This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSII.2015.2435354, IEEE Transactions on Circuits and Systems II: Express Briefs

A Novel Design for Memristor-Based OR Gate

Yang Zhang, Yi Shen, Xiaoping Wang, and Yanwen Guo

***Abstract*—This brief proposes a logic gate that performs a stateful OR logic operation on memristor memory. The presented logic structure execute an OR operation in the nanocrossbar architecture concurrently in a single step, which enables fast logic operation and reduces the number of the required memristors. The proposed circuit completes the *in situ* logic operation on memristor memory, which alleviates the burden of the processor significantly. Through analysis and simulation, the feasibility of the OR operation is demonstrated and the parameter optimization is analyzed.**

***Index Terms*—Memristor, material implication, OR gate, stateful logic, nanocrossbar memory.**

I. Introduction

As it has become increasingly difficult to overcome various physical limitations of the traditional CMOS technology [1], alternative methods are desired for manufacturing higher performance memory and logic applications. One of the possible candidates is memristor. Memristor was first theoretically postulated by L. Chua in 1971 [2] and later Williams's team presented a resistance variable device as a memristor at HP Labs in 2008 [3]. As a new nanoscale device, memristor provides a lot of advantageous features, such as non-volatility, high-density, low-power, and good-scalability.

In the traditional Von Neumann computer architecture, the data is stored in memory and processed in the processor. Moreover, it must be read from memory to the processor first and then written back to the memory afterwards. This memory accessing process becomes a bottleneck of the computer system. One of the possible solutions is to use memristor. A memristor-based material-implication (IMP) logic gate is proposed in [4] as stateful logic, which can process data on memory *in situ* without reading it out or writing back explicitly, thus reducing the bandwidth needed for accessing data. Afterwards, memristors have drew the researchers' interests in logic circuits [5]-[10], memristive logic arrays [8], [11]-[14], and other structures like PLA, PMLA TLA [8], [15], memristor-CMOS hybrids [8], [16], [17], and FPGA-like memristor systems [11], [18], [19]. However, the IMP-based AND, OR, and NOT logic gates need multiple IMP operations which take much time and cause the accumulation of error [15]. A memristor ratioed logic (MRL) gate is presented in [17], which needs a CMOS inverter and cannot store logic state. In [12], a memristor-based AND logic circuit structure is proposed which provides the memory with the function of *in situ* logic operation and thus can potentially reduce the amount of memory accessing actions. In [9], memristor-aided logic (MAGIC) is presented, which is also a Non-Von Neumann architecture. In each MAGIC logic gate, memristors serve as an input with previously stored data, and an additional memristor serves as an output, which can store the input and output data respectively and needs only one applied voltage, however, it needs initialization for every operation.

In this brief, we propose a novel design for OR logic circuit structure that completes the *in situ* logic operation on memristor memory. In addition, the feasibility of the operation and the parameter optimization are analyzed. Furthermore, we simulate the proposed memristor-based logic circuits and demonstrate the feasibility to proceed the OR operation.

II. Motivation

A. Performing Logic Operation in Situ

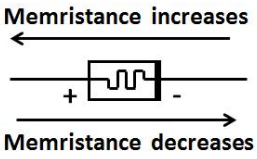


Fig. 1. Memristor symbol. The negative polarity of the memristor is represented by a thick black line. When current flows from negative polarity to positive polarity (the upper arrow), the memristance increases. When current flows from positive polarity to negative polarity (the lower arrow), the memristance decreases.

The memristor released by Williams's team in 2008 consists of two regions: the doped region and the undoped region [3]. The width of the doped region (w) changes based on the direction of the bias voltage across the memristor [3]. Meanwhile, the width of undoped region ($D - w$) changes accordingly. R_{ON} denotes the internal low resistance when the memristor is completely doped ($w = D$) and R_{OFF} denotes the internal high resistance when the memristor is completely undoped ($w = 0$). Therefore, the memristance is changed according to the direction of the bias voltage, as shown in Fig. 1. Logic operations performed *in situ* can be realized by memristors [4], [12], which may change the computer architecture in the future.

Performing logic operations in traditional computer architecture, as shown in Fig. 2(a), memory cell A[i] and A[j] are

Y. Zhang is with the School of Automation, Huazhong University of Science and Technology, Wuhan, Hubei 430074, China.

Y. Shen (the corresponding author) is with the School of Automation and the Key Laboratory of Ministry of Education for Image Processing and Intelligent Control, Huazhong University of Science and Technology, Wuhan, Hubei 430074, China (e-mail: yishen64@163.com).

X. P. Wang and Y. W. Guo are with the School of Automation, Huazhong University of Science and Technology, Wuhan, Hubei 430074, China.

The work is supported by the State Key Program of the National Natural Science Foundation of China (Grant No. 61134012), the National Natural Science Foundation of China (Grant Nos. 61374150 and 11271146), and the Doctoral Fund of Ministry of Education of China (Grant No.20130142130012).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSII.2015.2435354, IEEE Transactions on Circuits and Systems II: Express Briefs

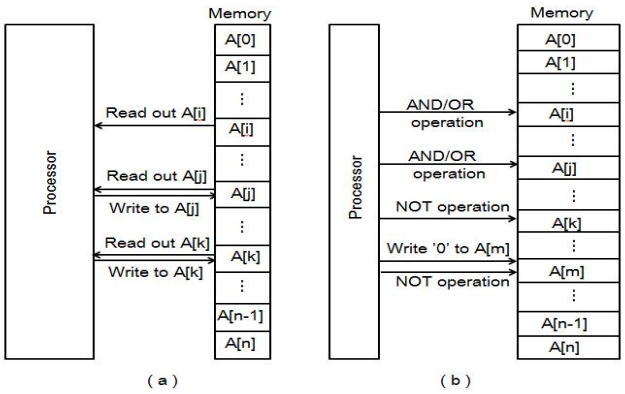
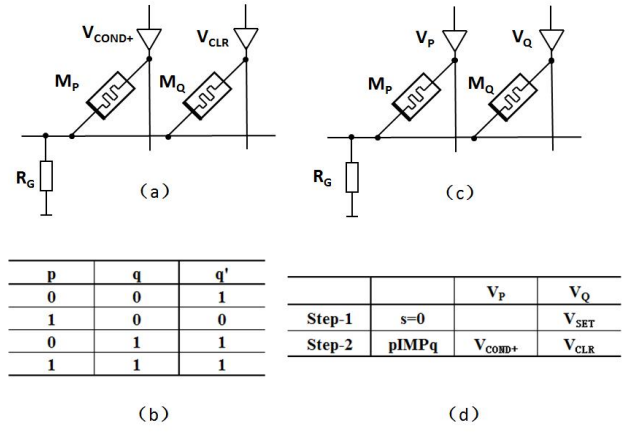


Fig. 2. (a) Action sequence of moving data on traditional memory. (b) Action sequence of *in situ* logic operation.

selected to proceed AND/OR operation and A[k] is selected to proceed NOT operation. The selected memory cells need to be read from memory into the processor first and then written back to memory.

Performing logic operation *in situ* is a significant difference from the traditional mechanism, as shown in Fig. 2(b). Once the memory cells to proceed logic operation is selected, the processor just need to apply different voltages to the cells, thus alleviating the burden of the processor significantly. AND, OR, or NOT logic operation can be proceeded in any two cells and the output logic value can be stored in one cell directly without explicit reading and writing. The size of the processor is therefore smaller and the required memory bandwidth is lower than that of the traditional computer architecture.

B. Proposed Logic Gates in Situ



p	q	q'
0	0	1
1	0	0
0	1	1
1	1	1

		V_P	V_Q
Step-1	s=0		V_{SET}
Step-2	pIMPq	V_{COND+}	V_{CLR}

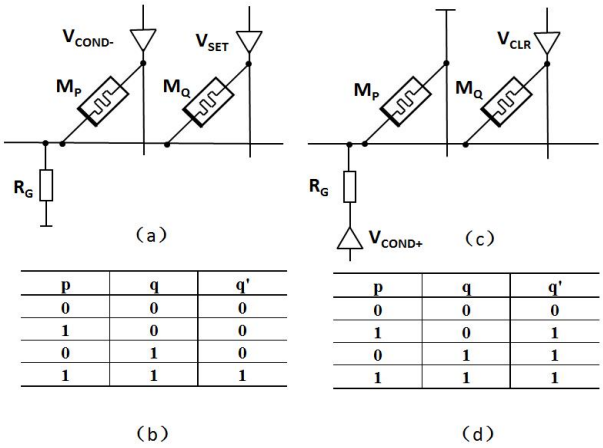
Fig. 3. Material implication and stateful NOT gate. (a) Material implication performed by two voltage pulses $V_{\rm COND+}$ and $V_{\rm CLR}$, which are applied to memristors P and Q, respectively. (b) Truth table for the material implication. (c) NOT operation performed by a two-step schedule. (d) Schedule to execute a NOT operation.

In 2010, the memristor-based IMP operation is first proposed by HP Laboratories [4]. Fig. 3(a) illustrates an IMP operation in stateful logic [4], where two memristors P and Q are connected by a common horizontal nanowire to the load resistor $\rm R_G$. The states of $M_{\rm P}$ and $M_{\rm Q}$ are represented by p and q, respectively. When the memristance is $R_{\rm OFF}$ ($R_{\rm ON}$), the state of the memristor x ($x = w/D$) is 0 (1) which corresponds to logic 0 (1). When the voltage of the memristor is larger than the positive threshold voltage $V_{\rm OPEN}$, the memristance decreases and the state of memristor can change to 1. When the voltage of the memristor is smaller than the negative threshold voltage $V_{\rm CLOSE}$ ($V_{\rm CLOSE} = -V_{\rm OPEN}$), the memristance increases and the state of memristor can change to 0. A memristor is set to logic 0 (1) by applying a negative (positive) voltage pulse $V_{\rm SET}$ ($V_{\rm CLR}$) through its corresponding voltage driver. Applying a positive condition voltage pulse $V_{\rm COND+}$ on $M_{\rm P}$ will make the logic of $M_{\rm Q}$ to 1 dependent on the state of $M_{\rm P}$, as shown in Fig. 3(b). When $M_{\rm P}$ is at 0 state, voltage $V_{\rm CLR}$ makes the voltage of $M_{\rm Q}$ larger than $V_{\rm OPEN}$, which decreases the memristance of $M_{\rm Q}$ and makes the state of $M_{\rm Q}$ to 1. When the state of $M_{\rm P}$ is 1, the effect of voltage $V_{\rm COND+}$ makes the voltage of $M_{\rm Q}$ smaller than $V_{\rm OPEN}$, which makes the state of $M_{\rm Q}$ unchanged.

An IMP-based NOT gate (shown in Fig. 3(c)) is executed through two sequential steps as shown in Fig. 3(d) [4]. The equivalent operation based on IMP can be described as

$$\text{NOT } q = q \text{ IMP } 0.$$

The input is logic value of p stored in $M_{\rm P}$, and the output is the logic value q' stored in $M_{\rm Q}$ after the operation. Initially, a voltage pulse $V_{\rm SET}$ is applied to $M_{\rm Q}$ to execute $q = 0$. The second step $q' = \overline{p} + q = \overline{p}$ is performed by applying a voltage pulse $V_{\rm COND+}$ to $M_{\rm P}$ concurrently with a voltage pulse $V_{\rm CLR}$ to $M_{\rm Q}$.



p	q	q'
0	0	0
1	0	0
0	1	0
1	1	1

p	q	q'
0	0	0
1	0	1
0	1	1
1	1	1

Fig. 4. Stateful AND and OR gates. (a) AND operation performed by a one-step schedule. (b) Truth table for AND operation. (c) OR operation performed by a one-step schedule. (d) Truth table for OR operation.

An AND logic operation is proposed in [12], as shown in Fig. 4(a). The AND operation is executed through only one step, and it is similar to IMP operation in circuit mechanism, which uses negative conditional voltage pulse $V_{\rm COND-}$ and negative voltage pulse $V_{\rm SET}$ to perform conditional set. The truth table of AND operation is shown in Fig. 4(b).

Similarly, we propose a novel design for OR logic circuit structure that completes the *in situ* logic operation as shown in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSII.2015.2435354, IEEE Transactions on Circuits and Systems II: Express Briefs

Fig. 4(c). The OR operation is also executed through only one step. In the proposed OR circuit, positive conditional voltage pulse $V_{\mathrm{COND}+}$ and positive voltage pulse V_{CLR} are applied to R_{G} and M_{Q}, respectively, and M_{P} is connected to GND. The truth table of OR operation is shown in Fig. 4(d). When M_{P} is at 1 state, voltage V_{CLR} makes the voltage of M_{Q} larger than V_{OPEN}, which decreases the memristance of M_{Q} and makes the state of M_{Q} to 1. When the state of M_{P} is 0, the effect of voltage $V_{\mathrm{COND}+}$ makes the voltage of M_{Q} smaller than V_{OPEN}, which makes the state of M_{Q} unchanged.

The voltages in the IMP and AND logic operations in this brief are exactly opposited to those in [4], [12] for the reason that the polarities of the memristors connected with the voltages are opposited. Our design is consistent to the read/write circuits of the memristors in memory architecture, in which the voltages are connected to the positive polarities of the memristors. The memristive device in our design is only used as a binary switch for logic 1 and 0. Other memristive devices with voltage thresholds have the possibilities to be used in the logic operations.

III. Structure and Analysis

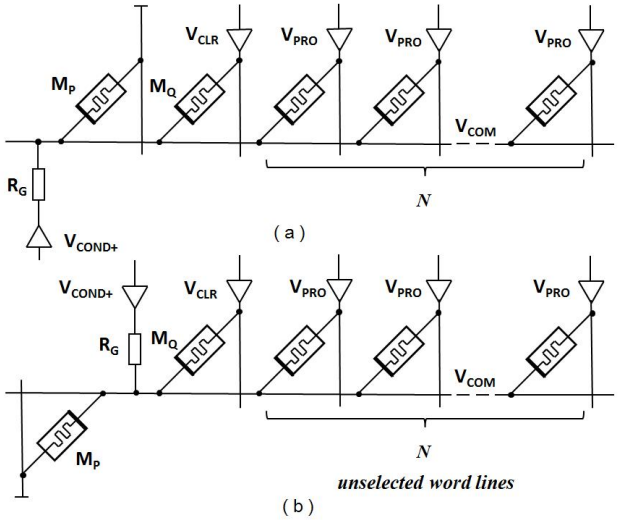


Fig. 5. (a) Simplified circuit of a single bit line of the OR operation. (b) Equivalent circuit of a single bit line of the OR operation.

Many proposed nanoscale architectures have focused on the crossbar architecture due to its fabrication simplicity and good noise tolerance [20]. We choose the crossbar architecture as the basic memory structure to operate stateful logic function. We can proceed AND, OR, or NOT logic operation in any two cells and store the output logic value in one cell directly. In addition, combined logic gates can be performed through multi-step operations in the same crossbar array and we only need to apply different voltages in each step, which shows the reconfigurability of the proposed design. For example, the NAND logic operation can be realized by choosing any two memristors to proceed the AND operation first, then the NOT operation. Similarly, the NOR gate can also be realized.

Nevertheless, researchers have found that one of the main challenges memristors facing at the architecture level is the sneak-path problem [21]. One possible solution to the sneak-path problem is to connect the unselected word lines to a protect voltage V_{PRO}, which makes the voltage level of each bit line independent to each other [12]. The level of V_{PRO} should be carefully chosen, otherwise, V_{PRO} itself will also influence the voltage level of each bit line.

An updated version of memristor SPICE model is presented in [22]. We choose Biolek's model for the reason that even the simple linear dopant drift model is valid to characterize the property of memristor and only threshold-based memristor models can perform the logic operations correctly. As input voltage conditions for all bit lines are identical, we only need to take the situation on one bit line into consideration. Fig. 5(a) shows the simplified circuit of a single bit line connected with N + 2 memristors and Fig. 5(b) shows the equivalent circuit. We choose arbitrary two memristors P and Q to participant in the OR operation and apply voltages GND and V_{CLR} to them, respectively. $V_{\mathrm{COND}+}$ is applied to R_{G} and V_{PRO} is applied to the other unselected memristors. Only the worst case is considered when all the unselected memristors are at 1 state, as V_{PRO} has the biggest impact in this case. According to Kirchhoffs law, we can get

$$\begin{aligned}(V_{\mathrm{COND}+}-V_{\mathrm{COM}})/R_{\mathrm{G}}+(V_{\mathrm{CLR}}-V_{\mathrm{COM}})/R_{\mathrm{Q}}\\ +N\times(V_{\mathrm{PRO}}-V_{\mathrm{COM}})/R_{\mathrm{ON}}=V_{\mathrm{COM}}/R_{\mathrm{P}}\end{aligned}\quad(1)$$

where V_{COM} is the voltage level of the bit line.

We set V_{PRO} to the same level of V_{COM} when the states of M_{P} and M_{Q} are 1 and 0, respectively, then the unselected memristors will not influence the OR operation process in such case. That is

$$\begin{cases}\frac{V_{\mathrm{COND}+}-V_{\mathrm{COM}}}{R_{\mathrm{G}}}+\frac{V_{\mathrm{CLR}}-V_{\mathrm{COM}}}{R_{\mathrm{OFF}}}=\frac{V_{\mathrm{COM}}}{R_{\mathrm{ON}}}\\ V_{\mathrm{PRO}}=V_{\mathrm{COM}}.\end{cases}\quad(2)$$

V_{PRO} can be derived as

$$V_{\mathrm{PRO}}=\frac{1}{1+\alpha+\beta}(\beta V_{\mathrm{COND}+}+V_{\mathrm{CLR}})\quad(3)$$

where $\alpha=R_{\mathrm{OFF}}/R_{\mathrm{ON}}$, and $\beta=R_{\mathrm{OFF}}/R_{\mathrm{G}}$.

According to the second line of the truth table in Fig. 4(d), when M_{P} is at 1 state, the effect of the two voltages V_{CLR} and $V_{\mathrm{COND}+}$ should change the logic of M_{P} to 1, which demands the voltage drop on M_{Q} higher than the threshold voltage V_{OPEN}

$$\begin{cases}\frac{V_{\mathrm{COND}+}-V_{\mathrm{COM}}}{R_{\mathrm{G}}}+\frac{V_{\mathrm{CLR}}-V_{\mathrm{COM}}}{R_{\mathrm{OFF}}}=\frac{V_{\mathrm{COM}}}{R_{\mathrm{ON}}}\\ V_{\mathrm{CLR}}-V_{\mathrm{COM}}>V_{\mathrm{OPEN}}.\end{cases}\quad(4)$$

During the process, the state of M_{P} should be kept unchanged, which demands the voltage drop on M_{P} lower than the threshold voltage V_{CLOSE} ($V_{\mathrm{CLOSE}}=-V_{\mathrm{OPEN}}$) before the V_{CLR} pulse is applied to M_{Q} when V_{CLR} is grounded

$$\begin{cases}\frac{V_{\mathrm{COND}+}-V_{\mathrm{COM}}}{R_{\mathrm{G}}}+\frac{0-V_{\mathrm{COM}}}{R_{\mathrm{OFF}}}=\frac{V_{\mathrm{COM}}}{R_{\mathrm{ON}}}\\ V_{\mathrm{COM}}<V_{\mathrm{OPEN}}\end{cases}\quad(5)$$

According to the first line of the truth table in Fig. 4(d), when M_{P} is at 0 state, the effect of the two voltages V_{CLR}

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSII.2015.2435354, IEEE Transactions on Circuits and Systems II: Express Briefs

and $V_{\mathrm{COND+}}$ should keep the state of M_{Q} unchanged, which demands the voltage drop on M_{Q} lower than V_{OPEN}

$$\begin{cases} \frac{V_{\mathrm{COND+}}-V_{\mathrm{COM}}}{R_{\mathrm{G}}} + \frac{V_{\mathrm{CLR}}-V_{\mathrm{COM}}}{R_{\mathrm{OFF}}} + N \times \frac{V_{\mathrm{PRO}}-V_{\mathrm{COM}}}{R_{\mathrm{ON}}} = \frac{V_{\mathrm{COM}}}{R_{\mathrm{OFF}}} \\ V_{\mathrm{CLR}} - V_{\mathrm{COM}} < V_{\mathrm{OPEN}} \end{cases} \tag{6}$$

During the process, the state of M_{P} should be also kept unchanged, which demands the voltage drop on M_{P} lower than the threshold voltage V_{CLOSE} ($V_{\mathrm{CLOSE}} = -V_{\mathrm{OPEN}}$) before the V_{CLR} pulse is applied to M_{Q} when V_{CLR} is grounded

$$\begin{cases} \frac{V_{\mathrm{COND+}}-V_{\mathrm{COM}}}{R_{\mathrm{G}}} + \frac{0-V_{\mathrm{COM}}}{R_{\mathrm{OFF}}} + N \times \frac{V_{\mathrm{PRO}}-V_{\mathrm{COM}}}{R_{\mathrm{ON}}} = \frac{V_{\mathrm{COM}}}{R_{\mathrm{OFF}}} \\ V_{\mathrm{COM}} < V_{\mathrm{OPEN}} \end{cases} \tag{7}$$

The values of V_{COM} in (4)–(7) are different.

From (4)–(7), we can get

$$V_{\mathrm{CLR}} > \frac{\beta}{\alpha+\beta} V_{\mathrm{COND+}} + \frac{1+\alpha+\beta}{\alpha+\beta} V_{\mathrm{OPEN}} \tag{8}$$

$$V_{\mathrm{COND+}} < \frac{1+\alpha+\beta}{\beta} V_{\mathrm{OPEN}} \tag{9}$$

$$\begin{aligned} V_{\mathrm{CLR}} < & \frac{\beta(1+\alpha+\beta+N\times\alpha)}{(\beta+1)^2+\alpha[N(\alpha+\beta)+\beta+1]} V_{\mathrm{COND+}} \\ & + \frac{(1+\alpha+\beta)(N\times\alpha+2+\beta)}{(\beta+1)^2+\alpha[N(\alpha+\beta)+\beta+1]} V_{\mathrm{OPEN}} \end{aligned} \tag{10}$$

$$\begin{aligned} V_{\mathrm{COND+}} + & \frac{N\times\alpha}{\beta(1+\alpha+\beta+N\times\alpha)} V_{\mathrm{CLR}} \\ & < \frac{(1+\alpha+\beta)(N\times\alpha+2+\beta)}{\beta(1+\alpha+\beta+N\times\alpha)} V_{\mathrm{OPEN}}. \end{aligned} \tag{11}$$

IV. Simulations

MATLAB has been used to simulate the valid solution space of parameter sets ($V_{\mathrm{COND+}}$, V_{CLR}) by using the following parameters: $V_{\mathrm{OPEN}} = 1V$; $\alpha = 1000$; $\beta = 2000$. Fig. 6 shows the valid solution space of parameter sets ($V_{\mathrm{COND+}}$, V_{CLR}), which satisfies (8)–(11) at different scales. As the circuit extends to larger size, for instance, N increases from 0 to 64, the area of the solution range shrinks continuously.

As increasing the value of V_{OPEN} in (8) and decreasing it in (9), (10), and (11) will narrow the solution space, there exists a maximum tolerance of the positive and negative errors ΔV_{OPEN}, which narrows the solution range to a point. Assuming that the distribution of V_{OPEN} is symmetrical to a standard value, which means that the maximum positive and negative errors are assumed to be in the same amplitude. The parameter set ($V_{\mathrm{COND+}}$, V_{CLR}), which satisfies (8)–(11) under such maximum error distribution, is the optimal solution that we need, as shown in Table I.

Fig. 7 shows the tolerable range of ΔV_{OPEN}. One can see that the tolerable range of ΔV_{OPEN} also decreases continuously as the circuit scales larger.

PSPICE has been used to simulate the proposed memristor-based OR logic circuits. Biolek's threshold memristor model

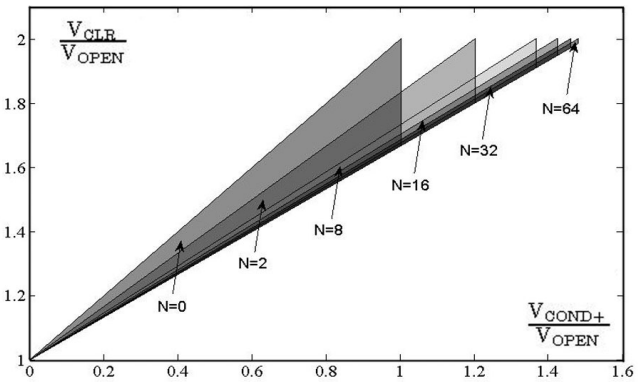


Fig. 6. Normalized solution space of parameter ($V_{\mathrm{COND+}}$, V_{CLR}).

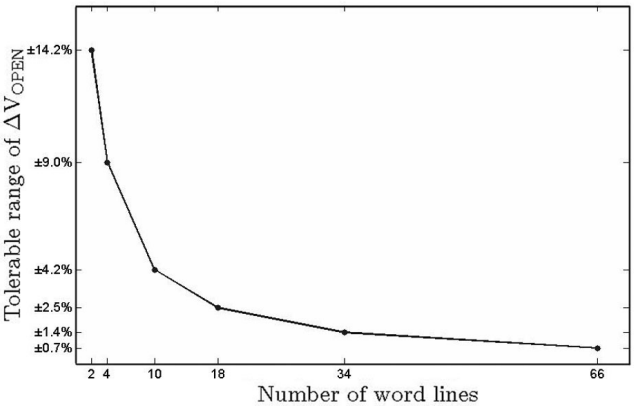


Fig. 7. Tolerable range of ΔV_{OPEN} at different circuit scales.

TABLE I
Optimal Parameters of the Proposed Logic OR Gates at Different Circuit Scales

	($V_{\mathrm{COND+}}$, V_{CLR}) (V)
$N = 0$	(0.86, 1.72)
$N = 2$	(1.09, 1.82)
$N = 8$	(1.30, 1.91)
$N = 16$	(1.38, 1.95)
$N = 32$	(1.44, 1.97)
$N = 64$	(1.47, 1.99)

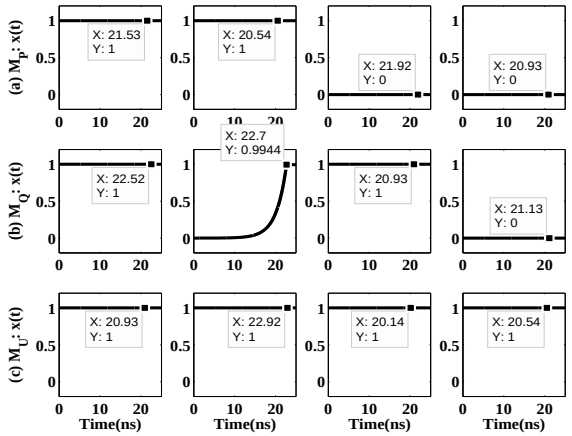


Fig. 8. Simulation results for the OR operation of ten memristors (N=8). (a) State change of M_{P}. (b) State change of M_{Q}. (c) State change of unselected memristors.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSII.2015.2435354, IEEE Transactions on Circuits and Systems II: Express Briefs

TABLE II
COMPARISON OF THE PROPOSED LOGIC GATES WITH OTHER LOGIC GATES

		IMP Logic in [4]	*MAGIC in[9]*	*Proposed Logic*
Operation Step	AND	4	1	1
	OR	3	1	1
	NOT	2	1	2
	NOR	2	1	3
	NAND	3	1	3
Memristor Number	AND	4	3	2
	OR	3	3	2
	NOT	2	2	2
	NOR	3	3	2
	NAND	3	3	2

is used in the simulations. The circuits are designed by using the following parameters: $R_{\mathrm{ON}} = 1k\Omega$; $R_{\mathrm{OFF}} = 1000k\Omega$; $R_{\mathrm{G}} = 0.5k\Omega$; $V_{\mathrm{COND+}} = 0.9V$; $V_{\mathrm{CLR}} = 1.62V$; $V_{\mathrm{PRO}} = 0.6V$; $V_{\mathrm{OPEN}} = 1V$; $D = 1\mathrm{nm}$; $\mu_v = 1\mathrm{e}-7(\mathrm{m}^2\mathrm{v}^{-1}\mathrm{s}^{-1})$.

The switch behavior of memristor in IMP operations is simplified to only relative to the amplitude and polarity of the applied voltage. That is to say if the voltage drop across the memristor exceeds the threshold and lasts for a minimal time interval, resistive state of the memristor will be altered like a discrete switch [4], which is hard to be simulated in SPICE. However, the switch behavior of memristor in our design could be considered as a continuous switch, we can see its dynamic behavior through the PSPICE simulation results and Fig. 8 shows the simulated results of the two input OR circuit when N=8 (shown in Fig. 5(a)). We choose M_{P} and M_{Q} to proceed the OR logic operation and the states of unselected memristors remain the same. When M_{P} is at 1 state and M_{Q} is at 0 state, according to the second line of the truth table in Fig. 4(d), the state of M_{Q} changes from 0 to 1. The simulation results verify the correctness of the design.

Furthermore, the proposed memristor-based logic gates are compared to the IMP-based logic gates [4] and MAGIC gates [9], as shown in Table II. One can see that the number of memristors and the needed operation steps is significantly smaller than that of the IMP-based logic gates. Moreover, compared to MAGIC gates, the number of memistors in the OR and AND logic gates is smaller and the proposed logic gates don't need initialization and all the functionality (including NAND and NOR) can be implemented within crossbar, however, the proposed logic gates need more voltages and cannot reserve previously stored data. Additionally, compared to MRL basic AND and OR logic gates [17], the proposed logic gates have no CMOS inverter and can store logic value in M_{Q}.

V. CONCLUSION

In this brief, a novel design for OR logic circuit structure which completes the *in situ* logic operation on memristor memory is proposed. We also analyze the valid solution space and optimal solution of the parameters which are verified through comprehensive simulations. Furthermore, we simulate the proposed memristor-based logic circuits and demonstrate the feasibility to proceed the OR operation. The proposed OR logic circuit makes logic operations *in situ* logically complete and can reduce the size of the processor, decrease the required memory bandwidth and improve the speed of the logic operations due to less number of operational steps and no initialization.

REFERENCES

[1] K. J. Kuhn, "Considerations for ultimate CMOS scaling," *IEEE Trans. Electr. Dev.*, vol. 59, no. 7, pp. 1813-1828, Jul. 2012.

[2] L. O. Chua, "Memristor-the missing circuit element," *IEEE Trans. Circuit Theory*, vol. CT-18, no. 5, pp. 507-519, Sep. 1971.

[3] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, no. 7191, pp. 80-83, May 2008.

[4] J. Borghetti, G. S. Snider, P. J. Kuekes, J. J. Yang, D. R. Stewart, and R. S. Williams, "'Memristive' switches enable 'stateful' logic operations via material implication," *Nature*, vol. 464, no. 7290, pp. 873-876, Apr. 2010.

[5] S. Shin, K. Kim, and S. Kang, "Reconfigurable stateful nor gate for large-scale logic-array integrations," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 58, no. 7, pp. 442-446, Jul. 2011.

[6] Y. Zhang, Y. Shen, X. Wang, and L. Cao, "A novel design for memristor-based logic switch and crossbar circuits," *IEEE Trans. Circuits Syst. I, Reg. Papers*, 2015, doi: 10.1109/TCSI.2015.2407436.

[7] L. Gao, F. Alibart, and D. B. Strukov, "Programmable CMOS/memristor threshold logic," *IEEE Trans. Nanotechnol*, vol. 12, no. 2, pp. 115-119, Mar. 2013.

[8] G. S. Rose, J. Rajendran, H. Manem, R. Karri, and R. E. Pino, "Leveraging memristive systems in the construction of digital logic circuits," *Proc. IEEE*, vol. 100, no. 6, pp. 2033-2049, Jun. 2012.

[9] S. Kvatinsky, D. Belousov, S. Liman, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser, "Magic–memristor aided logic," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 61, no. 11, pp. 895-899, Nov. 2014.

[10] G. Papandroulidakis, I. Vourkas, N. Vasileiadis, G. C. Sirakoulis, "Boolean logic operations and computing circuits based on memristors," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 61, no. 2, pp. 972-976, Dec. 2014.

[11] K. Kim, S. Shin, and S. Kang, "Field programmable stateful logic array," *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 30, no. 12, pp. 1800-1813, Dec. 2011.

[12] X. Zhu, X. Yang, C. Wu, N. Xiao, J. Wu, and X. Yi, "Performing stateful logic on memristor memory," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 60, no. 10, pp. 682-686, Oct. 2013.

[13] 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)*, vol. 22, no. 10, pp. 2054-2066, Oct. 2014.

[14] I. Vourkas and G. C. Sirakoulis, "Memristor-based combinational circuits: A design methodology for encoders/decoders," *Microelectr. J.*, vol. 45, no. 1, pp. 59-70, 2014.

[15] E. Lehtonen, J. H. Poikonen, and M. Laiho, "Applications and limitations of memristive implication logic," in *13th Int. Workshop Cellular Nanoscale Netw. Appl.*, pp. 1-6, Aug. 2012.

[16] G. Medeiros-Ribeiro, J. H. Nickel, and J. J. Yang, "Progress in CMOS-memristor integration," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, pp. 246-249, Nov. 2011.

[17] S. Kvatinsky, N. Wald, G. Satat, A. Kolodny, U. C. Weiser, and E. G. Friedman, "MRL–memristor ratioed logic," in *Proc. 13th Int. Workshop Cellular Nanoscale Netw. Appl.*, pp. 1-6, Aug. 2012.

[18] W. Wang, T. T. Jing, and B. Butcher, "FPGA based on integration of memristors and CMOS devices," in *Proc. IEEE Int. Symp. Circuits Syst.*, pp. 1963-1966, 2010.

[19] G. S. Rose and H. Manem, "A hybrid CMOS-nano FPGA based on majority logic: From devices to architecture," in *CMOS Processors and Memories*, pp. 139-161, Jul. 2010.

[20] G. Snider, P. Kuekes, and R. S. Williams, "CMOS-like logic in defective nanoscale crossbars," *Nanotechnology*, vol. 15, no. 8, p. 881, Jun. 2004.

[21] M. A. Zidan, H. A. H. Fahmy, M. M. Hussain, and K. N. Salama, "Memristor-based memory: The sneak paths problem and solutions," *Microelectr. J.*, vol. 44, no. 2, pp. 176-183, 2013.

[22] 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.