A Universal Error Correction Method for Memristive Stateful Logic Devices for Practical Near-Memory Computing

*Jae Hyun In, Young Seok Kim, Hanchan Song, Gwang Min Kim, Jangho An, Jae Bum Jeon, and Kyung Min Kim**

Memristive stateful logic allows complete in-memory computing and is considered to be a next-generation computing technology for low power edge applications. Since the first stateful IMP gate was proposed in 2010, few studies have yet addressed the operating reliability issues that should be resolved before the technology is practically realized. Herein, a feasible near-memory error correction method for a typical bipolar-type memristor stateful logic system is proposed. An error correction principles using a HfO_2-based crossbar array device is explained, and two types of error correction methods checking if the number of FALSE data is zero and if the number of TRUE data is odd are proposed. Although the error correction modules require additional circuits and processing time, the resulting computing efficiency is comparable with conventional stateful logic techniques. Its application with a one-bit full adder is demonstrated and its feasibility for practical stateful logic devices is validated.

1. Introduction

Memristive stateful logic is a technology capable of performing a Boolean logic operation inside the memory array.[1–9] In the ideal stateful logic, the inputs and outputs are the resistance values of the memristor cells at each cross point in the crossbar array, and data are not sent outside of the memory array during operation. This format is known as complete in-memory computing.[10–13] Recent studies have identified various useful and efficient stateful gates for better computing efficiency, and as a result, stateful logic technology has advanced significantly.[14–24]

Such various gates are possible by simultaneously applying designed operating voltages on the multiple cells. In general, the word lines of input cells and output cells are biased to a conditioning voltage (V_{COND}) and programming voltages (V_{PGM}), respectively. The amplitudes of V_{COND} and V_{PGM} are relatively below and exceed the switching voltage. In addition, the shared bit line can be either biased or grounded or floated. Once the designed voltages have been correctly applied, the output cell is conditionally set- or reset-switched in accordance with the state of the inputs. When the switching voltages and resistance values of the memristor cells are given, one can calculate the range of the V_{PGM} and V_{COND}, which are dependent on the type of gates.

The presence of operating voltage range, i.e., the maximum and minimum values of the V_{PGM} and V_{COND}, can be understood as follows. The use of a higher V_{PGM} amplitude can increase the switching probability of the output cell. However, it may also increase the probability of unintentional set or reset switching of the output cell under nonswitching input conditions, which limits the increase in the V_{PGM}.[25,26] The input cells need to remain in their original states after the gate operation, which limits the maximum V_{COND} on the input cells. Also, if the V_{COND} is too low to inhibit the input cells strongly, their logical conditions cannot affect the conditional switching of the output cell.

Meanwhile, the switching voltage of the memristor cells has inherent cell-to-cell or cycle-to-cycle variations, due to the stochastic nature of the memristive switching.[16,22,27–30] If the variation of switching voltage is more severe than the range of operating switching voltage, it can result in an error output, which makes the gate operation unpractical. In our previous study, we proposed a methodology to evaluate and quantify the tolerance of the stateful logic gates against the variation in switching voltage.[31] The study revealed that 2NOT, 3NOR, and 2IMP gates (note that these gates are categorized to NOR-type gates in this study) are more tolerant than others so that their use was more practically feasible. The 4CARRY and 5SUM gates, which execute the carry and sum operations in one voltage clocking each, were proposed by Sun et al. to be the most innovative gates, but that seems unpractical considering their tight switching voltage variation requirements.[32]

Considering the memristor's inherent stochastic switching characteristic, most of the stateful logic gates will suffer from unintentional operating error, although the degree of these

J. H. In, Y. S. Kim, H. Song, G. M. Kim, J. An, J. B. Jeon, Prof. K. M. Kim
Department of Materials Science and Engineering
KAIST Advanced Institute of Science and Technology (KAIST)
291 Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea
E-mail: km.kim@kaist.ac.kr

The ORCID identification number(s) for the author(s) of this article can be found under https://doi.org/10.1002/aisy.202000081.

DOI: 10.1002/aisy.202000081

errors will be different gate by gate. In conventional memory devices, error correction functions are essential parts of the device.[33–38] Similarly, an appropriate error correction method must be introduced to fix the unavoidable errors in the stateful logic device. Such a solution has not yet been reported.

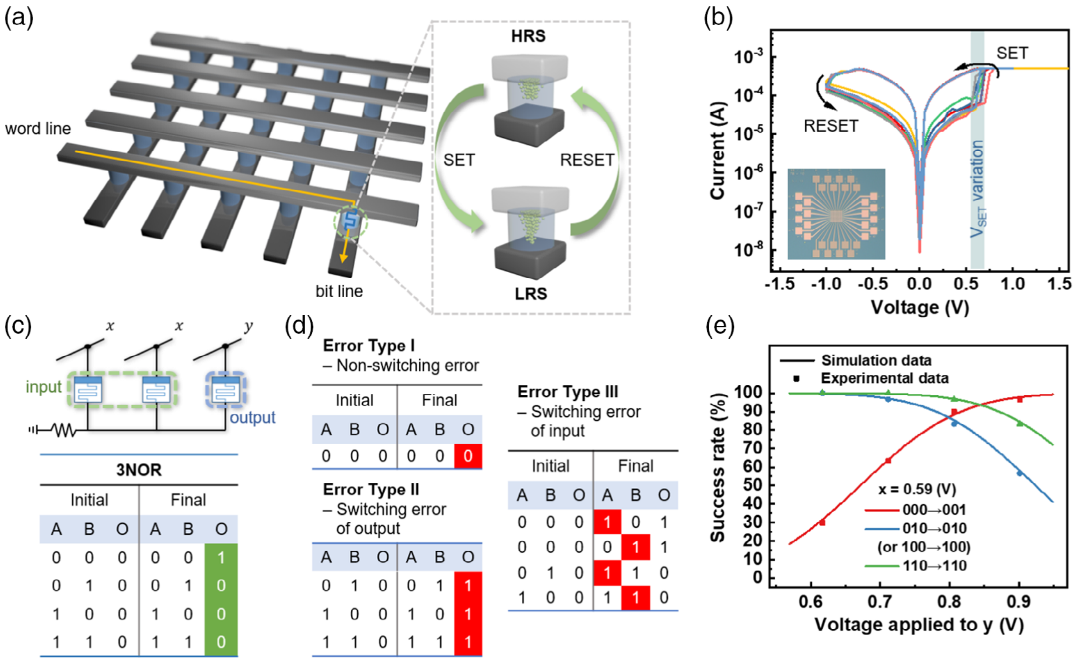
In this study, we propose a near-memory error correction method for stateful logic devices. We discuss possible error cases during gate operation and show how to control the probability of errors using a HfO_2-based crossbar array device. Then, we propose two error correction strategies, a zero FALSE counting method for the NOR-type gates and an odd TRUE counting method for the non-NOR-type gates. To apply the method, we introduce a transistor-based error correction module at the periphery of the crossbar array that receives the data and checks the errors, using the optimized error correction algorithm. Finally, we demonstrate how the error correction method can improve computing efficiency in a one-bit full adder operation, compared with the conventional method, despite its complicated configuration and operating sequence.

2. Error Control in NOR-type Gates by Adjusting Switching Voltage

Figure 1a schematically shows a crossbar array device, where a memristive oxide layer is sandwiched by word lines (top electrodes) and bit lines (bottom electrodes). Here, by selecting each word line and bit line, the target memristor cell can be set-switched to the low-resistance state (LRS) or reset-switched to the high-resistance state (HRS) with respect to a bias polarity. Figure 1b shows the resistance switching characteristic of the $Ta/HfO_2/Pt$ memristor used in this study. The device was integrated into a crossbar structure with a cross-sectional area of $5 \times 5\ \mu m^2$. The detailed device fabrication process is described in Experimental section. The inset shows a top view image of the device. The HfO_2-based memristor device is one of the most reliable memristive systems because of its high endurance and long retention.[39–41]

The device showed switching voltage variations on both the set and reset voltages, which can limit the practicality of the stateful logic device.[39,42–45] When switching voltage variation is present, as shown in Figure 1b, two types of error cases are possible; a nonswitching error in the switching condition, and an unintended switching error in the nonswitching condition. Examples of the error cases and their control are discussed using the 3NOR gate operation, which is representative of NOR-type gates. In this study, to specify the gate, we labeled each gate with a digit (2, 3, 4, or 5) followed by the name of the gate, where the digit refers to the number of cells needed for gate operation. For example, the 3NOR gate needs three cells (two inputs cells and output cell) to execute the NOR gate operation.

Figure 1c shows a standard operating unit for the 3NOR gate (top panel), showing two input cells (A and B) and one output cell (O). Their word lines are biased to x, x, and y, respectively. Here, x and y are the V_{COND} and the V_{PGM}, where $V_{COND} < V_{SET,min} < V_{SET,max} < V_{PGM}$. The $V_{SET,min}$ and $V_{SET,max}$ are the minimum and maximum set voltages of the device, with values of 0.65 and 0.75 V, respectively. The bottom panel shows the truth



3NOR

Initial			Final		
A	B	O	A	B	O
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	0	1	1	0

Error Type I
– Non-switching error

Initial			Final		
A	B	O	A	B	O
0	0	0	0	0	0

Error Type II
– Switching error of output

A	B	O	A	B	O
0	1	0	0	1	1
1	0	0	1	0	1
1	1	0	1	1	1

Error Type III
– Switching error of input

Initial			Final		
A	B	O	A	B	O
0	0	0	1	0	1
0	0	0	0	1	1
0	1	0	1	1	0
1	0	0	1	1	0

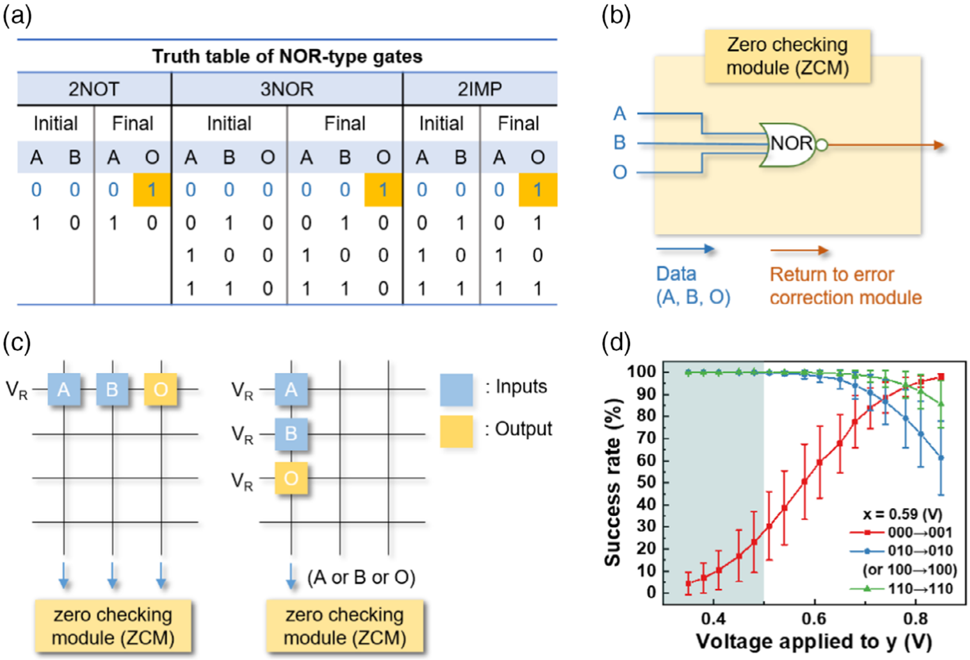
Figure 1. a) A schematic illustration of a memristive crossbar array used for stateful logic. The inset shows the typical operating mechanism of the Valence change mechanism (VCM) type bipolar memristor device. b) The *I–V* memristive characteristics of the $Ta/HfO_2/Pt$ device used in this study. The inset shows a top view image of an 8 × 8 crossbar array device. c) The top panel shows a schematic diagram of the 3NOR gate unit and its biasing condition. The bottom panel shows the truth table of the 3NOR gate operation. The final output in green refers to the successful operating output. d) All of the possible error cases during the 3NOR gate operation. Red boxes indicate the errors. e) Success rates of the 3NOR gate operation as a function of the V_{PGM} (y), when V_{COND} ($=x$) is 0.59 V, for various input cases. Dots are experimental data, and lines are simulation data based on the V_{SET} distribution.

table of the gate. For ideal NOR gate operation, the output cell should be set-switched when all of the inputs are FALSE (“0”). Figure 1d shows all of the error cases that are possible during the 3NOR gate operation. Due to the variation in switching voltage, unintended nonswitching of the output cell (Type I) or unintended switching of the output cell (Type II) or that of input cell (Type III) can occur. Those error cases are typically hard to manage together because their optimized conditions are competing. For example, if one increases the V_{PGM}, the nonswitching error rate will decrease, and the unintended switching error rate will increase.

In conventional approaches, to deal with errors, the operating voltage is chosen where the sum of the error rates is the minimum. However, for error correction, rather than reducing the total error rate, controlling the number of error cases is crucial. Figure 1e shows the success rate of the 3NOR gate operation as a function of y values when x is 0.59 V for various initial states. The x value is the optimized one high enough to act as the V_{COND} and low enough to avoid the unintended set switching of input cells ($x < V_{SET,min}$). Thus, Error Type III can be negligible, and only Error Type I and II are shown in Figure 1e. The symbols are the experimentally obtained success rates from the HfO_2 device, and the lines are simulation data based on the varying switching voltage of the device. For example, the red line and symbol show the success rate when initial “000” (A, B, and O) states convert to “001” states. When the two inputs are both TRUE (“1”), the output cell is rarely set-switched because the increased bit line potential from the two inputs reduces the applied voltage on the memristor cell (V_M). Therefore, at the given voltage ranges of x and y, keeping the output state in “0” is highly possible; its success rate can be 100%. When the two inputs are “10” or “01,” the increase in the bit line potential is less than that of the “11” input case. Thus, it results in a higher chance of unintended set-switching error in the output cell. In the HfO_2 device, the output cell can stably remain in “0” below a y of 0.6 V. Above that y value, the output cell can be stochastically switchable, and the success rate decreases. Therefore, to inhibit unintended switching errors for “01” inputs, the y value should be restricted to below 0.6 V. When the inputs are “00,” a y value of up to 0.94 V cannot guarantee the switching of the output cell because the V_M cannot be higher than the $V_{SET,max}$. Therefore, to inhibit unintentional nonswitching errors at (00) inputs, the y value should be higher than 0.94 V. However, in that condition, unintentional set-switching errors at (10) or (01) inputs are unavoidable. As such, by controlling the applied voltages, one can limit the error cases to some degree.

3. Error Correction of NOR-Type Gates by Zeros Counting

Controlling the error case is crucial for error correction. When the error case is limited to a specific input case, its correction can be more comfortable, even though the total error rate is higher than the sum of multiple error cases. For example, if the nonswitching error of the “000” states is the only error case by selecting $x = 0.59$ V and $y = 0.62$ V in Figure 1e, one can focus on it and neglect other errors. This method can be applied to the NOR-type gates: 3NOR, 2IMP, and 2NOT. **Figure 2**a shows a



Truth table of NOR-type gates

2NOT				3NOR						2IMP			
Initial		Final		Initial			Final			Initial		Final	
A	B	A	O	A	B	O	A	B	O	A	B	A	O
0	0	0	1	0	0	0	0	0	1	0	0	0	1
1	0	1	0	0	1	0	0	1	0	0	1	0	1
				1	0	0	1	0	0	1	0	1	0
				1	1	0	1	1	0	1	1	1	1

Figure 2. The ZCM for NOR-type gate correction. a) The truth table of the NOR-type gates: 2NOT, 3NOR, and 2IMP. They commonly give “1” output (yellow) only for “00” inputs. b) The circuit layout of the ZCM. The blue lines are the input data of the ZCM coming from the NOR-type gates, and the red line is the output of the ZCM going to the error correction module. c) Two possible data locations after the NOR-type gate operation, for a horizontal gate operation (left) and a vertical one (right). d) Experimental success rates of the 3NOR gate operation as a function of the V_{PGM} (y), when V_{COND} ($=x$) is 0.59 V, for various input cases. For each data point, the average success rate and their error was obtained from 9 cells and 30 trials per cell. The section in green is the desired operating condition for the ZCM where the error cases are controlled.

truth table of the NOR-type gates. In the 2NOT and 2IMP gates, the output (O) is overwritten on B. It shows that all gates generate "1" output when the inputs are "00." After limiting the error case to "00" inputs, if one reads all "0" values from inputs and output after the NOR-type gate execution, one can conclude that the inputs were "00" and the output was an erroneous "0." In other cases, there is at least one "1" value among the inputs and output, and there is no chance of error. Therefore, by counting the number of "0" from all inputs and output, one can easily conclude whether the gate operation was successful or not.

Counting whether the number of "0" is zero or not is simply possible using a NOR gate. Figure 2b shows the zeros checking module (ZCM) composed of the NOR gate, which can be easily built by a transistor circuit. If all inputs are "0," which is the only error case of the NOR-type gates, the ZCM will return a "TRUE" output. Then, the output signal can trigger the error correction module at the controller that forces an update of the output value to "1" by a set switching of the output cell.

In the stateful logic, gate operation is possible in two directions in the array, horizontally and vertically. The ZCM, located at the end terminal of the vertical line to sense the signal, can be applied to both cases. Figure 2c shows data location and data delivery pathway for error correction of the horizontal gate operation (left panel) and the vertical gate operation (right panel). Here, the data of two inputs (A, and B) and output (O) are delivered from the array to the ZCM by applying the reading voltage (V_R). For the 2NOT and 2IMP gates, two data (A, and O) are read from the array, and the remaining input terminal (B) is assigned to "0." For horizontal operation, three logic data (two inputs and one output) can be delivered to the three inputs of the ZCM independently along individual vertical lines. For vertical operation, all the data are read via the shared vertical line together. In this case, the sensing current corresponds to the OR gate operation of the data. Therefore, the sensing current is "0" if all data are "0." Then, the zero counting is applicable by assigning other inputs of ZCM to "0."

Due to the stochastic set voltage variation of the memristor, the success rate shows cell-to-cell variation. The proposed error correction method is applicable regardless of the cell-to-cell variation. Figure 2d shows the experimentally obtained success rates at the given voltage conditions. For each data point, the success rate was measured from 9 cells, 30 times each. At $x = 0.59$ V and y less than 0.5 V (green section), the success rates of the "010 or 100" and the "110" states remaining in their initial states were 100% for all cells, although that of the "000" state was less than 25% and fluctuating. As such, the error case can be limited to Error Type I even the presence of cell-to-cell variation in the device, which makes the error correction is possible.

4. Error Correction of Other Gates

The ZCM is not applicable for other stateful logic gates in addition to the NOR-type (2NOT, 2IMP, and 3NOR) gates, because the conditional switching is triggered at "10 or 01" and/or "11" input conditions. **Figure** 3a shows the theoretical (lines) and experimental (symbols) error rates of 3NAND operation, which is one of the examples of the non-NOR-type gates. Here, x was fixed at 0.59 V, and y was controlled. It shows that the success rate of the "000" and "110" states remaining "001" and "110" states, respectively, can be 100% at the optimized voltage condition of $y = 1$ V at $x = 0.59$ V, whereas the success rate of the "010 (or 100)" states being "011 (or 101)" states is below 20%. Then, the error case is "010 (or 100)," which gives two zeros, and thus, cannot be distinguished with the successful operation of the "001" states. If y increases, the error rate of the "010 (or 100)" states will decrease significantly, but the error rate of the "110" state will increase. It may reduce the total error rate but increase the number of error cases, which makes the error correction more tricky. Also, applying a higher voltage to the output may result in a "set-stuck" problem.[45]

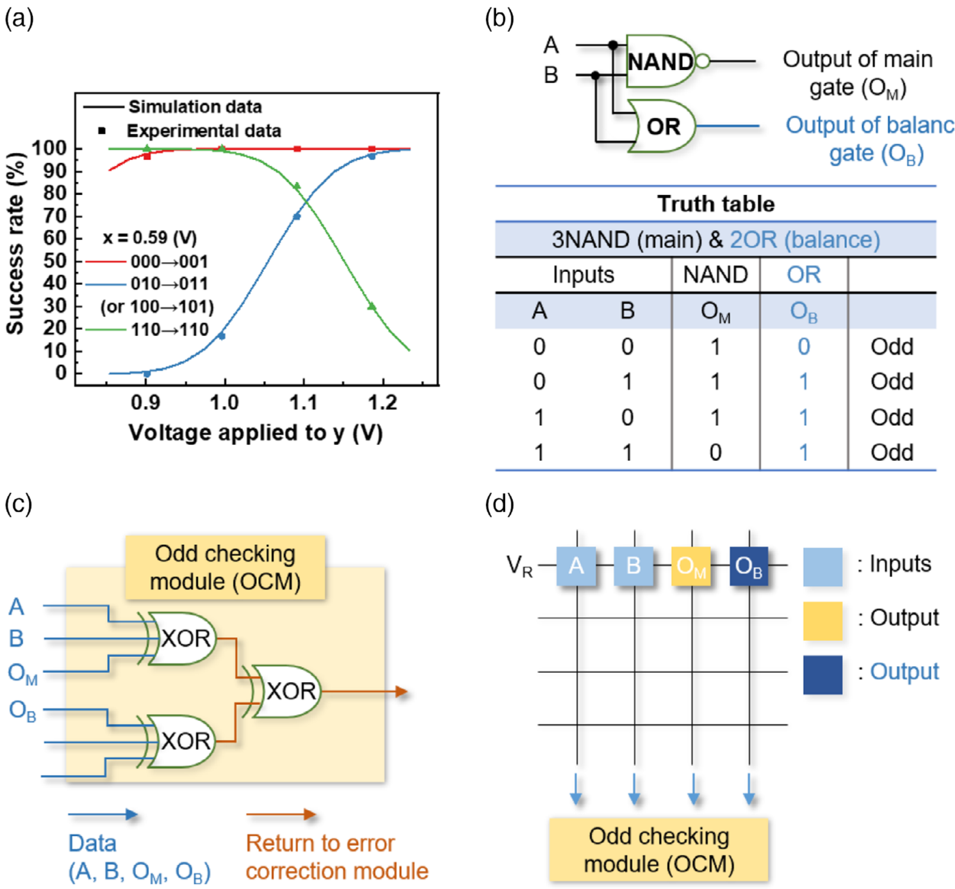
Error correction of the non-NOR-type gates is more crucial because their error rate is typically higher than that of the NOR-type gates. To deal with errors of the non-NOR-type gates, we propose a second type of error checking module, an odd checking module (OCM), that counts the number of "1" and checks whether it is odd or not.

To apply the OCM, the correct gate operation should always result in odd numbers of "1" values among the inputs and output, which is not possible using a single gate operation. Therefore, we introduce a "balance gate" that makes all of the inputs and outputs of the main gate and the balance gate an odd number. The balance gate should be free of error, or one will not be able to distinguish the origin of error using the OCM. Accordingly, it is chosen from one of the NOR-type gates that can be corrected by the ZCM. Figure 3b shows one of the pairs of the main gate and balance gate, NAND gate and OR gate, with their truth tables as an example. It shows that the number of "1"s from four data (two inputs and two outputs) are either one or three, that is, odd numbers. As such, the OCM can be applicable after executing a pair of two gates and reading their inputs and outputs together.

Figure 3c shows the circuit diagram of the OCM. The OCM is composed of multiple layers of XOR gates, which permits checking whether the number of "1"s in the data is odd or not. Figure 3d shows the data locations after the paired gate operation, where A and B are two inputs and O_M and O_B are two outputs of the main gate and balance gate, respectively. Unlike the ZCM, the OCM can only handle the data along the horizontal direction operation.

Figure 4a shows all pairs of the main gate (except the NOR-type gates) and the balance gate. The number of required balance gate steps indicates the number of steps for the balance gate operation by cascading the NOR-type gates. For example, the 3OR gate is a balance gate for the 3NAND gate operation that can be achieved by cascading 3NOR and 2NOT gates ($O_B{}' \leftarrow$ A 3NOR B, $O_B \leftarrow$ 2NOT O_B). (A truth table showing the combination of the main and balance gates is shown in Figure S1, Supporting Information. Also, a timing diagram of all gate operation with error correction steps is shown in Figure S2, Supporting Information).

Figure 4b shows the truth table of the 5SUM gate, which is the most efficient and crucial gate in Boolean computing. The 5SUM gate requires four inputs (A, B, C_{in}, and C_{out}) to obtain the sum (S) output. Interestingly, the 5SUM gate itself can result in a complete even number of "1"s, when selecting three inputs (A, B, and C_{in}) and output (S) except one input (C_{out}). Therefore, its balance gate can be a TRUE value that can be directly selected from a virtual "1" input without the additional balance gate operation.



Truth table

3NAND (main) & 2OR (balance)

Inputs		NAND	OR	
A	B	O_M	O_B	
0	0	1	0	Odd
0	1	1	1	Odd
1	0	1	1	Odd
1	1	0	1	Odd

Figure 3. The OCM for non-NOR-type gate correction. a) The success rate of the 3NAND gate operation as a function of V_{PGM} (y), when V_{COND} ($=x$) is 0.59 V, for various input cases. The dots are experimental data, and the lines are simulation data based on the V_{SET} distribution. b) A schematic diagram showing a pair of the main NAND gate and balance OR gate. (top panel). The truth table of the main NAND gate and balance OR gate (bottom panel). The sum of the two inputs (A, B) and two outputs (O_M, O_B) are odd numbers. c) The circuit layout of the OCM. The blue lines are the input data of the OCM coming from the crossbar array when executing non-NOR-type gates. The red line is the output of the OCM going to the error correction module. d) The data location after the non-NOR-type gate operation using a horizontal gate operation. Unlike the NOR-type gates, its vertical operation cannot be corrected by the OCM.

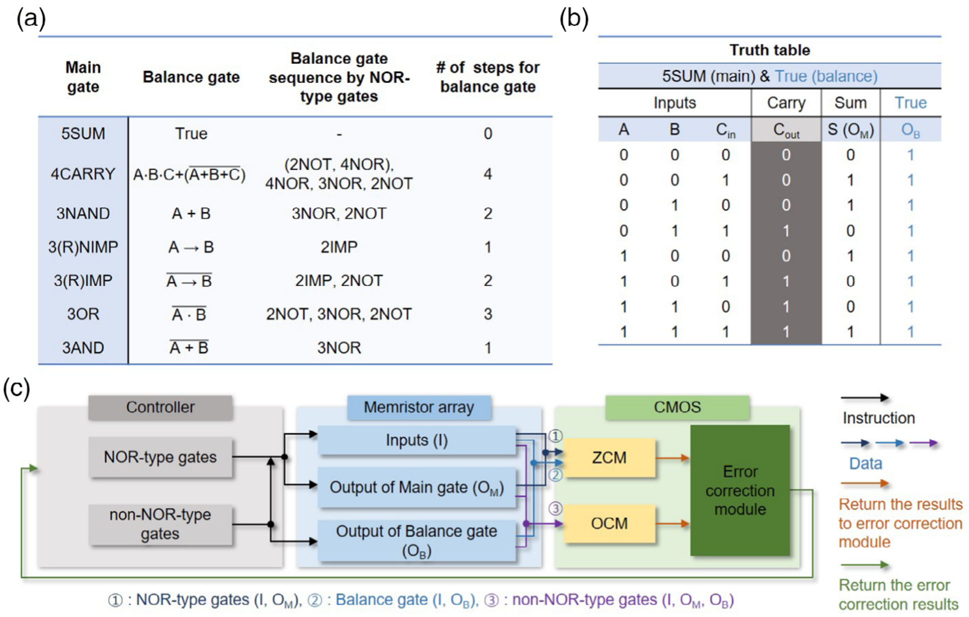
Figure 4c shows a system flow chart utilizing the ZCM and OCM combination to correct all logic gate errors. In the chart, the controller executes the designed gate operation on the memristor array. It can be either the main gate operation only for NOR-type gates or the main gate and balance gate together for non-NOR-type gates. After the gate execution, the controller sends the logic dataset to the ZCM for the NOR-type gates or the ZCM and OCM for the non-NOR-type gates. Then, the error correction module determines whether the gate execution was successful or not based on the delivered dataset. Here, the error correction module on the complementary metal-oxide semiconductor (CMOS) periphery can alternately be located in the controller CMOS.

5. Discussion on the Efficiency of the Error Correction Process

The primary purpose of the error correction is to improve the reliability of the gate operation. Using the error correction strategy, errors can be acceptable, and thus, device functionality can be ensured. In exchange, however, there seems to be an inevitable loss in computing efficiency due to the additional reading, verifying, and optional correcting steps for error correction. Assuming each additional step requires one clocking time, it costs at least three additional clocking times per gate operation, which is considerable.

However, optimizing the use of the error correction method can minimize the loss of computing efficiency by allowing a complicated gate operation such as a 5SUM gate. Furthermore, it can even improve the computing efficiency for some specific calculations, such as a full adder. In this study, we show how the error correction module enhances the full adder operation.

The first step of the full adder operation is to calculate the carry-out value from three inputs (A, B, and C_{in}). **Figure 5**a shows an equivalent circuit configuration of the optimized carry-out operation. The use of the 4CARRY gate here is impractical because its correction requires non-NOR-type gates, as shown in Figure 4a. Instead, the carry-out value can be obtained by cascading three NOR gates and one NOT gate. Figure 5b shows the data location for the carry-out operation where three inputs (A_1, B_1, and C_1) are



(a)

Main gate	Balance gate	Balance gate sequence by NOR-type gates	# of steps for balance gate
5SUM	True	-	0
4CARRY	$A\cdot B\cdot C+(\overline{A+B+C})$	(2NOT, 4NOR), 4NOR, 3NOR, 2NOT	4
3NAND	A + B	3NOR, 2NOT	2
3(R)NIMP	$A \rightarrow B$	2IMP	1
3(R)IMP	$\overline{A \rightarrow B}$	2IMP, 2NOT	2
3OR	$\overline{A \cdot B}$	2NOT, 3NOR, 2NOT	3
3AND	$\overline{A + B}$	3NOR	1

(b)

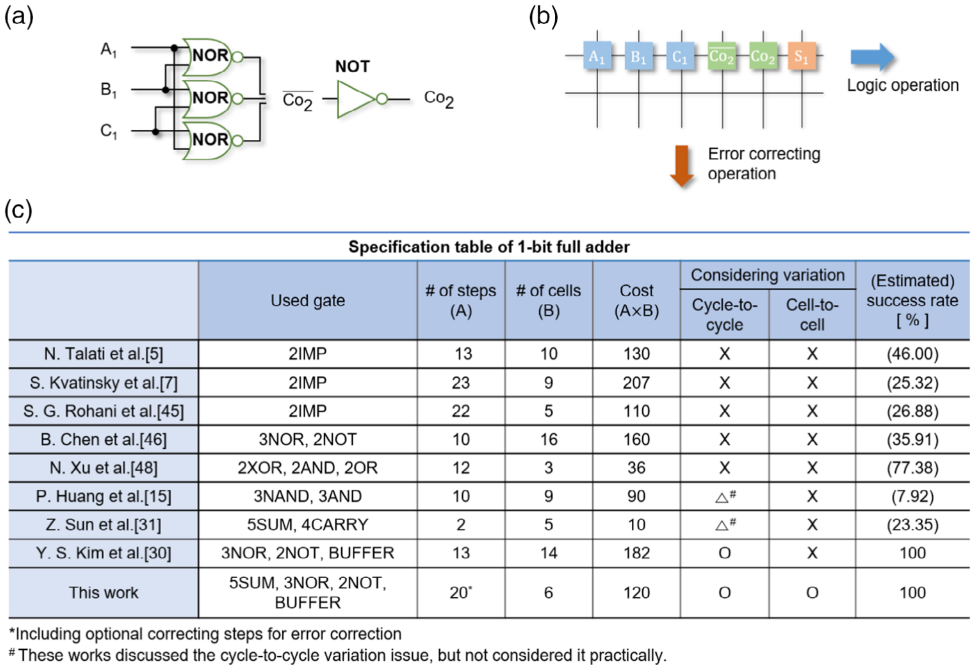
Truth table

5SUM (main) & True (balance)

Inputs			Carry	Sum	True
A	B	C_{in}	C_{out}	S (O_M)	O_B
0	0	0	0	0	1
0	0	1	0	1	1
0	1	0	0	1	1
0	1	1	1	0	1
1	0	0	0	1	1
1	0	1	1	0	1
1	1	0	1	0	1
1	1	1	1	1	1

(c)
Controller
NOR-type gates
non-NOR-type gates
Memristor array
Inputs (I)
Output of Main gate (O_M)
Output of Balance gate (O_B)
①
②
③
CMOS
ZCM
OCM
Error correction module
Instruction
Data
Return the results to error correction module
Return the error correction results
① : NOR-type gates (I, O_M), ② : Balance gate (I, O_B), ③ : non-NOR-type gates (I, O_M, O_B)

Figure 4. A combined ZCM and OCM system for universal error correction. a) A table summarizing the combination of the non-NOR-type main gates and the balance gate, where the balance gate is conducted by cascading the NOR-type gates. The required balance gate step shows the number of NOR-type gates needed for cascading the balance gate. b) The truth table of the main 5SUM gate and its balance gate. The odd checking of the 5SUM gate can be simplified by excluding the C_{OUT} value. c) The data flow diagram of the universal error correction system utilizing the OCM and ZCM.



Specification table of 1-bit full adder

	Used gate	# of steps (A)	# of cells (B)	Cost (A×B)	Considering variation		(Estimated) success rate [%]
					Cycle-to-cycle	Cell-to-cell	
N. Talati et al.[5]	2IMP	13	10	130	X	X	(46.00)
S. Kvatinsky et al.[7]	2IMP	23	9	207	X	X	(25.32)
S. G. Rohani et al.[45]	2IMP	22	5	110	X	X	(26.88)
B. Chen et al.[46]	3NOR, 2NOT	10	16	160	X	X	(35.91)
N. Xu et al.[48]	2XOR, 2AND, 2OR	12	3	36	X	X	(77.38)
P. Huang et al.[15]	3NAND, 3AND	10	9	90	△#	X	(7.92)
Z. Sun et al.[31]	5SUM, 4CARRY	2	5	10	△#	X	(23.35)
Y. S. Kim et al.[30]	3NOR, 2NOT, BUFFER	13	14	182	O	X	100
This work	5SUM, 3NOR, 2NOT, BUFFER	20*	6	120	O	O	100

*Including optional correcting steps for error correction

These works discussed the cycle-to-cycle variation issue, but not considered it practically.

Figure 5. Evaluation of efficiency for a one-bit full adder operation. a) A schematic diagram showing a 4CARRY gate unit consisting of three 3NOR gates and one 2NOT gate. b) The data delivering direction during the one-bit full adder operation. The logic operation is executed horizontally and the data transfer for the error correction is done vertically. c) Specification comparison table showing various stateful logic techiques and their efficiency for a one-bit full adder operation.

located side by side along the horizontal direction, and the interim and final inputs ($\overline{Co_2}$ and Co_2) are recorded in the fourth and fifth columns of the line by the horizontal gate operation. During the operation, the error correction can be conducted along the vertical direction per the gate operation. Therefore, the four gate operations require 16 timeunits in total.

After completing the carry-out operation, the sum operation is possible via the 5SUM gate. In Figure 5b, the sum value is recorded in the sixth column. The sum operation requires only four timeunits, one for the 5SUM gate and three for the error correction. Overall, the full adder operation with the error correction requires 20 timeunits, and during the operation, it involves six cells.

Figure 5c shows the comparison of the efficiency and practical feasibility of various stateful logic technologies for executing a one-bit full adder.[5,7,15,31,32,46–48] It shows the gate used, the number of cells involved, and the number of steps required to execute the one-bit full adder in each study. Here, to compare the computing efficiency, we introduce a total efficiency cost value, which is a multiplication of the number of cells (spatial cost) and the number of steps (temporal cost). The cost value can be as low as 10 when 4CARRY and 5SUM gates are used. However, the estimated success rate based on our methodology reveals they will suffer from errors in massive device applications. (The estimated success rate was calculated assuming the variation in the HfO_2 device used in this study. For example, when operating a one-bit full adder with 13 steps of 2IMP gate operation,[5] the success rate of a one-bit full adder was the 13th power of the 2IMP gate success rate). The cost value in this work is 120, which is not the best but comparable with other methods or even higher than some of them. However, it is the only practically feasible method that can deal with the cycle-to-cycle variation and cell-to-cell variation.

6. Conclusions

We have described error correction methods for stateful logic devices, based on counting the number of “0” or “1” values at inputs and outputs. By adopting two types of error correction modules after carefully selecting the operation voltages, all gate operation errors can be corrected. These error correction modules can be used to the stateful logic device without significant loss of calculation efficiency. The error correction modules require additional transistor circuits on the periphery of the crossbar device, which may reduce spatial efficiency and increase the cost of the circuit. Nevertheless, the error correction system is worth implementing in a device, since without it, practical stateful logic technology cannot be realized. This study suggests that the in-memory computing device would require the help of near-memory computing at any cost. Then, the next can be about maximizing the functionality of the near-memory computing circuit or optimizing the functions of in-memory and near-memory computing units. For example, considering the error correction modules are composed of NOR and XOR gates and free of errors, those gate operations can be directly executed in the periphery error correction circuit instead of in-memory, which may increase the computation efficiency further.

7. Experimental Section

A $Pt/HfO_2/Ta$ crossbar array device was fabricated using the following procedure. First, adhesive 15 nm Ti followed by 20 nm Pt were deposited by e-beam evaporation and patterned using a lift-off process. Then, a 5 nm HfO_2 layer was deposited using a thermal ALD process at 250 °C using tetrakis (ethylmethylamido)hafnium (TEMAHf) and O_3 as a precursor and an oxidant. Finally, a 50 nm Ta was sputtered and patterned using a lift-off process.

Supporting Information

Supporting Information is available from the Wiley Online Library or from the author.

Acknowledgements

This research was supported by the MOTIE (Ministry of Trade, Industry & Energy) (Grant numbers 20003655 and 20003789) and KSRC (Korea Semiconductor Research Consortium) support program for the development of the future semiconductor device; and 2020 UP Research Project of KAIST.

Conflict of Interest

The authors declare no conflict of interest.

Keywords

error corrections, memristors, near-memory computing, resistive switching memories, stateful logics

Received: April 21, 2020
Revised: May 23, 2020
Published online: June 24, 2020

[1] J. Borghetti, G. S. Snider, P. J. Kuekes, J. J. Yang, D. R. Stewart, R. S. Williams, *Nature* **2010**, *464*, 873.
[2] E. Linn, R. Rosezin, S. Tappertzhofen, U. Bottger, R. Waser, *Nanotechnology* **2012**, *23*, 305205.
[3] M. Maestro-Izquierdo, J. Martin-Martinez, A. C. Yepes, M. Escudero, R. Rodriguez, M. Nafria, X. Aymerich, A. Rubio, *IEEE Trans. Electron Devices* **2018**, *65*, 404.
[4] F. S. Marranghello, V. Callegaro, M. G. A. Martins, A. I. Reis, R. P. Ribas, *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2015**, *5*, 267.
[5] N. Talati, S. Gupta, P. Mane, S. Kvatinsky, *IEEE Trans. Nanotechnol.* **2016**, *15*, 635.
[6] S. Kvatinsky, D. Belousov, S. Liman, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, U. C. Weiser, *IEEE Trans. Circuits Syst. II: Express Briefs* **2014**, *61*, 895.
[7] S. Kvatinsky, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, U. C. Weiser, *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2014**, *22*, 2054.
[8] D. B. Strukov, G. S. Snider, D. R. Stewart, R. S. Williams, *Nature* **2008**, *453*, 80.
[9] K. M. Kim, N. Xu, X. Shao, K. J. Yoon, H. J. Kim, R. S. Williams, C. S. Hwang, *Phys. Status Solidi (RRL) – Rapid Res. Lett.* **2019**, *13*, 1800629.
[10] M. A. Zidan, J. P. Strachan, W. D. Lu, *Nat. Electron.* **2018**, *1*, 22.
[11] K. M. Kim, R. S. Williams, *IEEE Trans. Circuits Syst. I: Regular Pap.* **2019**, *66*, 4348.
[12] G. C. Adam, B. D. Hoskins, M. Prezioso, D. B. Strukov, *Nano Res.* **2016**, *9*, 3914.
[13] A. Siemon, S. Menzel, D. Bhattacharjee, R. Waser, A. Chattopadhyay, E. Linn, *Eur. Phys. J. Spec. Top.* **2019**, *228*, 2269.

[14] W.-H. Chen, C. Dou, K.-X. Li, W.-Y. Lin, P.-Y. Li, J.-H. Huang, J.-H. Wang, W.-C. Wei, C.-X. Xue, Y.-C. Chiu, Y.-C. King, C.-J. Lin, R.-S. Liu, C.-C. Hsieh, K.-T. Tang, J. J. Yang, M.-S. Ho, M.-F. Chang, *Nat. Electron.* **2019**, *2*, 420.

[15] P. Huang, J. Kang, Y. Zhao, S. Chen, R. Han, Z. Zhou, Z. Chen, W. Ma, M. Li, L. Liu, X. Liu, *Adv. Mater.* **2016**, *28*, 9758.

[16] D. S. Jeong, K. M. Kim, S. Kim, B. J. Choi, C. S. Hwang, *Adv. Electron. Mater.* **2016**, *2*, 1600090.

[17] G. Papandroulidakis, I. Vourkas, A. Abusleme, G. C. Sirakoulis, A. Rubio, *IEEE Trans. Nanotechnol.* **2017**, *16*, 491.

[18] W. Shen, P. Huang, M. Fan, R. Han, Z. Zhou, B. Gao, H. Wu, H. Qian, L. Liu, X. Liu, X. Zhang, J. Kang, *IEEE Electron Device Lett.* **2019**, *40*, 1538.

[19] A. Siemon, R. Drabinski, M. J. Schultis, X. Hu, E. Linn, A. Heittmann, R. Waser, D. Querlioz, S. Menzel, J. S. Friedman, *Sci. Rep.* **2019**, *9*, 14618.

[20] A. Siemon, S. Menzel, R. Waser, E. Linn, *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2015**, *5*, 64.

[21] N. Xu, T. G. Park, H. J. Kim, X. Shao, K. J. Yoon, T. H. Park, L. Fang, K. M. Kim, C. S. Hwang, *Adv. Intell. Syst.* **2019**, *2*, 1900082.

[22] J. J. Yang, D. B. Strukov, D. R. Stewart, *Nat. Nanotechnol.* **2013**, *8*, 13.

[23] X. Zhu, X. Yang, C. Wu, N. Xiao, J. Wu, X. Yi, *IEEE Trans. Circuits Syst. II: Express Briefs* **2013**, *60*, 682.

[24] N. Xu, K. J. Yoon, K. M. Kim, L. Fang, C. S. Hwang, *Adv. Electron. Mater.* **2018**, *4*, 1800189.

[25] K. M. Kim, G. H. Kim, S. J. Song, J. Y. Seok, M. H. Lee, J. H. Yoon, C. S. Hwang, *Nanotechnology* **2010**, *21*, 305203.

[26] J. van den Hurk, V. Havel, E. Linn, R. Waser, I. Valov, *Sci. Rep.* **2013**, *3*, 2856.

[27] D. Ielmini, H. S. P. Wong, *Nat. Electron.* **2018**, *1*, 333.

[28] S. Ambrogio, S. Balatti, A. Cubeta, A. Calderoni, N. Ramaswamy, D. Ielmini, *IEEE Trans. Electron Devices* **2014**, *61*, 2912.

[29] K. M. Kim, D. S. Jeong, C. S. Hwang, *Nanotechnology* **2011**, *22*, 254002.

[30] Z. Wei, K. Eriguchi, S. Muraoka, K. Katayama, R. Yasuhara, K. Kawai, Y. Ikeda, M. Yoshimura, Y. Hayakawa, K. Shimakawa, T. Mikawa, S. Yoneda, presented at *IEEE International Electron Devices Meeting*, USA **2015**, pp. 7.7.1–7.7.4.

[31] Y. S. Kim, M. W. Son, H. Song, J. Park, J. An, J. B. Jeon, G. Y. Kim, S. Son, K. M. Kim, *Adv. Intell. Syst.* **2020**, *2*, 1900156.

[32] Z. Sun, E. Ambrosi, A. Bricalli, D. Ielmini, *Adv. Mater.* **2018**, *30*, 1802554.

[33] J. Maiz, S. Hareland, K. Zhang, P. Armstrong, presented at *IEEE International Electron Devices Meeting*, USA **2003**, pp. 21.4.1–21.4.4.

[34] A. K. Nieuwland, S. Jasarevic, G. Jerin, presented at *12th IEEE Int. On-Line Testing Symp.*, Italia **2006**, pp. 6–11.

[35] S. Mitra, M. Zhang, S. Waqas, N. Seifert, B. Gill, K. S. Kim, presented at *2006 IEEE Int. Test Conf.*, USA **2006**, pp. 1–9.

[36] R. Baumann, presented at *Digest. Int. Electron Devices Meeting*, USA **2002**, pp. 329–332.

[37] W. Liu, J. Rho, W. Sung, presented at *IEEE Workshop on Signal Processing Systems Design and Implementation*, Canada **2006**, pp. 303–308.

[38] G. Dong, N. Xie, T. Zhang, *IEEE Trans. Circuits Syst. I: Regular Pap.* **2011**, *58*, 429.

[39] G. H. Kim, H. Ju, M. K. Yang, D. K. Lee, J. W. Choi, J. H. Jang, S. G. Lee, I. S. Cha, B. K. Park, J. H. Han, T. M. Chung, K. M. Kim, C. S. Hwang, Y. K. Lee, *Small* **2017**, *13*, 1701781.

[40] H. Jiang, L. Han, P. Lin, Z. Wang, M. H. Jang, Q. Wu, M. Barnell, J. J. Yang, H. L. Xin, Q. Xia, *Sci. Rep.* **2016**, *6*, 28525.

[41] S. Yu, H.-Y. Chen, B. Gao, J. Kang, H. S. P. Wong, *ACS Nano* **2013**, *7*, 2320.

[42] J. J. Yang, M. X. Zhang, J. P. Strachan, F. Miao, M. D. Pickett, R. D. Kelley, G. Medeiros-Ribeiro, R. S. Williams, *Appl. Phys. Lett.* **2010**, *97*, 232102.

[43] K. M. Kim, J. J. Yang, E. Merced, C. Graves, S. Lam, N. Davila, M. Hu, N. Ge, Z. Li, R. S. Williams, C. S. Hwang, *Adv. Electron. Mater.* **2015**, *1*, 1500095.

[44] H. Jiang, D. Belkin, S. E. Savel'ev, S. Lin, Z. Wang, Y. Li, S. Joshi, R. Midya, C. Li, M. Rao, M. Barnell, Q. Wu, J. J. Yang, Q. Xia, *Nat. Commun.* **2017**, *8*, 882.

[45] K. M. Kim, J. J. Yang, J. P. Strachan, E. M. Grafals, N. Ge, N. D. Melendez, Z. Li, R. S. Williams, *Sci. Rep.* **2016**, *6*, 20085.

[46] S. G. Rohani, N. TaheriNejad, presented at *IEEE 30th Canadian Conf. on Electrical and Computer Engineering*, Canada **2017**, pp. 1–4.

[47] B. Chen, F. Cai, J. Zhou, W. Ma, P. Sheridan, W. D. Lu, presented at *IEEE Int. Electron Devices Meeting*, USA **2015**, pp. 17.5.1–17.5.4.

[48] N. Xu, L. Fang, K. M. Kim, C. S. Hwang, *Phys. Status Solidi (RRL) – Rapid Res. Lett.* **2019**, *13*, 1900033.