESHOLD VOLTAGE DEGRADATION IN IMPLY LOGIC

Case	P	Q	P'	Operation failure	Q'	Operation failure
1	0	0	0	√ (Unexpected SET)	1	√ (SET failure)
2	0	1	0	×	1	×
3	1	0	1	×	0	√ (Unexpected SET)
4	1	1	1	√ (Unexpected RESET)	1	×

2) Type-II failure: unexpected RESET on P can occur in Case 4, when the $|V_{open,P}|$ drops below the $|V_P|$;

3) Type-III failure: SET failure on Q can occur in Case 1, when the $V_{close,Q}$ rises above the V_Q;

4) Type-IV failure: unexpected SET on Q can occur in Case 3, when the $V_{close,Q}$ drops below the V_Q.

Note that the operation failure of unexpected SET and RESET on P does not occur for Case 2 and Case 3, respectively, because the V_P always has an opposite polarity to the threshold voltage, i.e., the $V_{close,P}$ and $V_{open,P}$ for the Case 2 and Case 3, respectively (Refer to Fig. 3 (b) and Fig. 3 (c), respectively). As for the memristor Q, the $V_{close,Q}$ does not cause the operation failure of unexpected SET on Q in Case 2 or Case 4, because the initial state of Q is already "1". On the other hand, the $V_{open,Q}$ also does not cause the operation failure of RESET, because the V_Q is positive for both Case 2 and Case 4 (Refer to Fig. 3(b) and Fig. 3(d), respectively).

Unfortunately, only two of the above possible failures summarized in Table II can be avoided by the guardband protection scheme. For the P in Case 1, as the V_P is always lower than the $V_{close,P}$ within the guardband as depicted in Fig. 3 (a), the unexpected SET does not happen under the effective protection of the guardband. Similarly, for the P in Case 4, the unexpected RESET in Case 4 also does not happen, because the $|V_P|$ is always lower than the $|V_{open,P}|$ within the guardband as depicted in Fig. 3 (d). For the Q in Case 1 and Case 3, the guardband cannot completely avoid the occurrence of SET failure and unexpected SET, respectively, because there is always a limited margin for the practical guardband, as depicted by $V_{gb\_close\_max}$, $V_{gb\_close\_min}$ and $V_{gb\_open\_min}$ (Refer to Fig. 3 (a) and Fig. 3(c)) The red areas of guardband in Fig.3 sketch the two major types of operation failure on Q. i.e., Type-III and Type-IV.

In summary, there are two major failures in the IMPLY operation that can occur even under the guardband protection, due to aging-induced increased $V_{close,Q}$ [7] or decreased $V_{close,Q}$ [8], respectively:

1) Type-III failure in Case 1: an increased $V_{close,Q}$ leads to the SET failure on Q caused by an insufficient V_Q, as illustrated in Fig. 3 (a);

2) Type-IV failure in Case 3: a decreased $V_{close,Q}$ leads to the unexpected SET on Q caused by an excessive V_Q, as illustrated in Fig. 3 (c).

Obviously, a wider guardband is not a practical solution to solve the above two failures on Q, because the memristor threshold voltage continues to drift, and the device aging process inevitably occurs during the whole lifetime before the device breakdown.

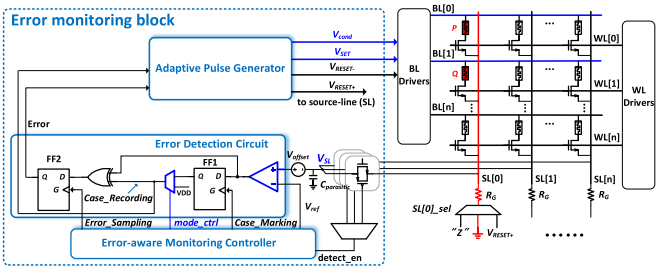

Fig. 4. Schematic of the proposed in-situ error monitoring circuit in 1T1R array: the monitoring circuit as a built-in block can be shared by the memristor array for detecting aging-induced IMPLY operation failures and adaptively adjusting the operation pulses to improve reliability.

Similarly but much simpler, the failure of FALSE operation could also happen when R_{HRS} does not reach the expected value by the RESET operation due to the aging-induced V_{open} increasing (called by RESET failure for FALSE), while decreased V_{open} never cause any operation failure for the FALSE operation. As same to the IMPLY logic, the RESET failure cannot be well addressed by the guardband scheme.

The straightforward memristor monitoring method uses SET or RESET pulse to program and read pulses to verify [9] (called as program-verify scheme in short). This method can be used to check the correctness of memristor SET or RESET process to detect the aging-induced reliability issues. However, this program-verify scheme has a major drawback: it cannot be used for adaptively in-situ monitoring without interrupting the IMPLY-based functional operations. This is because the intermediate status of the memristors under test must be temporally stored for the subsequent normal operations after test, which deteriorates the overall system efficiency of IMPLY-based CIM systems.

Currently, there is lack of an efficient in-situ monitoring and detection mechanism for IMPLY-based memristive CIM systems to address the memristor's threshold voltage degradation issue, without sacrificing the system efficiency.

C. Proposed In-Situ Error Monitoring Scheme for IMPLY Based Computing-in-Memory System

Fig. 4 presents the proposed aging-aware in-situ error monitoring circuit for IMPLY-based CIM systems based on 1T1R memristor array, which can be embedded in a LIM block and shared by the whole logic array as depicted in Fig. 1. This in-situ monitoring scheme enables real-time and in-place error detection when the memristors are accessed during IMPLY and FALSE operations. The proposed error monitoring block consists of an error detection circuit, an adaptive pulse generator, and an error-aware monitoring controller. Compared to the conventional voltage driver, the voltage drivers in our proposed scheme requires the pulse generator with adjustable amplitude, a power management unit to provide supply voltage tuning, and the test controller for the voltage tuning and working-state switching. The error detection circuit can be programmed to timely detect whether operation failures occur in the memristor array by presetting the detection times, and send an error signal to the adaptive pulse voltage generator

after a failure detection. The adaptive pulse generator module adaptively adjusts the operation voltage pulses for ensuring the reliability of the LIM block under monitoring, according to the error signal and the *case_recording* signal.

The proposed in-situ error detection circuit can work under two detection modes, for monitoring the normal IMPLY or FALSE operations, respectively, through the selection signal (i.e., *mode_ctrl*) of MUX left to the flip-flop FF1. Under the IMPLY monitoring mode, the error detection circuit works in two phases: In Phase 1, it firstly performs an extra identification operation to distinguish Case 3 input from the other three input cases; Subsequently in Phase 2, the error detection circuit performs the built-in error detection together with the normal IMPLY operation without any interruption. The error monitoring circuit under the FALSE monitoring mode, can directly perform built-in error detection of failure RESET process together with for the normal FALSE operation. In this working manner, the proposed in-situ monitoring scheme only introduces an overhead operation of input case identification for IMPLY monitoring.

The key idea of the proposed error monitoring is from the observation that operation failure can be detected by sensing the source-line voltage V_{SL}. This is because the change of memristors' resistance state after RESET/SET operation will be directly reflected in the change of V_{SL}. In other words, the operation failure can be determined and classified through the different changing patterns of V_{SL} during the logic operation, as long as the input cases have been identified.

For the IMPLY operation, to detect the two types of operation failures on Q discussed previously, we propose a two-phase error detection scheme implemented in the IMPLY monitoring mode by the proposed in-situ error detection circuit in Fig. 4. Fig. 5 (a) shows the IMPLY monitoring flow and Fig. 5 (b) shows the operational waveforms for a detection of Type-IV failure in Case 3 as an illustrative example. Fig. 6 (a) and (b) show the V_{SL} under different input cases during Phase 1 and Phase 2, respectively. Note that V_{SET} is twice as large as V_{cond} following the equations (1) to (3).

During the Phase 1 of IMPLY monitoring, after the source line is connected to a high-impedance state by the selection switch controlled by *SL_sel*, voltage pulses V_{cond} and V_{SET} are simultaneously applied to P and Q, respectively (Refer to Fig. 4). A reference voltage V_{ref1} can be used by the comparator circuit to distinguish Case 3 input from the others, as shown in Fig. 6 (a). Derived from the voltage-dividing on P and Q, the V_{SL} in Phase 1 is depicted by

$$V_{SL\_Phase1} = V_{cond} + R_P \frac{V_{SET} - V_{cond}}{R_P + R_Q} \tag{7}$$

For distinguishing out the Case 1, V_{ref1} is set to the midpoint between the V_{SL} levels of input Case 3 and Case 1 to ensure a sufficient detection margin, as shown in Fig. 6 (a). Hence the output of comparator falls to low only for input Case 3. Then the flip-flop FF1 can capture the output of comparator and pass to the subsequent multiplexer controlled by the signals *Case_Marking* and *mode_ctrl* from the controller. The output of multiplexer as *Case_recording* is used by the XOR gate in the Phase 2.

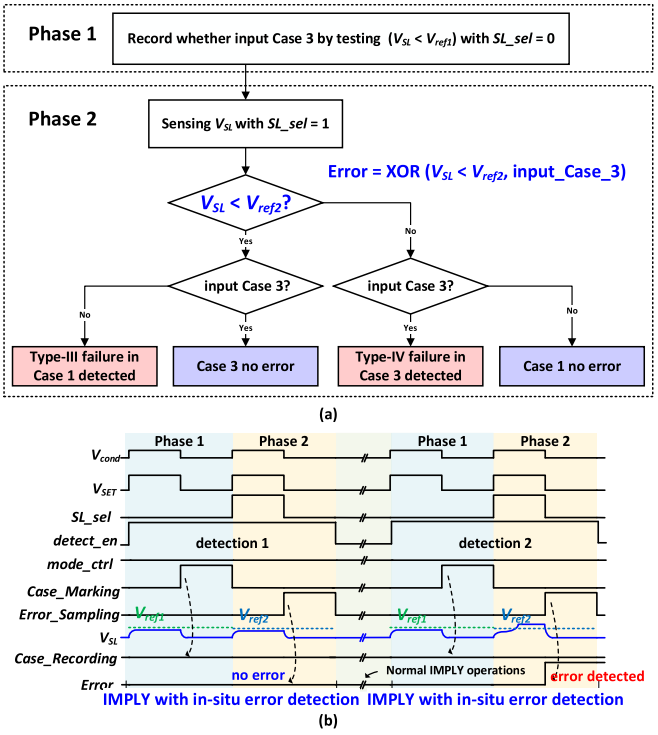

Fig. 5. (a) The IMPLY monitoring flow: two steps are required to classify the two major operation failures in IMPLY operations: (i) identifying input Case 3; (ii) sensing the V_{SL} and detecting error during normal IMPLY operation; (b) Operational waveforms of IMPLY monitoring for detection of Type-IV failure in Case 3.

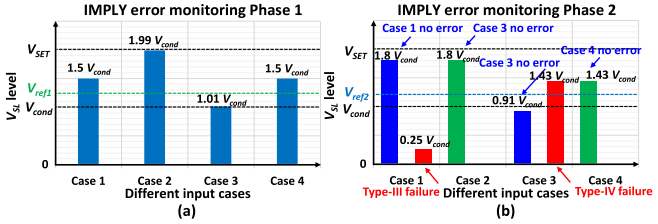

Fig. 6. (a) Different V_{SL} levels in Phase 1 of IMPLY monitoring: Case 3 input can be distinguished by a suitable V_{ref1} from the other input cases; (b) Different V_{SL} levels in Phase 2 of IMPLY monitoring: a suitable V_{ref2} is used to detect the error with recorded case information.

During the Phase 2 of the proposed IMPLY monitoring, after the source line is connected to ground by the switch, V_{cond} and V_{SET} are simultaneously re-applied to P and Q, respectively. Another reference voltage V_{ref2} is used to detect the two major operation failures, i.e., Type III failure in Case 1 and Type IV in Case 3, which happens when the V_{SL} is below the V_{ref2} in Case 1 (i.e., SET failure on Q, $Case\_Recording = 1$) or exceeds the V_{ref2} in Case 3 (i.e., unexpected SET on Q, $Case\_Recording = 0$) after the normal IMPLY operation, respectively, as shown in Fig. 6 (b). Similar to the V_{ref1}, V_{ref2} is set to the midpoint between the V_{SL} levels of input Case 3 and Case 4. As a result, the output of XOR gate is high according to *Case_Recording* and the output of comparator, and the output *Error* of another flip-flop FF2 is set to high correspondingly at the positive edge of *Error_Sampling*.

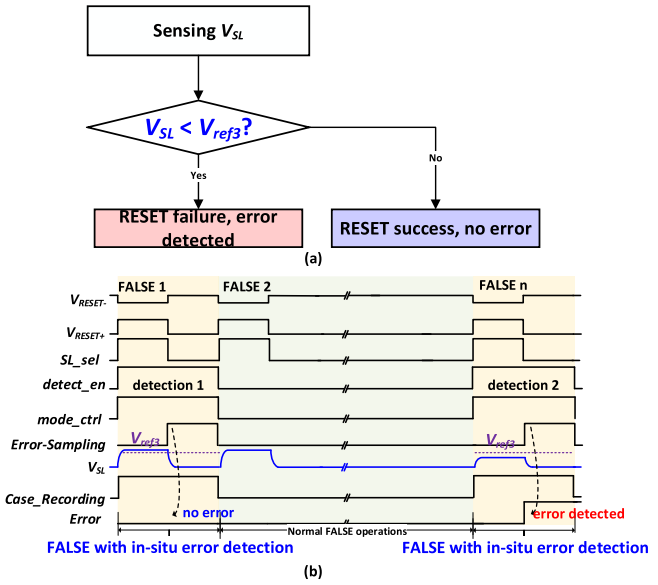

Fig. 7. (a) The FALSE test flow: note that only one step is required to perform FALSE operation with built-in monitoring; (b) Operational waveforms of built-in FALSE monitoring for detection of RESET failure by checking whether the memristor has reached the target R_{HRS} value.

As for the FALSE operation, Fig. 7 (a) shows the monitoring flow for FALSE monitoring, and Fig. 7 (b) shows the operational waveforms for a detection of RESET failure in FALSE operation as an illustrative example. In the FALSE operation, after the source line is connected to the V_{RESET+}, the positive pulse V_{RESET+} and negative pulse V_{RESET-} are used to the R_G and positive polarity of the memristor respectively, to complete the RESET operation. It should be mentioned that the initial state of the memristor does not need to be recorded, so the FF1 is bypassed, and the multiplexer controlled by *mode_ctrl* selects the path of *VDD* to make *Case_ Recording* always high in the FALSE monitoring mode. The third reference voltage V_{ref3} is used by the comparator to detect the RESET failure, because a memristor fails to RESET to the target R_{HRS} and results in V_{SL} below a certain level that can be measured by the V_{ref3}, calculated by

$$V_{ref3} = V_{RESET-} + (V_{RESET+} - V_{RESET-}) \times \frac{R_{HRS\_min}}{R_{HRS\_min} + R_G} \quad (8)$$

When the RESET failure happens, the output of comparator becomes low and the output of XOR gate is high. The flip-flop FF2 will capture the detection result and pass the *Error* signal to the adaptive pulse generator for adjusting the RESET pulses.

In general, our proposed method has the advantage of non-interruption and fast failure detection. It only inserts one extra identification step (i.e., the IMPLY monitoring Phase 1) to complete the failure detection for IMPLY and FALSE operations, which is faster than the program-verify method that must interrupt the normal operations and requires at least 4 extra steps to complete a program-verify cycle (e.g., 2 steps for SET verification and 2 steps for RESET verification).

TABLE III
PARAMETER SETTINGS

Parameter	Value	Parameter of TEAM model	Value
V_{SET}	1 V	V_{close}	0.7 V
V_{cond}	0.5 V	V_{open}	-1.5 V
V_{RESET+}	1.5 V	k_{on}	-0.05
V_{RESET-}	-0.5 V	k_{off}	0.1
Pulse Width	500 ns	α_{on}	3
VDD	1.8 V	α_{off}	3
V_{ref1}	0.625 V	R_{HRS}	100 kΩ
V_{ref2}	0.585 V	R_{LRS}	1 kΩ
V_{ref3}	1.2 V		

D. Simulation and Discussion

The proposed error detection circuit for CIM systems based on memristive IMPLY logic is designed and simulated in a 0.18-μm CMOS technology in this study. A mixed-simulation with widely-used TEAM memristor Verilog-A model [14] and transistor SPICE model in Cadence Virtuoso has been performed to verify the function of the proposed error detection circuit. The settings of relevant parameters are listed in Table III, which are basically aligned with [4]. A conventional two-stage open-loop comparator as the core sub-circuit is used in our work. The proposed error detection circuit totally requires 72 transistors, where the digital sub-circuits including flip-flops, MUX, and XOR gate, are all designed with the minimized size. The power and delay overhead will be discussed in the section. III.

It is worth mentioning that the key requirement for a reliable operation of the proposed monitoring scheme is a sufficient detection margin of the comparator to compare the different V_{SL} levels robustly, and also a wide enough input common mode range (ICMR) of the comparator to be using in the two monitoring modes requiring three different reference voltage levels. This makes the detection margin and stability of the comparator a key part of the comprehensive metrics of the scheme reliability.

As shown in Fig. 8, to evaluate the detection margin used by the comparator, we have simulated the distribution of V_{SL} considering the different input cases in IMPLY monitoring Phase 1 and Phase 2, the adjustment range of operational pulses from the adaptive pulse generator, and the random variation of the memristor's resistance states. In our study, for different voltage combinations $\{V_{cond}, V_{SET}\}$, the adjustment range of voltage amplitudes is $\pm 20\%$ and 50mV stride for V_{cond}. We assume R_{HRS} and R_{LRS} to have the worst programming predictability according to the experimental results in [15], i.e., $\pm 10\%$ random variation for R_{LRS}, and $\pm 40\%$ random variation for R_{HRS}, respectively.

Fig. 8 (a) shows the detection margin in IMPLY monitoring Phase 1. The upper bound of detection margin can be determined by input Case 1 with the worst R_{HRS}, as depicted in Fig. 4 (a) ((i.e., $R_P = R_{HRS\_min}$, $R_Q = R_{HRS\_max}$)), and the lower bound of detection margin can be determined by input Case 3 (i.e., $R_P = R_{LRS\_max}$, $R_Q = R_{HRS\_min}$). Similarly, the limitation of detection margin in Phase 2 can

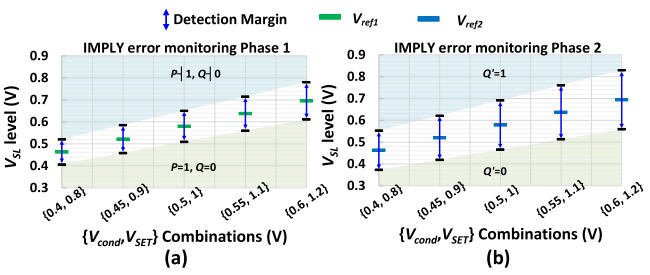

Fig. 8. Distribution of V_{SL} due to different input cases for the IMPLY operation for different voltage combinations $\{V_{cond}, V_{SET}\}$ in IMPLY monitoring: (a) Phase 1 and (b) Phase 2. The Detection margins—blue arrows and the reference voltages (V_{ref1} and V_{ref2}) for the comparator are demonstrated. The adjustment range of operational pulses is ±20%, and the random variation of memristors' resistance states for R_{LRS} and ±10% random variation for ±40% random variation for R_{HRS}.

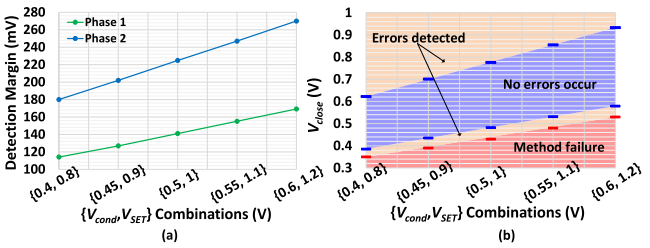

Fig. 9. (a) Detection margin for the comparator under different voltage combinations $\{V_{cond}, V_{SET}\}$ in IMPLY monitoring Phase 1 and Phase 2; The adjustment range of operational pulses is ±20%, and the random variation of memristors' resistance states for R_{LRS} and ±10% random variation for ±40% random variation for R_{HRS}. (b) The correct functional range of detection circuit under different V_{close} for the different voltage combinations $\{V_{cond}, V_{SET}\}$. The blue area indicates that no error occurs, the orange areas indicate the IMPLY operation failure that can be detected, and the red area indicates the functional failure of monitoring circuit.

also be evaluated by considering the worst case of resistance state value, as depicted in Fig. 8 (b). In addition, adjustable reference voltages are required to keep the V_{ref} on the midpoint of the detection margin, with the aim of a reliable output of comparator.

As shown in Fig. 9 (a), the detection margin is always above 114 mV in Phase 1 and 180 mV in Phase 2, respectively. This detection margin will be used as a key metric in the comparator design. More importantly, the offset voltage of comparators (refer to V_{offset} in Fig. 4) is an essential factor which can reduce the detection accuracy of the proposed scheme, so a systematic analysis of all the possible types of detection failures affected by the offset voltage is carried out:

1) Missing event of the operation failure, which can be measured by missing rate.

2) Incorrect identification of the IMPLY failure type, which can be measured by Error-Identification-Failure rate.

3) Overkill event of the correct operation, which can be measured by overkill rate.

For ease of understanding, the detection failure of Type-IV failure (i.e., unexpected SET on Q in IMPLY Case 3) as an example is discussed in detail:

1) Firstly, Case 3 will be identified as Case 1 mistakenly when the V_{offset} is larger than the half of detection margin in

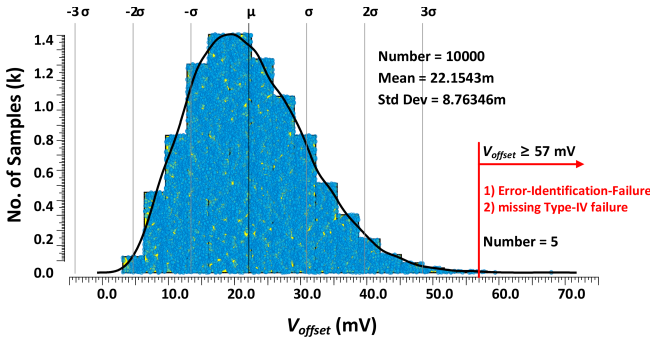

Fig. 10. Monte-Carlo simulation with 10k samples for comparator's offset voltage.

Phase 1 (refer to Fig. 6). Type IV failure will be missed when the identification error happens, which is regarded as missing of Type-IV failure.

2) Secondly, correct Case 3 operation with an identification error in Phase 1 will be treated as Type-III failure (i.e., SET failure on Q in Case 1) mistakenly if the V_{offset} is lower than the half of the detection Margin of Phase 2, which is regarded as Error-Identification-Failure.

3) Thirdly, correct Case 3 operation without an identification error will be treated as Type-IV failure if the V_{offset} is larger than the detection Margin of Phase 2, which is regarded as overkill.

To evaluate the detection accuracy of the proposed monitoring scheme, we analyze the V_{offset} of the conventional two-stage open-loop comparator without any optimization in our study. In our simulation, the ICMR is 0.05 V∼1.6 V, which has covered the three reference voltages as required by our proposed scheme. Fig. 10 shows the result of 10k-point Monte-Carlo simulation of the comparator's V_{offset}. Only 5 points of the 10k points exceed the half of the minimum detection margin, i.e., 57 mV (note the maximum offset value is 68 mV in this simulation), which will cause the Error-Identification-Failure of IMPLY Case 3 and missing of Type-IV failure. This result means that the detection accuracy can reach up to 99.95%. In addition, we have also reviewed the comparator designs in 0.18-μm technology from the literature. The simulated maximum offset voltage of a state-of-the-art comparator design in [16] is 1.6 mV, and the maximum value of measured offset voltage of comparator products in [17] is 50 mV. Both the offset results are better than that of the comparator design in our study. In other words, the Monte-Carlo simulation results in our study exhibits that the proposed in-situ monitoring circuit employing a comparator is an effective method to detect the major operation failures due to the aging-induced threshold voltage degradation for IMPLY-based CIM systems.

Although encouraging, the memristors' unexpectedly SET (i.e., unexpected SET on Q in Case 1 and Case 3) may happen in IMPLY monitoring Phase 1 when the V_{close} is drifting too far from the nominal value, which makes the proposed scheme failure, i.e., identification failure. Fig. 9 (b) shows the correct functional range of monitoring scheme under the different voltage combinations $\{V_{cond}, V_{SET}\}$, where the boundary of

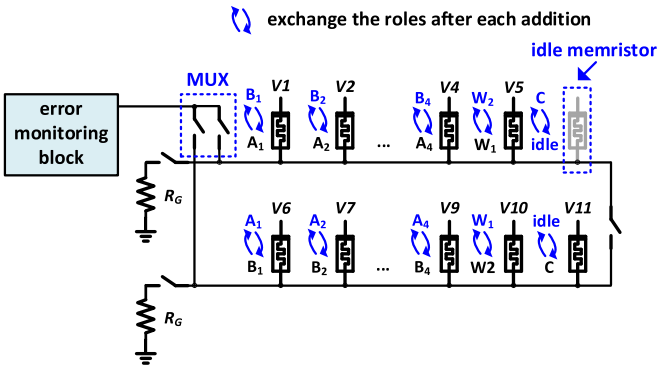

Fig. 11. Implementation of the proposed monitoring circuit in semi-parallel adder with improved/novel operation scheme for Aging-matching; the unused memristor within the memristor array is idle during addition process, and the monitoring circuit can be connected to the two rows through switches.

method failure is determined by input Case 3 due to the higher possibility of unexpected SET on Q. Notably, V_{open} is not covered in Fig. 9 (b) according to the simulation that unexpected RESET of the memristors and corresponding identification failure does not happen in Phase 1. The margin for correct detection is above 50 mV for a decreased V_{close}. It is necessary to monitor errors in the IMPLY operation and adjust the voltage combinations over time with a rational frequency.

III. Case Study: Error Monitoring of IMPLY-Based Adder for BNN Convolution Operations

Adders and multipliers are primary building blocks in digital circuit systems including the machine learning based AIoT systems and have become increasingly important given the widespread usage of successful Deep Neural Networks. To verify the feasibility and efficiency of our proposed scheme in more practical operations, a case study of error monitoring scheme design of 4-bit IMPLY-based adder for memristive CIM system is carried out, as the IMPLY logic is more suitable to adder instead of multiplier requiring the much longer operation steps [18]. Moreover, the IMPLY-based memristive CIM is a promising low-power solution to implement simple neural network models, including the binary neural network (BNN) and ternary neural network that only have XOR and add operations instead of multiplication operations. Hence, a further analysis of the proposed error monitoring is performed on a VGG-11 based BNN that is a hardware-efficient model to perform cifar-10 classification task very well [13].

A. Error Monitoring Scheme Analysis of IMPLY-Based Adder

1) Implementation of the Proposed Monitoring Scheme for IMPLY-Based Full Adder: Our proposed monitoring circuit as a built-in circuit can be implemented in various IMPLY-based adder. We use a state-of-art IMPLY-based adder [12] to evaluate the feasibility of our proposed in-situ error monitoring scheme. Fig.11 shows the implementation of the proposed monitoring circuit in IMPLY-based semi-parallel adder. The monitoring circuit is connected to the memristor cells of each

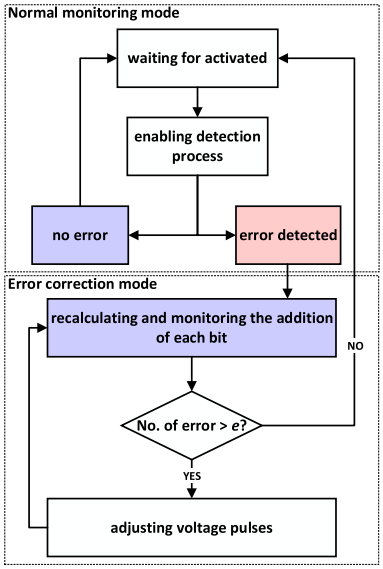

Fig. 12. Flow chart of error monitoring process: When an error is detected, the monitoring circuit enters the error correction mode and returns to the normal operation mode after error correction.

row through a multiplexer. In this 4-bit semi-parallel adder, 11 memristors are needed to finish the entire addition process, where memristors A and B as input memristors are written two 4 bits input operands respectively, memristor W_1 and W_2 as work memristors are used to deliver the intermediate results, while carry memristor C are used in the addition of each bit to deliver the bit of carry out or carry in. When the detection circuit is activated, it can select one of the rows through the multiplexer, to detect a specific IMPLY or FALSE operation in the adder operation process without interrupting the current addition operation.

Fig.12 shows the flow chart of one monitoring process including normal detection mode and error correction mode. When an error is detected, the monitoring circuit enters the error correction mode and then returns to the normal operation mode after error correction. When the monitoring circuit is disabled, it is in the sleep state. When activated, the monitoring circuit starts to work in the normal monitoring mode, and continuously monitors the target memristor which is currently performing addition process. If no error is detected, the circuit will return to sleep state and wait for the next activation, otherwise the detection circuit will enter error correction mode. In the error correction mode, the detection circuit will recalculate the current addition and detect every IMPLY operation that may have a SET process.

To evaluate the validity of the monitoring scheme, we should consider the operation step that is most likely to be input Case 1 and Case 3 for error detection. Fig.13 shows the possibility of switching in 1-bit addition operational steps through the simulation. It should be pointed that we select three representative memristors as illustrative examples for the analysis, i.e., input memristor A, work memristor W_2 and carry memristor C.

For work memristor W_2, after FALSE operation of the W_2 in step 1 and step 11 as shown in Fig. 13 (b), it will be

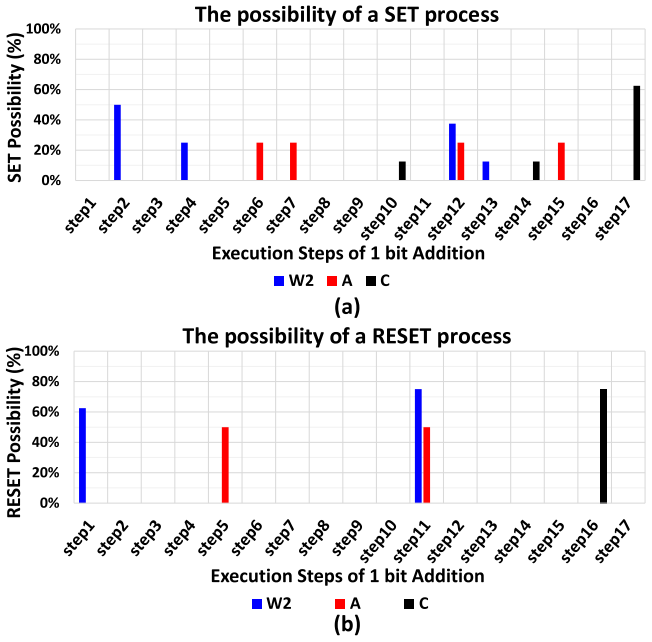

Fig. 13. The possibility of state switching of work and input memristor in 1-bit addition operational steps for: (a) SET; (b) RESET. semi-parallel adder needs 17 steps to finish addition of each bit.

TABLE IV

THE AVERAGE NUMBER OF SET OR RESET OPERATIONS REQUIRED FOR EACH MEMRISTOR TO FINISH N-BIT ADDITION*

memristors	Number of SET	Number of RESET
a_i	1	1
b_i	0.25	0
w_1	$1.25n$	$1.25n$
w_2	$1.375n$	$1.375n$
c	$0.625n$	$0.75n$

* Assuming that the possibility of each input case is the same.

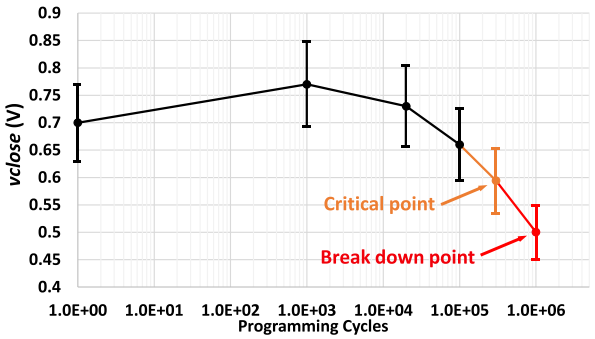

Fig. 14. Derived curve of aging-induced degradation of V_{close} from [8]: the breakdown of memristor will happen after 10^6 cycles.

used as the input Q of IMPLY operation with Case 1 or Case 3 in the subsequent step (i.e., step 2 and step 12, as shown in Fig. 13 (a)). In the common monitoring mode, Step 2 of each bit addition will be monitored. If operation failure happens, the monitoring circuit enters error correction mode, where step 2 and step 12 of each bit addition are monitored, with the aim of determining the number of operation failures. The voltage combinations adjustment is required when the number of operation failure exceeds the predefined threshold e.

Similarly, adders' input memristors A will perform FALSE operation in Step 5 and Step 11, as shown in Fig. 11 (a), and the next step will also be IMPLY operation with Case 1 or Case 3. The aging degradation of the input memristors A can be detected by monitoring this operation directly. The carry memristors of adder (i.e., C) will perform FALSE operation in step 16, and the step 17 will also be IMPLY operation with Case 1 or Case 3.

Compared with the 17-step normal addition process of semi-parallel full adder, the delay of addition process with the activated monitoring circuit will increase by 5.8% against the normal monitoring mode, due to one extra step for IMPLY monitoring.

2) Aging-Matching Scheme for Semi-Parallel Full Adder: After analyzing the detection schemes for several main memristor, we estimate the trend of operation times including SET and RESET process of each memristor in the addition process. Through the simulation, we discover that memristors with different functions can be divided into two types according to their operation times, i.e., input memristors and others including work and carry memristors. The bit-width of the adder does not affect the input memristors, while the number of operations of other memristors has a linear relationship with the bit-width of the adder, as listed in Table IV. When the addition is performed a certain number of times, the degree of mismatch between memristors (due to mismatched aging) tends to increase, which leads to operation failure during IMPLY operation.

In our study, we use the curve of device aging-induced threshold voltage degradation derived from [8], as shown in Fig. 14. Experimental results in [8] shows that the breakdown of the devices is more closely related to the V_{close}. The V_{close} will increase when the programming cycles are near 10^3 times. When degradation phenomenon happens after 10^5 cycles, V_{close} continue to decrease until the device can no longer be RESET (demonstrated in [8]), and the device will breakdown. This drift trend makes both Type-III and Type-IV failure may happen during addition operations, which exacerbates the impact of mismatch between the devices, and optimization is required to reduce the mismatch of memristors.

Operations scheme for semi-parallel adder can be optimized in a simple manner. Note that each row of memristor contains input memristor and work memristor, while the memristor in the same column as carry memristor is idle, as shown in Fig. 11. Without changing the operation steps of the full adder, the aging degree of each memristor in this full adder can be matched and unified as much as possible by only exchanging the functions of the two rows of memristors alternately after each addition. Specifically, the input memristor A can act as input memristor B in the next addition.

Fig. 15 shows the trend of memristors' operation times without Aging-matching scheme, five different trends can be found in both SET operations and RESET operations.

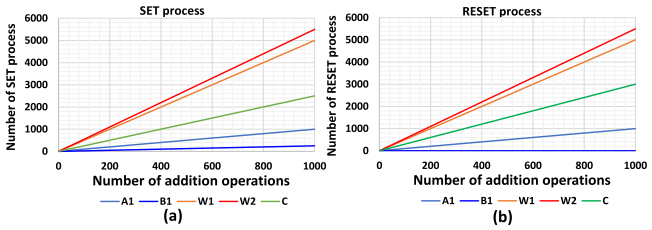

Fig. 15. The average number of switching events required for each memristor to finish 4-bit addition for (a) SET and (b) RESET operations, assuming that the probability of each input cases is the same.

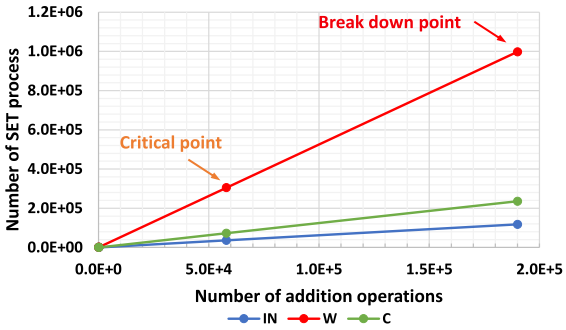

Fig. 16. The number of SET operations required for each memristor to finish 4-bit addition under optimized operation scheme.

Conversely, Fig. 16 shows the results of Aging-matching scheme of the adder, where input memristor A together with B can be unified as IN, and W_1 together with W_2 can be unified as W. 40% reduction on the trend of aging can be achieved by Aging-matching scheme, as compared with the original operation scheme.

3) Case Study of Aging-Matching Scheme With Error Detection: Combining with Fig.14 and Fig. 16, we propose a scheme to adaptively adjust the voltage pulse. Fig. 17 shows the flow chart of our specific detection scheme. Combined with the derived aging characteristics of the device, we design a complete detection scheme for the full adder. The monitoring circuit will be activated more frequently at several critical points of memristor degradation as shown in Fig. 14, and it is most likely that the operating voltage pulse needs to be adjusted adaptively near these critical points.

The threshold voltage of the work memristor W firstly drifts to the maximum value after 10^3 SET process when 200 additions are performed. The {0.5 V, 1 V} of the voltage combination $\{V_{cond}, V_{SET}\}$ will be chosen at first. In this stage, the monitoring circuit will focus on detecting the memristor W and IN alternately with a higher frequency before 200 addition operations (refer to Fig. 17), with the aim of detect the Type-III failure on the memristor W. The voltage combination may change to {0.55 V, 1.1 V} when the Type-III failure on W has been detected.

After 200 addition operations, the possibility of operational failure is relatively low. In this stage, the monitoring circuit detects the W at a relatively low frequency until 56000 additions are performed (e.g., monitoring circuit is activated per 5000 additions).

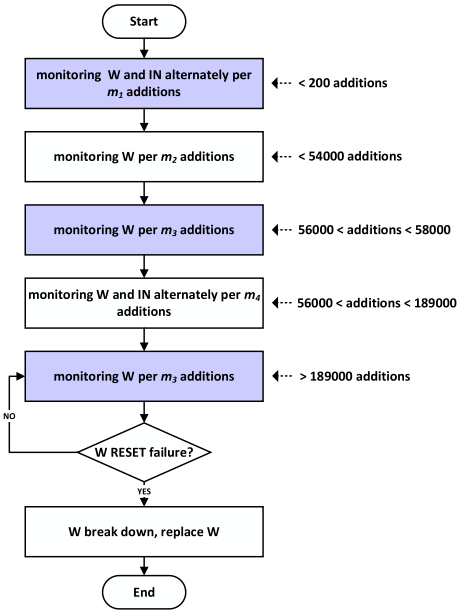

Fig. 17. Flow chart of error monitoring scheme for IMPLY-based semi-parallel adder: the monitoring circuit will be enabled more frequently when the degree of aging is approaching critical points.

After 56000 addition operations, since the number of SET process on W is close to 3×10^5, Type-IV failure related to W may happen at the critical point (refer to Fig. 14 and Fig. 16). In this stage, the monitoring circuit detects the addition operation of work memristor W at a high frequency until the Type-IV failure of W has been detected and adjust the voltage combinations to {0.5 V, 1 V} adaptively. In-situ RESET monitoring on W will be performed simultaneously to detect non-ideal RESET during FALSE operation.

Before the break down point of W reached (i.e., 190000 additions, refer to Fig. 16), the monitoring circuit can be activated at a low frequency because of the lower error rate (e.g., per 2500 additions). In-situ RESET monitoring on W will be performed to detect non-ideal RESET during FALSE operation.

When the number of addition operations is approaching 190000, the W is close to the break down point. Frequent monitoring process will be performed (e.g., activated per 100 additions). In this stage, the monitoring circuit detects the addition operation of the W at a high frequency until the Type-IV failure on the W has been detected and the voltage combination is adjusted to {0.45 V, 0.9 V} adaptively. When the RESET failure on W has been detected, work memristors should be replaced by other idle memristors, and the aging condition of adder should be estimated again to refresh the monitoring scheme. Under this adaptive monitoring scheme, the total number of error detection when the devices break down is about 270 times.

When activated, the average power consumption of the error detection circuit is 4.9 μW. Compared with the normal addition process, the power consumption of 4-bit addition with the proposed error monitoring mechanism is increased by 2.5%, i.e., from 21.894 μW to 22.447 μW. Since the error monitoring mechanism is activated timely, as shown in the Fig. 17, the increase of power consumption will be negligible.

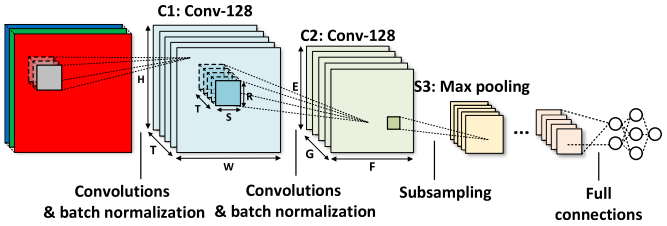

Fig. 18. Overview of a typical VGG-11 based BNN architecture: sequences of convolutional -subsampling layers perform hierarchical extraction.

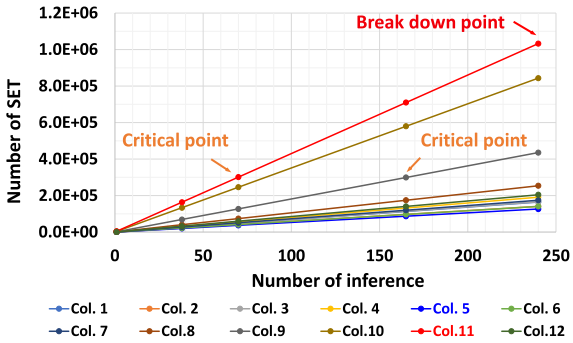

Fig. 19. The relationship between the number of SET process and the number of inferences on memristors in each column.

B. Error Monitoring Scheme Analysis for Binarized Neural Network

In section III. A, we discuss the proposed monitoring scheme in an IMPLY-based adder. In this part, we will further analyze inference process of a convolution filter in a BNN model [13] based on IMPLY-based CIM system, to estimate the potential improvement on the detection efficiency of our proposed scheme against the straightforward program-verify scheme as a case study.

Figure. 18 sketches an overview of a typical BNN architecture. A convolution layer, e.g., C1, transform input feature maps into output feature maps, each containing multiple units. Each unit in an output feature map ($E \times F \times 1$) is connected to local patches of units ($R \times S \times T$) in the input feature maps through a filter bank ($R \times S \times T \times G$). During inference process of BNN convolution layer, each channel of a filter ($R \times S \times 1$) slides on the input feature maps ($H \times W \times T$) to finish convolution operation, i.e., accumulating the XOR result of the 1-bit input vectors and 1-bit weight. Then a unit in an output feature map is obtained by adding up T partial sums from each channel. The required number of convolution operation of each channel can be calculated according to (9), where U is the stride of filter sliding.

$$\textit{Operations of conv} = \textit{Size of output feature maps} = E \times F,$$

where

$$E = (H - R + U)/U, \quad F = (W - S + U)/U \tag{9}$$

The size of a BNN convolution filter in C1 and C2 layer is fixed as $3 \times 3 \times 128$. In a typical application for cifar-10 classification, the size of the RGB input image is 32×32, which means that the size of the C1 convolutional layer is 30×30, and the size of the C2 convolutional layer is 28×28. Under this BNN model, the number of convolution operation required for a channel of C1 convolution filter is 784 according to (9). The C1 filtering during an inference process requires 784 XOR operations and 783 executions of accumulation. For the simplicity of a case study without loss of generality, we assume that the XOR and accumulation operations are sequentially performed with a minimized size of memristor array, i.e., 2×12, which is determined by an IMPLY based 10-bit semi-parallel adder for the accumulation of 784 XOR results. After one inference process, Row 1 and Row 2 of the adder can be exchanged alternately according to the proposed Aging-matching scheme illustrated in Section III. A. 2).

Fig. 19 shows the number of SET operations on memristors in each column during the BNN inference process of 1st convolution filter channel in C1 layer, where critical points of aging-induced degradation existing in the 70th and 165th inference process and corresponding to the 9th and 11th columns of memristors, respectively. When the number of inferences approaches the critical points of memristor aging degree, the detection circuit will be activated at a higher frequency (e.g., every 1 inference). Moreover, the break down point exists in the 240th inference and corresponds to the 11th column. Since there are no critical points existing in other inference processes, the monitoring circuit can be activated at the low frequency (e.g., every 10 inference). Under this adaptive monitoring scheme, the total number of error detection when the devices break down is about 45 times.

After the 240th inference, the 10th and 11th column memristors are close to the break down point. To further extend the lifetime of the full adder and maintain the reliability of the system, the 10th and 11th column memristors need to be replaced by other idle memristors due to the mismatch with most devices.

To show the improvement offered by the proposed error monitoring scheme against the straightforward program-verify in terms of detection efficiency, we compare the detection speed and total steps required to finish detection process in adder and BNN inference applications based on IMPLY-based CIM system, as listed in Table V. To objectively compare the detection efficiency as much as possible, we assume the program-verify process can be finished in the fewest steps (i.e., 2 steps for SET-verify and 2 steps for RESET-verify) without including the additional operations of intermediate data movement before the detection process, due to the interruption of normal logic operations. The detection frequency of the program-verify scheme in [9] is fixed at once per 4000 programming cycles of memristors. Therefore, according to the relationship between the number of memristor operations and the number of addition executions in Fig. 16, the detection frequency after using the program-verify scheme to this application can be evaluated, i.e., once per 700 additions. Similarly, the application of the program-verify method in BNN inference can also be evaluated from Fig. 19. The total steps of detection process are evaluated through the

TABLE V

COMPARISON OF MONITORING PERFORMANCE BETWEEN THE PROPOSED SCHEME AND CONVENTIONAL SCHEME

		Program-verify scheme	Proposed monitoring method	Improvement
4 bits IMPLY-based semi-parallel Adder	Steps for one detection process	4*	2 (normal detection mode)	-50%
	Detection frequency	Fixed (once per 700 additions)	adaptively	-
	Total steps of detection process **	1100 (estimated)	270 (estimated)	-75.5%
	Interrupt?	Yes	No	-
Inference of VGG-11 based BNN	Detection frequency	Fixed (once per inference)	adaptively	-
	Total number of detection process **	240	45	-74%
	Interrupt?	Yes	No	-

* At least 2 steps for SET-verify and 2 steps for RESET-verify process.
**The number of operations is evaluated until the breakdown of devices occurs.

multiplication of the steps for one monitoring process and total number of performing monitoring process until the memristor breaks down. In our estimation, 75.5% and 74% potential improvement on the detection speed can be achieved during the detection process, for the IMPLY-based adder and BNN model, respectively, as shown in Table V. Furthermore, our proposed scheme can effectively avoid the interruption of logic operations caused by the introduction of additional detection mode to the CIM systems, which is the major efficiency overhead of the program-verify scheme.

IV. CONCLUSION

This paper proposes a novel in-situ aging-aware error monitoring scheme for memristive IMPLY-based CIM system. Comprehensive and systematical analysis of operation failures on memristive IMPLY logic due to aging-induced threshold voltage degradation is conducted, and two major failures that cannot be addressed by the traditional guardband scheme is identified. The proposed in-situ error monitoring scheme can achieve up to 50% reduction in detection steps as compared to the straightforward program-verify monitoring scheme. Simulation results under Monte-Carlo simulation of the key detection circuit show that the proposed monitoring scheme can effectively detect the major operation failures existing in IMPLY and FALSE operations with a detection accuracy up to 99.95%. A case study of error monitoring scheme design for an IMPLY-based adder is carried out. The analysis result exhibits that the proposed in-situ error monitoring scheme can achieve 75.2% improvement on the detection speed against the conventional program-verify scheme. Further analysis shows that an 74% improvement on the detection speed during the inference process of VGG-11 BNN model on an IMPLY-based memristive CIM system can also be achieved. This study opens a new research direction on how to improve the reliability of memristive CIM systems by addressing the aging-induced device degradation issues from the circuits and systems design perspective.

REFERENCES

[1] A. Sebastian *et al.*, "Memory devices and applications for in-memory computing," *Nat. Nanotechnol.*, vol. 15, pp. 529–544, Jul. 2020, doi: 10.1038/s41565-020-0655-z.

[2] D. Ielmini and H. S. P. Wong, "In-memory computing with resistive switching devices," *Nature Electron.*, vol. 1, no. 6, pp. 333–343, 2018, doi: 10.1038/s41928-018-0092-2.

[3] J. Borghetti *et al.*, "Memristive' switches enable 'stateful' logic operations via material implication," *Nature*, vol. 464, pp. 873–876, Apr. 2010, doi: 10.1038/nature08940.

[4] S. Kvatinsky, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser, "Memristor-based material implication (IMPLY) logic: Design principles and methodologies," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 22, no. 10, pp. 2054–2066, Oct. 2014, doi: 10.1109/TVLSI.2013.2282132.

[5] T. Zanotti, F. M. Puglisi, and P. Pavan, "Smart logic-in-memory architecture for low-power non-von Neumann computing," *IEEE J. Electron Devices Soc.*, vol. 8, pp. 757–764, 2020, doi: 10.1109/JEDS.2020.2987402.

[6] T. Zanotti, F. M. Puglisi, and P. Pavan, "Reliability-aware design strategies for stateful logic-in-memory architectures," in *IEEE Trans. Device Mater. Rel.*, vol. 20, no. 2, pp. 278–285, Jun. 2020, doi: 10.1109/TDMR.2020.2981205.

[7] A. Grossi *et al.*, "Electrical characterization and modeling of 1T-1R RRAM arrays with amorphous and poly-crystalline HfO_2," *Solid-State Electron.*, vol. 128, pp. 187–193, Feb. 2017, doi: 10.1016/j.sse.2016.10.025.

[8] D. A. Robayo, G. Sassine, Q. Rafhay, G. Ghibaudo, G. Molas, and E. Nowak, "Endurance statistical behavior of resistive memories based on experimental and theoretical investigation," *IEEE Trans. Electron Devices*, vol. 66, no. 8, pp. 3318–3325, Aug. 2019, doi: 10.1109/TED.2019.2911661.

[9] S. Ning, T. O. Iwasaki, and K. Takeuchi, "50 nm Al_xO_y ReRAM program 31% energy, 1.6× endurance, and 3.6× speed improvement by advanced cell condition adaptive verify-reset," *Solid-State Electron.*, vol. 103, pp. 64–72, Jan. 2015, doi: 10.1016/j.sse.2014.10.003.

[10] G. Wang *et al.*, "Improving resistance uniformity and endurance of resistive switching memory by accurately controlling the stress time of pulse program operation," *Appl. Phys. Lett.*, vol. 106, no. 9, Mar. 2015, Art. no. 092103, doi: 10.1063/1.4907604.

[11] J. Xu, Y. Zhan, Z. Wang, G. Yu, Y. Li, and C. Wang, "A novel variation-aware error monitoring scheme for memristor-based material implication logic," in *Proc. IEEE Int. Conf. Integr. Circuits, Technol. Appl. (ICTA)*, Nov. 2020, pp. 110–111, doi: 10.1109/ICTA50426.2020.9332122.

[12] S. G. Rohani, N. Taherinejad, and D. Radakovits, "A semiparallel full-adder in IMPLY logic," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 1, pp. 297–301, Jan. 2020, doi: 10.1109/TVLSI.2019.2936873.

[13] I. Hubara *et al.*, "Binarized neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Barcelona, Spain, Dec. 2016, pp. 1–9.

[14] S. Kvatinsky, E. G. Friedman, A. Kolodny, and U. C. Weiser, "TEAM: Threshold adaptive memristor model," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, no. 1, pp. 211–221, Jan. 2013, doi: 10.1109/TCSI.2012.2215714.

[15] T. Zanotti *et al.*, "Reliability of logic-in-memory circuits in resistive memory arrays," *IEEE Trans. Electron Devices*, vol. 67, no. 11, pp. 4611–4615, Nov. 2020, doi: 10.1109/TED.2020.3025271.

[16] J. Jung, I.-H. Kim, S.-J. Kim, Y. Lee, and J.-H. Chun, "A 1.08-nW/kHz 13.2-ppm/°C self-biased timer using temperature-insensitive resistive current," *IEEE J. Solid-State Circuits*, vol. 53, no. 8, pp. 2311–2318, Aug. 2018, doi: 10.1109/JSSC.2018.2824307.

[17] D. Analog. (2009). *Op Amp Input Offset Voltage*. Dostupné Z. [Online]. Available: http://www.analog.com/static/imported-files/tutorials/MT-037.pdf

[18] D. Radakovits, N. Taherinejad, M. Cai, T. Delaroche, and S. Mirabbasi, "A memristive multiplier using semi-serial imply-based adder," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 5, pp. 1495–1506, May 2020, doi: 10.1109/TCSI.2020.2965935.

Jiarui Xu (Member, IEEE) was born in Gansu, China, in 1996. He received the B.Eng. degree in microelectronics engineering from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2019, where he is currently pursuing the master's degree with the School of Optical and Electronic Information.

His research interests include non-volatile computing-in-memory, energy-efficient VLSI architecture design and ultra-low-voltage circuit design for machine learning, and neuromorphic computing.

Yi Zhan (Member, IEEE) was born in Hubei, China, in 1998. He received the B.Eng. degree in electronics engineering from Northwestern Polytechnical University, Xi'an, China, in 2019. He is currently pursuing the master's degree with the School of Optical and Electronic Information, Huazhong University of Science and Technology (HUST), Wuhan, China.

His research interests include energy-efficient digital signal processor design and ultra-low-voltage circuit design for biomedical/healthcare, wireless sensor, neuromorphic computing, and robot applications.

Yujie Li (Member, IEEE) was born in Jiangsu, China, in 1998. He is currently pursuing the master's degree with the School of Optical and Electronic Information, Huazhong University of Science and Technology (HUST), Wuhan, China.

His research interests include energy-efficient VLSI architecture design and ultra-low-voltage circuit design for machine learning, neuromorphic computing, and robot applications.

Jiajun Wu (Member, IEEE) was born in Fujian, China, in 1999. He is currently pursuing the bachelor's degree with the School of Optical and Electronic Information, Huazhong University of Science and Technology (HUST), Wuhan, China.

His research interests include energy-efficient VLSI architecture design and ultra-low-voltage circuit design for machine learning, neuromorphic computing, and robot applications.

Xinglong Ji received the Ph.D. degree from the Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences in 2016.

He was with the Department of Engineering Product Development, Singapore University of Technology and Design. He is currently with the Center for Brain-Inspired Computing Research, Tsinghua University. His current research interests include non-volatile memory technologies and emerging nanoelectronics for neuromorphic applications.

Guoyi Yu (Member, IEEE) received the B.Eng., M.Eng., and Ph.D. degrees in electronics engineering from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2000, 2003, and 2006, respectively.

Since 2007, he has been with the Department of Electronic Science and Technology, HUST, as a Lecturer. He is currently an Associate Professor with the School of Optical and Electronic Information, HUST. His research interests include analog-mixed signal circuit design, sensor interface circuit design, and heterogeneous 3D-IC design for biomedical/healthcare, wireless sensor, and wearable and robot applications. He is a Treasurer of IEEE CASS-EDS-SSCS Wuhan Joint Chapter. He is an Active Reviewer of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS, IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS, IEEE ACCESS , and *Microelectronics Journal* (Elsevier).

Wenyu Jiang (Senior Member, IEEE) received the Ph.D. degree in computer science from Columbia University, New York City, in 2003.

From 2003 to 2011, he was with Dolby Laboratories, as a Staff Engineer, and researched on quality of service for streaming media over wireless, P2P-based digital rights management (DRM) friendly content distribution, and multimedia fingerprinting and retrieval. Since 2011, he has been with the Institute for Infocomm Research, A∗STAR. He is currently a senior scientist. His research areas span from memory computing-based multimedia retrieval and fuzzy pattern search, signal processing and applications for fiber Bragg gratings (FBG) sensors to neuromorphic computing, including spiking neural network algorithms and in-memory computing. He is the Vice Chair of IEEE SSCS Singapore Chapter.

Rong Zhao (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the National University of Singapore in 1999.

She had been worked as a Senior Scientist, the Principle Investigator, and an Assistant Division Manager of the Data Storage Institute, A∗STAR. She has been an Associate Professor in engineering product development with the Singapore University of Technology and Design (SUTD). She is currently a Professor with the Department of Precision Instruments, Tsinghua University, the Center for Brain-Inspired Computing Research, Tsinghua University, and the Innovation Center for Future Chip, Tsinghua University. She has authored or coauthored over 100 publications in international journals and international conference proceedings. Her main research interests include non-volatile memories, such as phase-change memory and resistive memory, and reconfigurable devices covering from material synthesis, device design/fabrication, to chip design/prototyping. More recently, she has broadened her activities, entering the field of artificial cognitive memory and energy harvesters. She is the Co-Chair of the Technical Committee of IEEE Non-Volatile Memory Technology Symposium for the period 2012–2014 and the Lead Organizer of the Material Research Society in 2011 and 2012 spring meetings of Phase Change Symposium.

Chao Wang (Senior Member, IEEE) received the B.Eng. degree in electronics engineering from the Huazhong University of Science and Technology (HUST), China, in 2000, and the Ph.D. degree in electronics engineering from Nanyang Technological University (NTU), Singapore, in 2008.

From 2005 to 2007, he was with the Center for Signal Processing, NTU, as a Research Engineer. From 2008 to 2010, he worked as an IC Design Engineer with the Asia-Pacific Design Centre, STMicroelectronics, Singapore. From 2010 to 2017, he was a Research Scientist and the Project Leader of the Institute of Microelectronics (IME), Agency for Science, Technology and Research (A∗STAR), Singapore, where he led projects on wearable medical devices for cardio-engineering applications, and MEMS accelerometer ASICs for medical and navigation applications. He is currently a Professor with HUST and the Wuhan National Laboratory of Optoelectronics (WNLO), Wuhan, China. He participated in the development of STM Bayer/YUV CMOS image sensing and processing ICs/SoCs. His major research interests include energy-efficient digital signal processor design, ultra-low-voltage circuit design, MEMS sensor ASIC design, and heterogeneous 3D-IC design, especially for biomedical/healthcare, wireless sensor, the IoT, neuromorphic computing, and robot applications.

Dr. Wang has served as a TPC member for a number of international devices, circuits and systems conferences. He used to serve as a Committee Member, the Vice Chair, and the Chair for IEEE SSCS Singapore Chapter. He is the Chair of IEEE CASS-EDS-SSCS Wuhan Joint Chapter. He used to serve as a Guest Editor for *Journal of Sensors* (Hindawi) in 2016 and IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS in 2019. He is an Associate Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS, *IEEE Circuits and Systems Magazine*, IEEE ACCESS , and *Microelectronics Journal* (Elsevier).