IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 28, 2019, accepted March 21, 2019, date of publication March 28, 2019, date of current version April 13, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2907976

A Carry Lookahead Adder Based on Hybrid CMOS-Memristor Logic Circuit

GONGZHI LIU, LIJING ZHENG, GUANGYI WANG, YIRAN SHEN, AND YAN LIANG
Institute of Modern Circuits and Intelligent Information, Hangzhou Dianzi University, Hangzhou 310018, China

Corresponding author: Guangyi Wang (wanggyi@163.com)

This work was supported by the National Natural Science Foundation of China under Grant 61771176 and Grant 61801154.

ABSTRACT Memristor-based digital logic circuits open new pathways for exploring advanced computing architectures, which provide a promising alternative to conventional IC technology. In several memristor-based logic design methods, the memristor ratioed logic (MRL) is compatible with traditional CMOS technology. Two kinds of carry-lookahead adders (CLA) based on the hybrid CMOS-memristor structure are proposed, within which one is based on MRL logic, and the other is an improved one that is implemented by MRL universal gate (MRLUG). The proposed CLAs are verified by theoretical analyses and simulations, showing that the proposed design method requires fewer memristors and CMOSs than the IMP-based or CMOS-based CLAs, which means smaller circuit size and lower power consumption.

INDEX TERMS Memristor, logic circuit, CLA.

I. INTRODUCTION

According to Moore's law, it is anticipated that the number of transistors on a chip doubles every 2 years [1]. However, traditional CMOS technology is facing a variety of challenges, in which limitations of the CMOS technology require looking for other technologies to overcome these limitations [2]. The memristor, a new nanoscale memory component, is a powerful competitor for the next generation of computational framework [3].

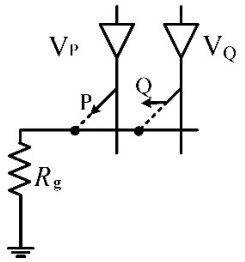
A memristor is a 2-terminal circuit element defined by the relationship between flux φ and charge q [4]. In 2008, the first practical memristor implementation was announced by the research group of Stanley Williams at the HP Labs, which aroused intense interests in academia and industry immediately [5]. The memristor is physically a two-terminal device, whose main features are non-volatility and nanoscale size, making the memristor capable of computing and storing simultaneously. The unique characteristics of memristor make it have good application prospects in analog circuits [6], neural networks [7], [8], logic circuits [9], [10], and memories [11].

Memristors provide a non-conventional computation framework which combines logic operation and storage in the memory itself [12]. Memristor-based logic circuits open a new pathway for exploring advanced computing architectures as a promising alternative to traditional logic circuits.

Thus, several memristor-based logical circuit design methods have been put forward. In 2010, the paper published by HP laboratories first mentioned that a simple circuit consisting of memristors and resistors can realize the material implication logic operation (IMP), and then combining with the FALSE operation to make up a computationally complete logic unit and realize the operation of any Boolean logic function [3]. Fig. 1(a) is a basic schematic of IMP operation, comprised by two memristors P and Q, and a load resistor R_g. Fig. 1(b) is the truth table of IMP operation. The material implication is considered to be a truth function in logic, and the p IMP q can be expressed as "$p \rightarrow q$" in a logical expression. However, the main drawback of the IMP logic lies in performing lengthy sequences. For example, it takes 3 steps to implement the logic NAND gate. In addition, the memristanc values are used to represent the logic states. Therefore, additional peripheral circuits are needed for reading and writing operations, which increases the complexity of the circuit design.

In view of the complexity of the memristor-based IMP logic, a design method similar to the IMP logic circuit is proposed, that is, the memristor-aided logic (MAGIC) [13], which also relies on a sequencer to realize the logic function. Fig. 2 is an AND gate based on MAGIC method. Being consistent with the IMP logic circuit, MAGIC's logic states are also represented by the memristances, where the high

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



(a)

p IMP q→q′		
In	In	Out
p	q	q′
0	0	1
0	1	1
1	0	0
1	1	1

(b)

FIGURE 1. Circuit implementation of the IMP operation and its truth table.

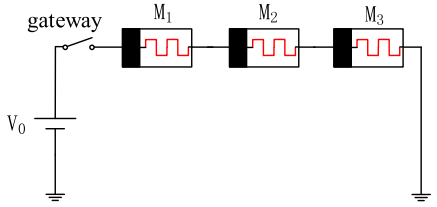


FIGURE 2. Logic AND gate realization using MAGIC.

resistance R_{OFF} and low resistance R_{ON} are considered as logical "0" and logical "1", respectively. The input and output of the logic gate are the memristances. Compared with the method of IMP logic circuit, the MAGIC circuit is simpler and more stable. Different Boolean logic operations can be realized through the series-parallel arrangement of the memristor.

MAGIC-based logic gate includes two sequential stages. The first stage is to initialize the output memristor to the specified memristance. The second stage is to apply voltage V_0 at the gateway. If the voltage (or current) across the output memristor exceeds the threshold voltage (or current), the logic state of the output memristor will change, otherwise the state of the memristor will remain unchanged. Thereby the logic operation can be obtained correctly.

Whether it is IMP logic or MAGIC, the logic gates require a series of sequences to operate the logic function, and the state variables are all defined by the memristanc values. They are incompatible with the traditional CMOS-based circuits, and the extra conversion circuits are needed for converting the memristances to the voltage levels. This implies that the complexity of the circuits is increased.

The memristor ratioed logic (MRL) gate design method suggests that any logic function can be achieved by using memristor and traditional CMOS buffers [14]. The logic states are voltage levels, which are compatible with current CMOS technologies. There are several computational building blocks have been proposed in the last few years, such as multiplier [15], ripple carry adder [16], full adder [17] and oscillator [18].

This paper proposes a carry-lookahead adder (CLA) based on MRL, and then an improved carry-lookahead adder is put forward. A performance comparison between the improved CLA and other CLAs has been conducted and discussed. As for the organization of this paper, Section II introduces the model of the memristor. Section III is the carry-lookahead adder which is composed of MRL logic. Section IV is the improved carry-lookahead adder. Section V is the conclusion.

II. THE MEMRISTOR MODEL

Operations of the memristor-based logic circuits rely on the switching dynamics of threshold-type voltage-controlled bipolar memristor. The *i-v* relation of the memristor model is given by [19]:

$$i(t) = \begin{cases} a_1 x(t) \sin(bv(t)), & v(t) \geq 0 \\ a_2 x(t) \sin(bv(t)), & v(t) < 0 \end{cases} \tag{1}$$

where $v(t)$ and $i(t)$ are the voltage and current across the memristor; $x \in [0,1]$is the internal state variable of the memristor; a_1, a_2 and b are constants greater than zero. The internal state variable satisfies:

$$\frac{dx}{dt} = g(v)f(x) \tag{2}$$

where $g(v)$ gives different thresholds for memristor, and it is given by

$$g(v) = \begin{cases} A_p(e^{v(t)} - e^{V_p}), & v(t) > V_p \\ -A_n(e^{-v(t)} - e^{V_n}), & v(t) < V_n \\ 0, & -V_n \leq v(t) \leq V_p \end{cases} \tag{3}$$

When the internal variable respectively reaches the boundaries x_p and x_n of the memristor, there will be a boundary effect described in [20]. In order to make the memristor switch smoothly at the boundaries, the function $f(x)$ of Eq. (2) is given by

$$f(x) = \begin{cases} e^{-\alpha_p(x-x_p)}\omega_p(x, x_p), & x \geq x_p \\ 1, & x < x_p \end{cases} \tag{4}$$

$$f(x) = \begin{cases} e^{-\alpha_n(x+x_n-1)}\omega_n(x, x_n), & x \leq 1 - x_n \\ 1, & x > 1 - x_n \end{cases} \tag{5}$$

where ω_p and ω_n are two window functions, which are defined as

$$\omega_p(x, x_p) = \frac{x_p - x}{1 - x_p} + 1 \tag{6}$$

$$\omega_n(x, x_n) = \frac{x}{1 - x_n} \tag{7}$$

Fig. 3 is the *i-v* characteristic curve of the memristor. The memristor contains different threshold voltages, and the voltages of different polarities correspond to the different threshold voltages. When the applied voltage exceeds the positive threshold voltage, the memristance will transfer from the HRS (High Resistance State) to the LRS (Low Resistance State). Similarly, when the applied voltage exceeds the negative threshold voltage, it will change the memristance from the LRS to the HRS.

III. MRL LOGIC

From the hysteretic current-voltage loop shown in Fig. 3, it can be seen that the memristance varies according to the direction of current. Fig. 4 (a) is the symbol of the memristor, in which the black side represents the negative polarity of the memristor. When the current flows from A to B, the memristance increases to R_{OFF}; while the current flows from B to A, the memristance decreases to R_{ON} [14]. By using the switching property of the memristor, the AND gate and OR gate can be built, as shown in Fig. 4(b) and Fig. 4(c) [14].

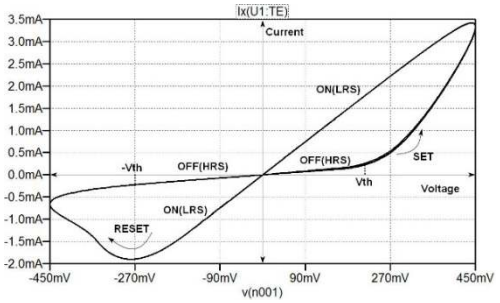


FIGURE 3. The i-v characteristic curve of the memristor.

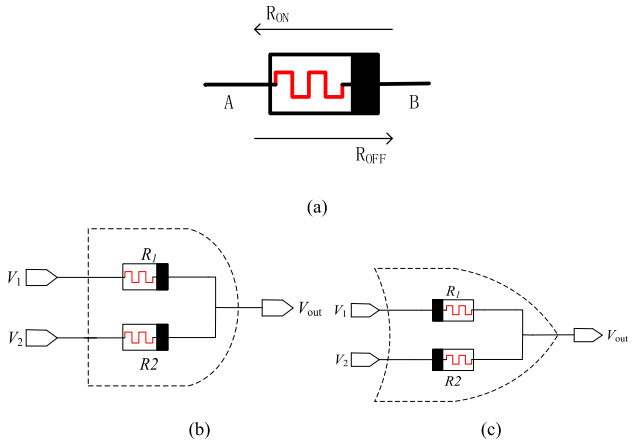


FIGURE 4. Memristor symbol and logic AND and OR gates.

Taking the AND logic gate as an example, the computing process is analyzed. The logic AND gate is composed of two memristors in series. The input terminal is connected to the positive polarity terminal of the two memristors, and the output terminal is the common node of the memristor. The output voltage of the AND gate is determined by the voltage divider of the two memristive devices.

The basic memristor-based logic AND gate, as shown in Fig. 4 (b), consists of two memristors connected in series with opposite polarity, of which the output node is the common node of the two memristors. The truth table of AND gate is shown as Table 1. Logic OR gate is similar to the AND gate, as shown in Fig. 4 (c), which consists of two memristors connected in series with identical polarity.

TABLE 1. Truth table of AND gate.

V_1	V_2	V_{out}
0	0	0
0	1	0
1	0	0
1	1	1

Fig. 5 shows four different cases for the input of the AND gate. Input voltages are V_1 and V_2, V_{cc} represents logic "1", and 0 V represents logic "0". For Fig. 5 (a) and (b), $V_1 = V_2 = 1$ and $V_1 = V_2 = 0$, and the circuits have no current passing from MR_1 to MR_2, so the outputs remain the same as the inputs. For Fig. 5(a), the input voltages are V_{cc} (logic "1"), and then the output voltage is V_{cc} (logic "1") as well; for Fig. 5(b), the input voltages are ground (logic "0"), and then the output voltage is ground (logic "0") as well.

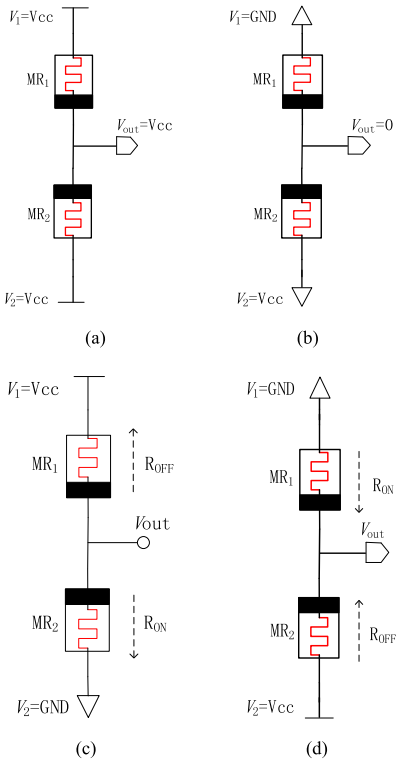


FIGURE 5. MRL-based logic AND computation.

In Fig. 5 (c), $V_1 = 1$ and $V_2 = 0$, and the circuit has a current passing from MR_1 to MR_2, which makes the resistance of memristor MR_1 increase to R_{OFF}, and that of the memristor MR_2 decrease to R_{ON}. Assume that $R_{OFF} >> R_{ON}$, the output voltage is evaluated by the principle of voltage divider:

$$V_{out} = \frac{R_{ON}}{R_{OFF} + R_{ON}} V_{cc} \approx 0 \tag{8}$$

In Fig. 5 (d), $V_1 = 0$ and $V_2 = 1$, and the circuit has a current passing from MR_2 to MR_1, which makes the resistance of

memristor MR_2 increase to R_{OFF}, and that of memristor MR_1 decrease to R_{ON}. Assume that $R_{OFF} >> R_{ON}$, the output voltage is evaluated by the principle of voltage divider:

$$V_{out} = \frac{R_{ON}}{R_{OFF} + R_{ON}} V_{cc} \approx 0 \tag{9}$$

Combined with traditional CMOS buffers, other logic gate circuits can be obtained. For example, a combination of AND (OR) gate and a CMOS buffer can construct a NAND (NOR) logic gate. Other more complex combinational logic circuits, such as the commonly used XOR gates, can also be built. Unlike the implication logic circuits and MAGIC logic circuits, the logic states of the MRL are the voltage levels, which is the same as the traditional logic circuits. Hence, it has a good compatibility with the traditional logic circuits.

IV. THE CARRY LOOK-AHEAD ADDER BASED ON MRL

In order to increase the operation speed of the multi-bit adder and reduce the transmission delays caused by the propagation of carry signal, a traditional carry adder is proposed [21]. In this paper, a 4-bit carry-lookahead adder (CLA) based on MRL is presented, and its circuit structure is shown in Fig. 6. The carry-lookahead adder defines two functions, i.e., the carry generate function G_i and the carry propagate function P_i:

$$\begin{cases} G_i = A_i B_i \\ P_i = (A_i + B_i) \end{cases} \tag{10}$$

Thus, the carry output function of the carry-lookahead adder is:

$$C_o = A_i B_i + (A_i + B_i) CI_i = G_i + P_i CI_i \tag{11}$$

According to the truth table of the 4-bit adder, the output sum function of the i-bit full adder can be deduced as:

$$S_i = A_i \oplus B_i \oplus C_i \tag{12}$$

According to the carry output function and the output sum function, the MRL gate circuit is used to build a carry-lookahead adder. The XOR gate in the circuit is a combinational logic gate consisting of AND gate and OR gate. In addition, the multiple input AND or OR gates are the extension of 2 input AND or OR gates.

V. IMPROVED CARRY-LOOKAHEAD ADDER

By using a suitable combination of MRL gates and a CMOS inverter, a new hybrid CMOS-memristor logic gate circuit, i.e., a universal logic gate (ULG) is presented in [9].

The circuit structure and its simplified circuit symbol are shown in Fig. 7. The main feature of this structure is that the output of logic AND gate is connected to the input of the inverter, the output of logic OR gate is connected to the source of the PMOS, and the output of the inverter happens to form an XOR gate.

The universal logic gate (ULG) uses only 4 memristors and 2 CMOSFET transistors to achieve 3 logic functions at the same time: AND, OR and XOR. By comparison, one MRL-based XOR gate needs 6 memristors, while the number of memristors used in the improved CLA is reduced by two. Thereby the area and power consumption of the integrated circuit design is reduced.

As shown in Fig. 6, the dotted line block X of the MRL-based CLA which contains one AND gate, one OR gate, and one XOR gate, can be replaced by only one ULG.

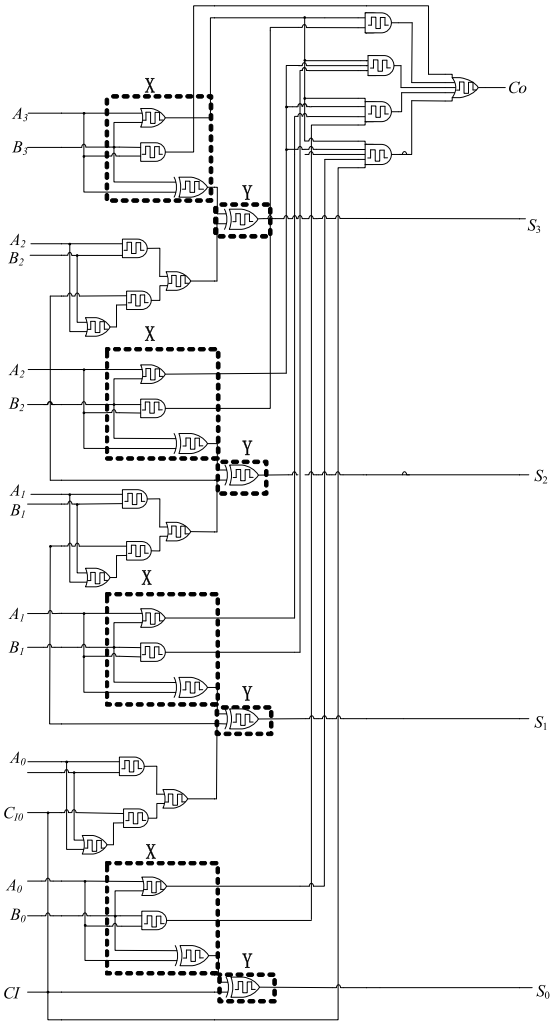


FIGURE 6. The carry-lookahead adder based on MRL logic.

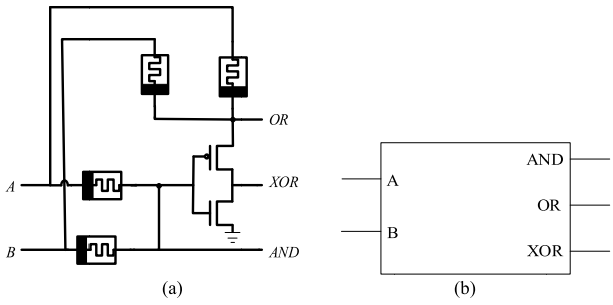


FIGURE 7. The structure diagram of the ULG and Symbol of the ULG.

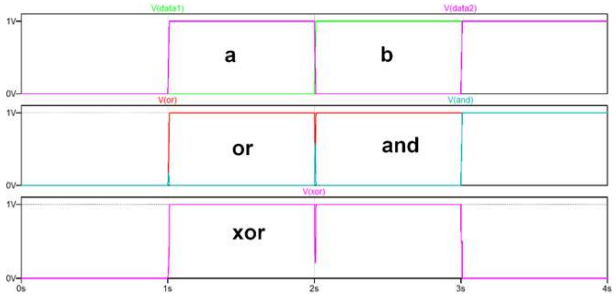


FIGURE 8. Simulation results of ULG.

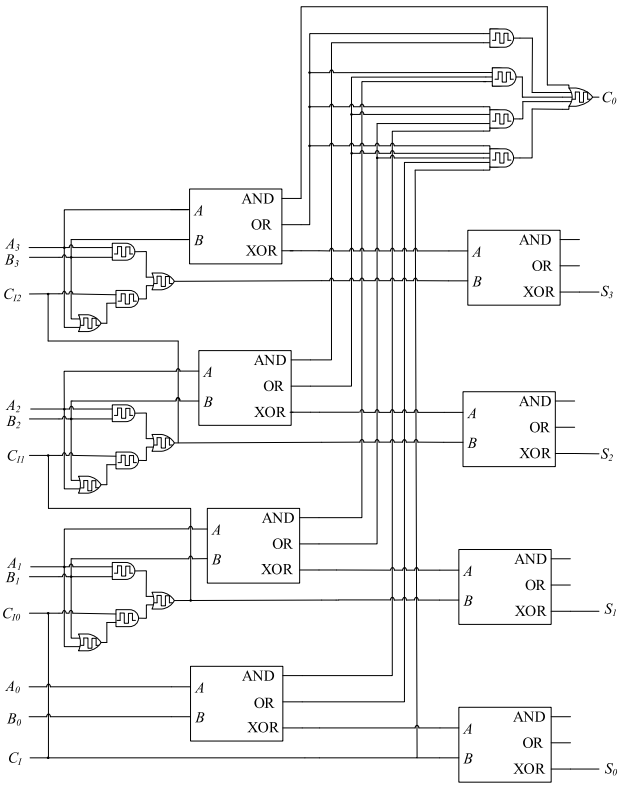


FIGURE 9. Improved carry-lookahead adder.

Furthermore, the dotted line block Y of the CLA can also be replaced by an ULG. Hence, an improved MRL-based CLA circuit with ULGs is obtained, as shown in Fig. 9.

The improved carry-lookahead adder is simulated by LTSPICE. The memristor used in the circuit is based on a memristor model proposed in [19]. Parameters of the model are: $V_p = 0.16$V, $V_n = 0.15$V, $A_p = 4000$, $A_n = 4000$, $x_p = 0.3$, $x_n = 0.5$, $\alpha_p = 1$, $a_1 = 0.17$,$a_2 = 0.17$, $b = 0.05$, $x_0 = 0.11$. Here c_i is from the lowest bit carry, and the waveforms of $b_3 - b_0$ are consistent with those of a_3-a_0. The simulation results are shown in Fig. 10, where $V(s_3)$-$V(s_0)$ are the output sum from the highest bit to the lowest bit, and $V(c_0)$ is the highest bit carry. The sum of the carry-lookahead adder and the carry waveforms are consistent with expectations.

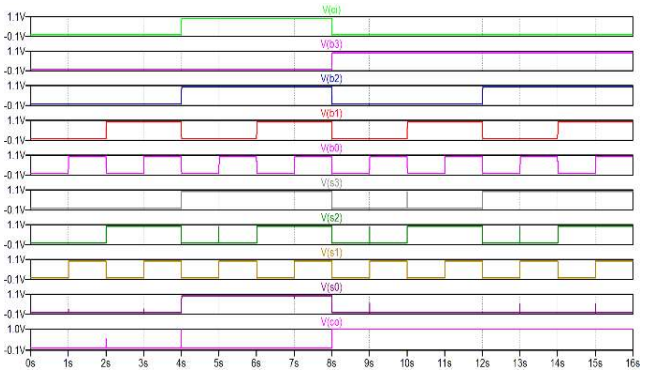


FIGURE 10. The simulation results of 4-bit carry-lookahead adder.

Using the ULGstructure to redesign the carry-lookahead adder can greatly reduce the number of memristors, simplifying the complexity of the circuit design, and thus reduce the power consumption of the circuit.

The performance comparisons of the improved CLA, the MRL-based CLA and the IMP-based CLA reported in [22] are shown in Table 2.

TABLE 2. Comparisons of the proposed CLA with the one reported in [22].

CLA	Input	Read/write circuit	Output	Initialization	No. of memristor	No. of CMOS
MRL(Improved)	Voltage	Unneeded	Voltage	Unneeded	76	80
MRL	Voltage	Unneeded	Voltage	Unneeded	108	96
CMOS	Voltage	Unneeded	Voltage	Unneeded	-	248
IMP logic	M	Needed	M	Needed	88	not specified

The IMP-based CLA takes memristance values as logic state variables to represent the inputs and outputs, in which the high memresistance R_{OFF} and low memresistance R_{ON} are considered as logic "0" and "1" respectively, and the final results of the operation remain in the memristors. Unlike the MRL-based CLA proposed in this paper, the IMP-based CLA requires an additional reading/writing circuit to read the results stored in the memristors, and needs to initialize the memristor to ensure logic operation. The additional reading/writing circuit and initialization circuit are made up of memristors, CMOSs and other circuit elements, so more chip area and more power consumption of the IMP-based logic circuit will be consumed.

However, the proposed MRL-based CLAs take the voltage as logic variable, where high voltage and low voltage represent logic variables "1" and "0" respectively, which is compatible with traditional CMOS circuits. Furthermore, the MRL-based CLAs do not require the initialization of the memristors and the reading/writing circuit.

On the other hand, the number of memristors used in the proposed CLA is less than that used in IMP-based CLA [22]. The size of the memristor is about 3 nm, while the size

of the CMOS is 180 nm. In the logic circuit of the MRL design, a MOSFET can completely accommodate multiple memristors in technology, so the chip area overhead of the MRL-based circuit is quite smaller than that of CMOS-based circuit. The number of memristors and CMOSs used in the improved CLA is further reduced than the MRL-based CLA, so the chip area of the improved CLA is the smallest compared with the other two CLAs, as shown in Table 2.

VI. CONCLUSION

This paper proposes two 4-bit carry-lookahead adders based on MRL logic, which is compatible with the traditional CMOS technology. On this basis, an improved CLA is designed by using ULGs, which decreases the number of the memristors and CMOSs significantly, and reduces the power consumption and the chip area. LTSPICE is used to simulate the designed circuits, and the results are consistent with the expectation. Furthermore, the design can be extended to a multi-bit carry-lookahead adder.

REFERENCES

[1] A. Karimi, A. Rezai, and M. M. Hajhashemkhani, "A novel design for ultra-low power pulse-triggered D-Flip-Flop with optimized leakage power," *Integration*, vol. 60, pp. 160–166, Jan. 2018.
[2] H. Rashidi, A. Rezai, and S. Soltany, "High-performance multiplexer architecture for quantum-dot cellular automata," *J. Comput. Electron.*, vol. 15, no. 3, pp. 968–981, Sep. 2016.
[3] I. Vourkas and G. C. Sirakoulis, "Emerging memristor-based logic circuit design approaches: A review," *IEEE Circuits Syst. Mag.*, vol. 16, no. 3, pp. 15–30, 3rd Quart., 2016.
[4] L. O. Chua, "Memristor-the missing circuit element," *IEEE Trans. Circuit Theory*, vol. 18, no. 5, pp. 507–519, Sep. 1971.
[5] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, pp. 80–83, May 2008.
[6] Y. V. Pershin and M. Di Ventra, "Practical approach to programmable analog circuits with memristors," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 8, pp. 1857–1864, Aug. 2010.
[7] M. Hu, H. Li, Y. Chen, Q. Wu, G. S. Rose, and R. W. Linderman, "Memristor crossbar-based neuromorphic computing system: A case study," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 10, pp. 1864–1878, Oct. 2014.
[8] Y. Zhang, X. Wang, and E. G. Friedman, "Memristor-based circuit design for multilayer neural networks," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 2, pp. 677–686, Feb. 2018.
[9] M. Teimoori, A. Ahmadi, S. Alirezaee, and M. Ahmadi, "A novel hybrid CMOS-memristor logic circuit using Memristor Ratioed Logic," in *Proc. IEEE Can. Conf. Elect. Comput. Eng.*, Mar. 2016, pp. 1–4.
[10] M. Teimoory, A. Amirsoleimani, A. Ahmadi, and M. Ahmadi, "A hybrid memristor-CMOS multiplier design based on memristive universal logic gates," in *Proc. IEEE 60th Int. Midwest Symp. Circuits Syst.*, Aug. 2017, pp. 1422–1425.
[11] J. K. Eshraghian *et al.*, "Maximization of crossbar array memory using fundamental memristor theory," *IEEE Trans. Circuits Syst., II, Exp. Briefs*, vol. 64, no. 12, pp. 1402–1406, Dec. 2017.
[12] 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.
[13] S. Kvatinsky *et al.*, "MAGIC—Memristor-aided logic," *IEEE Trans. Circuits Syst., II, Exp. Briefs*, vol. 61, no. 11, pp. 895–899, Nov. 2014.
[14] 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 Nanosc. Netw. Appl.*, Oct. 2012, vol. 8456, no. 24, pp. 1–6.
[15] A. Kanapyanov and O. Krestinskaya. (May 2018). "Analog multiplier design with CMOS-memristor circuits." [Online]. Available: https://arxiv.org/abs/1805.07680
[16] P. L. Thangkhiew, R. Gharpinde, P. V. Chowdhary, K. Datta, and I. Sengupta, "Area efficient implementation of ripple carry adder using memristor crossbar arrays," in *Proc. 11th Int. Design Test Symp. (IDT)*, Dec. 2016, pp. 142–147.
[17] A. Karimi and A. Rezai, "Novel design for a memristor-based full adder using a new IMPLY logic approach," *J. Comput. Electron.*, vol. 17, no. 3, pp. 1303–1314, 2018.
[18] H. Bao, N. Wang, H. Wu, Z. Song, and B. Bao, "Bi-stability in an improved memristor-based third-order Wien-bridge oscillator," *IETE Tech. Rev.*, vol. 36, no. 2, pp. 109–116, Jan. 2018.
[19] C. Yakopcic, T. M. Taha, G. Subramanyam, R. E. Pino, and S. Rogers, "A memristor device model," *IEEE Electron Device Lett.*, vol. 32, no. 10, pp. 1436–1438, Oct. 2011.
[20] Z. Biolek, D. Biolek, and V. Biolkova, "SPICE model of memristor with nonlinear dopant drift," *Radioengineering*, vol. 18, no. 2, pp. 210–214, Jun. 2009.
[21] B. Parhami, *Computer Arithmetic: Algorithms and Hardware Designs*, 2nd ed. 2010.
[22] A. H. Shaltoot and A. H. Madian, "Memristor based carry lookahead adder architectures," in *Proc. IEEE 55th Int. Midwest Symp. Circuits Syst.*, Sep. Aug. 2012, pp. 298–301.

Authors' photographs and biographies not available at the time of publication.