electronics

MDPI

Review

Progress on Memristor-Based Analog Logic Operation

Yufei Huang [1], Shuhui Li [1], Yaguang Yang [2] and Chengying Chen [1,*]

[1] School of Opto-Electronic and Communication Engineering, Xiamen University of Technology, Xiamen 361024, China; huangyf@xmut.edu.cn (Y.H.)
[2] Institute of Microelectronics of Chinese Academy of Sciences, Beijing 100029, China
* Correspondence: chenchengying363@163.com

Abstract: There is always a need for low-power, area-efficient VLSI (Very Large-Scale Integration) design and this need is increasing day by day. However, conventional design methods based on Metal-Oxide-Semiconductor Field Effect Transistor (MOSFET) devices and Complementary Metal-Oxide-Semiconductor Transistor (CMOS) technology cannot meet the performance requirements. The memristor, as a promising computing and memory integration device, offers a new research idea for conventional logic circuit structure and architecture innovation, given its non-volatility, scalability, low power consumption, fast switching speed, etc. This paper proposes a brief overview of the characteristics and current status of memristor-based logic circuits and analyzes their applications in numerical expression and memory. The benefits and drawbacks of various analog logic circuit structures are summarized and compared. In addition, some solution strategies for these issues are presented. Finally, this paper offers prospects for the applications of memristors in the logic implementation of large-scale memristor arrays, the novel structure of in-memory computing, and neural network computing.

Keywords: memristor; integrated circuits; logic operation; adders

Citation: Huang, Y.; Li, S.; Yang, Y.; Chen, C. Progress on Memristor-Based Analog Logic Operation. *Electronics* **2023**, *12*, 2486. https://doi.org/10.3390/electronics12112486

Academic Editor: Hyunchol Shin

Received: 23 April 2023
Revised: 18 May 2023
Accepted: 30 May 2023
Published: 31 May 2023

1. Introduction

According to the relationship between voltage, current, electric charge, and magnetic flux, Chua proposed the memristor in 1971 as the fourth fundamental electronic component that represents the relationship between electric charge and magnetic flux [1]. However, the memristor concept was not verified for a very long time. In 2008, HP Labs scientists successfully developed the first nano memristive device [2], supporting the theoretical derivation and opening up research opportunities based on memristors. Due to their nanometer size, non-volatility and nonlinearity, memristors are primarily studied and applied in memory, logic operations, and neural synaptic networks. Firstly, nanometer size is critical in following Moor's Law. In the traditional Von Neumann architecture of computers, memory and computation are separate components. To improve computing speed, researchers have widely studied accelerating the processing speed of processors to enhance computer performance. However, the processing speed of processors is much higher than the transmission speed of data, and the transmission time of data is greater than the processing time of processors. This results in significant power consumption and limits the development of computers, commonly known as the "Von Neumann bottleneck" [3]. To overcome this problem, the fundamental solution is to avoid the process of data transmission and achieve in-memory computing. As a result, memristors can be applied not only as memory but also perform logic operations, providing a new idea for breaking the bottleneck of Von Neumann. Secondly, non-volatility is of great significance to the study of computer memory and biological neural synapses. CMOS-based electronic synapses suffer from issues such as high power consumption, limited integration, and an inability to fully replicate the connections of biological neurons [4]. In contrast, the resistance of a memristor can be adjusted and maintained even after power has been turned off, similar

to how a biological neural synapse operates. Furthermore, the dual-port architecture of a memristor is similar to that of a synapse [5]. Thus, the memristor is a promising device for realizing biological neural synapses. Thirdly, nonlinearity represents the infinite states between "1" and "0" theoretically, which means that the changes in output and input can be continuous with time, and are widely applied in the field of binary and multilevel storage, as well as in analog computers, neural networks, and pattern recognition. Memristive devices with in-memory computing technology differ from existing computing system architecture based on independent logical operation and data storage and have greater prospects in artificial intelligence applications. With the development of the semiconductor industry, logic circuits consisting of transistors are widely used. Memristors are nanoscale components that can be compatible with the CMOS process and overcome short-channel effects through mixing with CMOS transistors. Hence, research on logic circuits based on memristors may provide new ideas and methods for designing new forms of logic.

The symbol of a memristive device is shown in Figure 1. The thick black line represents the polarity of the device. The resistance of a memristive device is determined by the direction of the current flowing through it. When voltage is applied at point A, the current flows from point A to point B, the resistance of the device decreases to "Ron", and the output is logical "1". When voltage is applied in reverse, the current flows from point B to point A, and the resistance of the device increases to "Roff", and the output is logical "0".

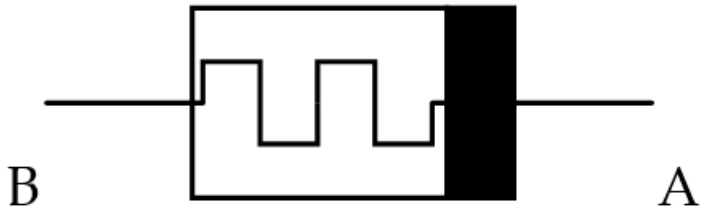


Figure 1. Memristive device symbol.

A large number of memristor experiments indicate that most memristive devices display threshold voltage characteristics instead of threshold current characteristics, and the memristive devices with threshold voltage characteristics are more suitable for logical applications [6,7]. Enlightened by these, logic gate circuits based on memristors will open a new era for the design and application of traditional integrated circuits.

In terms of memristor manufacture, the HP company was the first to successfully implement the manufacturing of a dual-port memristor. The two electrodes of the memristor are made of metal Pt with TiO_2 containing an oxygen deficiency in the middle [2]. Since then, a diverse range of compound materials have been used in memristor manufacturing, including metal-oxide materials [8], two-dimensional (2D) materials [9], and other new materials [10]. Among these materials, metal oxides possess the widest range of types and simple architectures, allowing for compatibility with semiconductor processes in the manufacturing process. Common metal-oxide materials used for memristor manufacturing include NiO [11,12], VO_2 [13], TiO_2 [14,15], SiO_2 [16], Ta_2O_5, and Al_2O_3 [17,18]. Atomically thin 2D materials possess superior electrical, optical, and thermal properties [9]. Currently, the main materials being studied are graphene, boron nitride, and tungsten sulfur compounds [19,20]. However, new materials are being utilized to achieve memristors with specific optimized performance. Perovskite is a research hotspot due to its superior performance in memristors, outperforming traditional memristors in resistance variation speed and switch ratio, and does not require electroforming to accomplish resistance variation. Further research is still needed for this technology [21]. Various materials have been studied and applied to the manufacturing of memristors, and some architectures and systems have also formed in memristor design.

2. Logic Operations Based on Memristive Devices

It has been proven that memristors outperform the traditional CMOS circuits in the field of neuromorphic computing and approximate calculation. One of the primary goals of logic operation based on memristors is to innovate logic circuit architecture and replace traditional digital logic circuits. A general hardware architecture for in-memory computing has been proposed [22–24], which consists of both analog and digital in-memory computing. Because digital in-memory computing uses traditional digital circuits and hasno analog circuit, it can fulfill high-precision in-memory computing, including addition and multiplication calculations [25,26]. Based on the resistive subdivision technique, analog in-memory computing technology can achieve multi-bit calculation; however, the precision and reliability of the circuits still require improvements. Research on logic circuits based on memristor devices focuses on circuit design and circuit architecture, particularly on the design of adders and multipliers. In order to complete the calculation, at first the logical states need to be defined, followed by the establishment of a calculation system. For instance, the establishment of a computing system is based on binary Boolean logic, which is represented by the status of CMOS, where the on and off of CMOS correspond to binary Boolean logic. Memristor-based logic operations are classified as exploring novel logic circuits.

2.1. Material Implication (IMPLY)

In 1910, Whitehead and Russell proposed four basic logic operations, including AND, OR, NOR, and implication logic [27]. The logical state in an IMPLY gate is represented by a memristor resistance instead of voltage or current [28]. The logic function is denoted as p IMPLY q (that is, p implies q, and if p then q), where p and q are variables. The logical relationship in IMPLY logic shows that if p is true, q is true as well. The IMPLY operation is described as the comparison between NOT p and q, where a similarity and a difference in the results are considered, respectively, as true and false. The symbol "→" represents IMPLY logic, where the arrow is pronounced as implication. The expression in digital logic is

$$q' = \overline{p} + q \quad (1)$$

The truth table of IMPLY logic is shown in Table 1. IMPLY logic can be applied to prove cause–effect relation and deductive relation, by introducing zero (setting the logic state to logical "0") to form a complete set [7]. The logic state of memristive IMPLY logic is represented by memristor resistance. The high-resistance state of a memristor represents logical "0", and the low-resistance state of a memristor represents logical "1". Different voltage magnitudes are selected to switch the resistance of a memristive device and therefore to set the memristive logical states. As the resistance value between the high and low resistances of memristors is not infinitely ideal, it is essential to maintain a significant difference between the two values. The proposed logic state is saved in a memristive crossbar array. A nanoscale crossbar array has been studied for the implementation of IMPLY operation, by eliminating the data transmission between memory and processor to provide a more efficient computing mode. Another approach to applying IMPLY logic is through complementary memristor switches that can avoid the effectof sneak circuits on adjacent units.

Table 1. The truth table of p IMP q.

Input p	Input q	Output q′
0	0	1
0	1	1
1	0	0
1	1	1

Multi-bit memory crossbar arrays suffer from complex circuit structure, leakage current, low density, etc., which limits their practical application. IMPLY gates bring out high delays for logic operations and require more steps in the design of read and write circuits [6]. To accomplish IMPLY operations, sequential voltages need to be applied at different points in the circuits, and the operation result is stored in one of the same memristors as the input, which causes reading difficulties. Meanwhile, extra circuits are required to assist logic operations, which increases the power consumption and circuit complexity. Bickerstaff and Swartzlander [29] were the first to design an IMPLY-based multiplier, which had a complicated procedure involving 73 steps to complete a 4-by-4 array multiplier. The research shows the advantages of memristive logic operation in density and speed, but the greater delay and area consumption of an IMPLY gate are still problems. Shaltoot and Madian [30] compared two different memristive architectures of carry lookahead adders with conventional carry lookahead adders and found that with an increase in the number of bits, the delay and implication cycles of the two memristive carry lookahead adder architectures had more decrease than the conventional carry lookahead adder, showing better performance. However, each logic function requires many memristors and operation steps, resulting in higher demand on components and increased operation time and complexity of circuit structure. The CMOS/nanowire/Molecular hybrid (CMOL), by combining nanotechnology and traditional CMOS implementations, has high integration density as nanotechnology and different logic functions as CMOS. It is considered that CMOL has the most potential as a technique to replace CMOS in the implementation of IMPLY logic by vector operation [31]. However, there are still many defective nanodevices in CMOL circuits during manufacturing [32].

The Memristors-As-Drivers Gate (MAD gate) is an alternative approach to implementing IMPLY logic. By combining IMPLY operations with readout circuits, MAD gates would offer wider applicability and greater integrity than other approaches. Guckert and Swartzlander [33] proposed a MAD gate, which uses three memristors and two drivers to accomplish a Boolean operation. The value of input memristors is selected to drive output memristors so that the delay of a Boolean operation can be simplified to a single step and the power consumption is reduced to 30fJ. However, the large-size components, such as resistors and switches, used in MAD gates make them unsuitable for large-scale integrated circuits. Additionally, since memristor resistances represent the input and output values of a MAD gate, the initialization of memristive resistance before each operation is necessary, which limits the application in cascade circuits.

Furthermore, IMPLY logic can be applied to the design of an adder, demultiplexer, encoder, priority encoder, decoder, comparator, etc. [34,35].

2.2. Boolean Logic

As with traditional CMOS circuits, building a logic family by a memristive logic gate based on Boolean operation is a common circuit design method [31]. The main difference between an AND logic gate and an OR logic gate based on Boolean logic is the polarity position of a memristive device within the structure. For the AND logic gate, if memristors A and B are both logical "0" or logical "1", there will be no current flow through the logic gate and the memristive resistance will remain constant. In this case, the voltage of the output is $V_{out} = a = b$. If memristors A and B have different logical states, current will flow from the memristor with logical "1" to the memristor with logical "0", causing the resistance of the memristor with logical "1" to gradually increase to "R_{off}", while the other decreases to "R_{on}". As a result, the output voltage V_{out} is chosen to be zero to accomplish an AND operation.

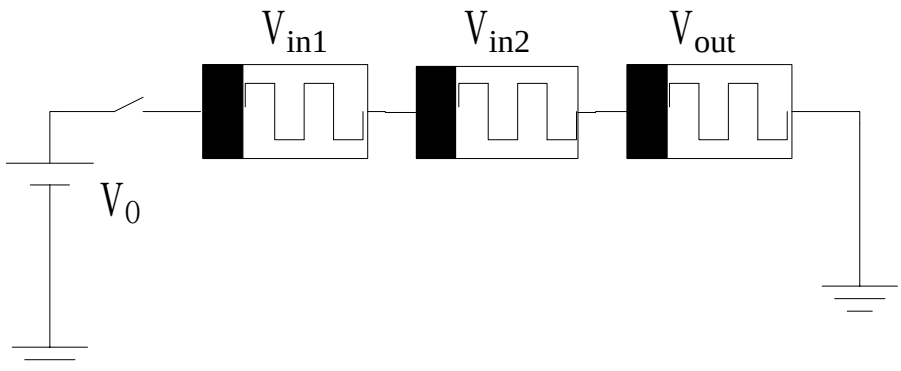
To ensure compatibility with a standard CMOS process, it is necessary to convert the logical state of the output into either a voltage or a current. This approach provides advantages in Computer-Aided Design and allows for easy circuit extension based on circuit units when integrated with CMOS logic. In basic Boolean logic, AND and OR

operations can be accomplished solely through the use of memristors, but to perform the NOT operation, a CMOS inverter is required [7].

Scouting logic is a type of Boolean operation that allows for limiting all logic gate operations into a single read operation [36]. This method can reduce the upset rate of the memristor without compromising its lifespan. Another way to achieve Boolean logic is to use two or more lines of memory cells to perform bit operations. However, this method is limited in its ability to perform complicated operations.

2.3. Memristor-Aided Logic (MAGIC)

In 2014, Kvatinsky et al. [37] proposed Memristor-Aided Logic (MAGIC), which uses the memristors with previous data as input and output. There is no need for a complicated structure to execute MAGIC, which needs only a voltage to perform various basic logic gates, as shown in Figure 2. The logical state of a MAGIC gate can be represented by the value of memristive resistance, where high resistance represents logical "0" and low resistance represents logical "1". The high resistance is named "R_{off}", and the low resistance is named "R_{on}". The logical states of the memristor represent the input and output values of a MAGIC gate. Compared to an IMPLY logic gate, a MAGIC gate requires separate memristors for the input and output. The inputs depend on the initial logical states of input memristors, and the output is determined by the final logical state of the output memristor. The input and output states of MAGIC logic are stored in separate memristors, ensuring a stable and potentially repeatable operation process. MAGIC can accomplish NOT, AND, NAND, OR, and NOR operations, which form a complete set. Figure 2 displays the various architectures of MAGIC-based NAND gates, NOR gates, and other gates, which vary in accordance with the amount of input data. Multi-bit input logic gates require a greater number of serial or parallel memristors. Therefore, current circuit designs and simulations based on MAGIC primarily concentrate on dual-input logic gates, which can be performed in a crossbar array [38]. The application of MAGIC NOR gates in memristive crossbar arrays greatly improves circuit performance and reduces circuit power consumption. All basic operations based on MAGIC require more than one clock, so the operation speed of MAGIC is slightly slow. In addition, MAGIC gates suffer from state drift and lack signal restoration, which puts forward a higher demand to circuit design.



(a)

Figure 2. *Cont.*

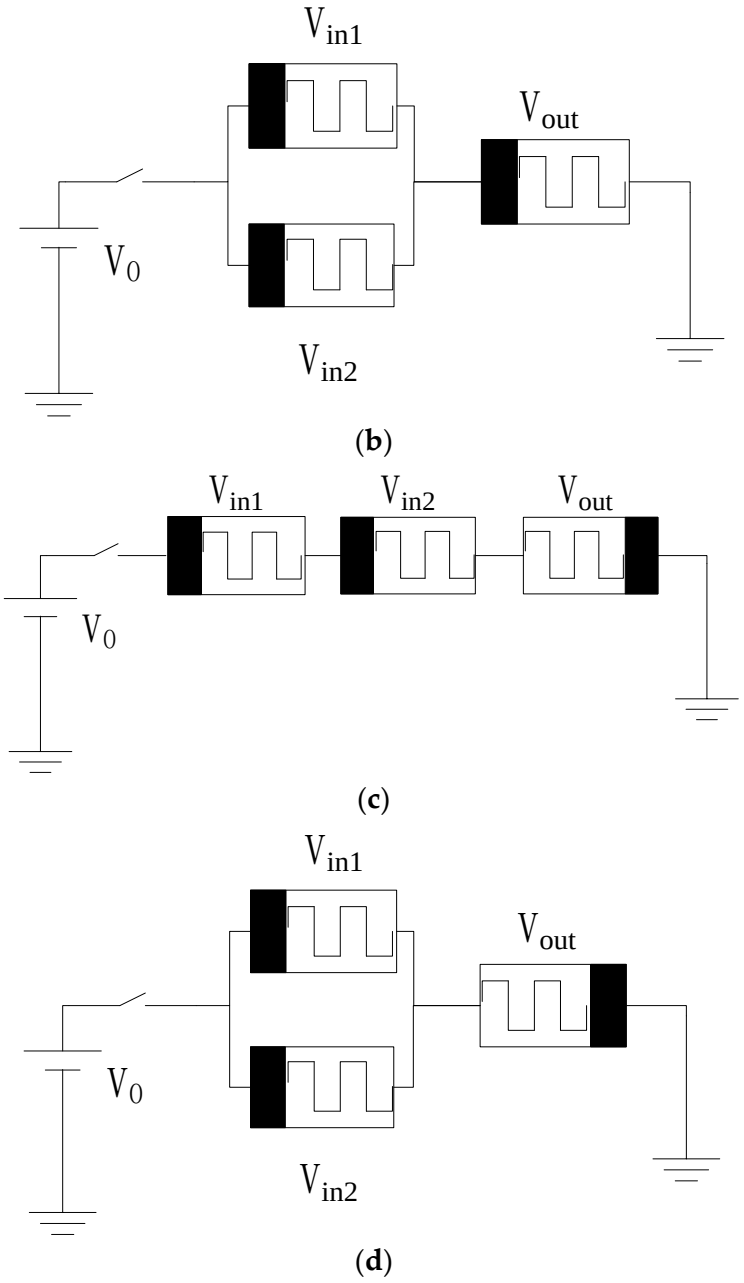


Figure 2. MAGIC gates [37]. (**a**) Schematic of a two-input AND gate. (**b**) Schematic of a two-input OR gate. (**c**) Schematic of a two-input NAND gate. (**d**) Schematic of a two-input NOR gate.

2.4. Memristor Ratioed Logic (MRL)

It can be challenging to integrate standard CMOS logic and memristive logic in a crossbar array. However, in order to achieve compatibility between memristive devices and CMOS in logic circuits, there are a few requirements that must be met. Firstly, the process of memristors should be compatible with CMOS technology. Secondly, the input and output logical states must be represented by voltage instead of resistance. Thirdly, the additional circuitry required for connecting the memristive device layers to the CMOS layers should be minimized. To address these concerns, Kvatinsky et al. [39] proposed a memristor-based logic that is compatible with CMOS logic and is named Memristor Ratioed Logic (MRL). The schematic of MRL AND, OR, NAND and NOR logic gates is demonstrated in Figure 3. Boolean AND/OR operations are accomplished by memristors with programmable resistance, while NOT operation is achieved by CMOS inverters, making it a complete logic family. In MRL, memristors serve as computing components instead of memory for saving

logical states. The logical state in MRL is represented as a voltage, where logic "1" and logic "0" correspondto high and low voltages, respectively, similar to CMOS. The initial logical state of a memristive device does not affect the output logical state, but only the computing efficiency. Furthermore, MRL logic enables more convenient logic operations without the need for extra reading-writing circuits but cannot implement computation and memory simultaneously. Due to the segregation of computation and memory, MRL is not an effective solution to the Von Neumann bottleneck problem, but rather a method to achieve compatibility between memristors and CMOS circuits. MRL-based circuit designs have been extensively studied, with adders and comparators being accomplished through simulations. Vinukollu et al. [40] proposed an MRL-based four-bit carry lookahead adder. Without changing any operations, this carry lookahead adder employs MRL gates to replace some other logic gates, which reduces the number of memristors, minimizes circuitry and power consumption, and decreases the delay time. Wang et al. [41] proposed a D flip-flop and a JK flip-flop based on MRL. The D flip-flop consists of five memristors and an NMOS transistor, while the JK flip-flop is composed of seven memristors and two NMOS transistors. Compared to the traditional design method, the flip-flop based on MRL requires fewer MOSFETs. Additionally, due to the nanoscale size of memristors, MRL-based flip-flops have as impler circuit structure, lower power consumption, and smaller circuit area. Paramasivam et al. [42] proposed a two-bit CMOS digital comparator utilizing MRL gates. MRL gates, consisting of memristive devices and CMOS inverters, decrease the number of memristors needed in the circuit, leading to reduced power consumption, a smaller area, and lower computing complexity for the comparator. Experimental results show an 18.74% reduction in circuit power consumption compared to traditional CMOS logic, and a 32.14% decrease in circuit area compared to resistance threshold logic.

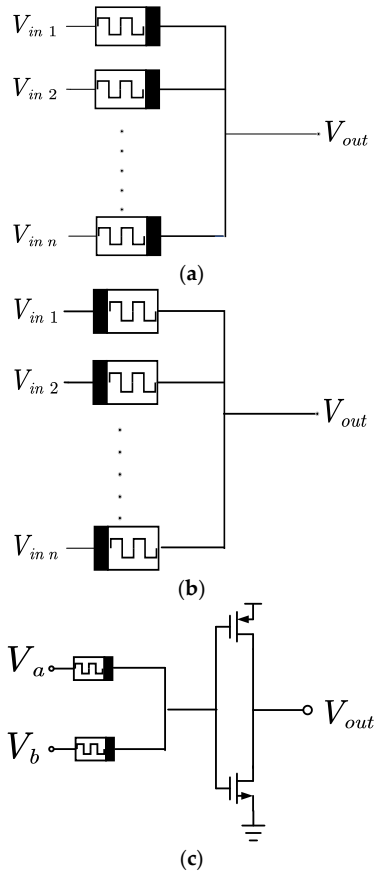
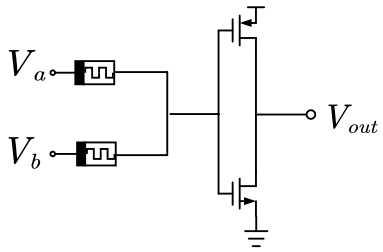


Figure 3. *Cont.*



(**d**)

Figure 3. MRL logic gates [39]. (**a**) Schematic of an N-input AND gate. (**b**) Schematic of an N-input OR gate. (**c**) Schematic of a two-input NAND gate. (**d**) Schematic of a two-input NOR gate.

2.5. Memristive Threshold Logic (MTL)

Memristive threshold logic is a non-traditional form of logic that utilizes memristors to determine input weights and achieve threshold control. The output is generated if the comparison surpasses the threshold. Lageweg et al. [43] introduced the Linear Threshold Gate (LTG), which uses tunnel junctions, capacitors, and voltage sources. It can perform any linear separable Boolean operations as [44]:

$$\mathrm{F(X)} = \mathrm{sgn}\{\mathcal{F}(\mathrm{X})\} = \begin{cases} 0, \text{ if } \mathcal{F}(\mathrm{X}) < 0 \\ 1, \text{ if } \mathcal{F}(\mathrm{X}) \geq 0 \end{cases} \tag{2}$$

$$\mathcal{F}(\mathrm{X}) = \sum_{\mathrm{i}=1}^{\mathrm{n}} \omega_{\mathrm{i}} \mathrm{x}_{\mathrm{i}} - \psi \tag{3}$$

where $\mathrm{x_i}$ are Boolean inputs and ω_i are integer weights of the Boolean inputs. LTG is capable of comparing the weighted sum of inputs, represented by $\sum_{\mathrm{i}=1}^{\mathrm{n}} \omega_{\mathrm{i}} \mathrm{x}_{\mathrm{i}}$, with threshold value ψ. If the input weighted sum is equal to or greater than the threshold value, the output of LTG will be logical "1". Conversely, if the input weighted sum is smaller than the threshold value, the output of LTG will be logical "0" [43,44]. Voting logic, which utilizes binary inputs and equal weights, is a subset of threshold logic. It is essential to note that the numerical expressions of threshold logic form the basis of neural computation. The simulation of threshold logic designs has verified the optimization in both area and power consumption. To accomplish threshold logic, the design that integrates programmable CMOS and memristors has been proposed, where CMOS logic is applied for signal amplification and inversion [45]. Another design that adds current mirror at the input to perform threshold logic is introduced in [46]. A PMOS current mirror is used as a current comparator for the purpose of comparing the sum of currents flowing through all the input memristors with a pre-specified threshold value. The input memristors determine the input weights of the threshold gate and convert the input voltages into currents, with the magnitude of the current generated by each memristor being defined by its respective weight. By comparing the input current with the reference current, namely the threshold value, the output of the threshold gate is determined as either logical "1" or logical "0", depending on whether the sum of the input current exceeds the threshold value or not, respectively. However, as the application of a current mirror can lead to a reverse flow of current, a design has been proposed in [47] whereby transfer transistors replace the current mirror after the input memristors. Moreover, the transistors are equipped with controllable switches that enable the reference current to be increased up to one to six times. In terms of current comparison, the Traff comparator [48], which comes with positive feedback properties, has a faster process speed when compared to the design proposed in [46]. Consequently, the threshold logic gates [47] that come with programmable input weights and threshold values provide a higher degree of flexibility for the implementation of logical functions.

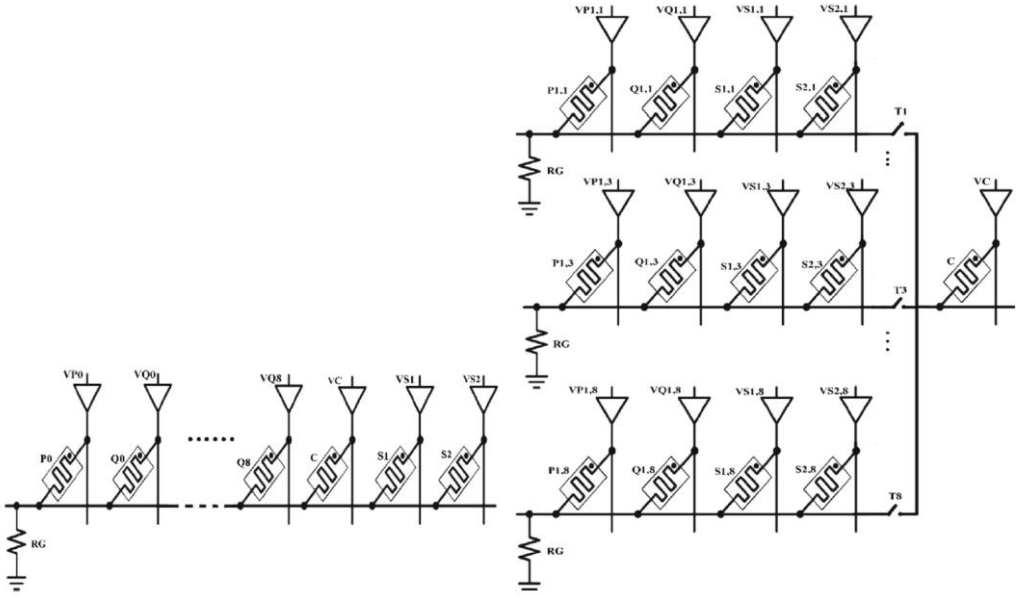
3. Memristor-Based Logic Gates

Compared to traditional logic circuits, memristor-based logic circuits have two major breakthroughs. First, logic gates with programmable resistance of memristors break the Von Neumann bottleneck and promote the development of in-memory computing. Second, designing more efficient logic circuits based on the electric characteristics of memristors can greatly decrease the area and power consumption of the circuits, improve the energy efficiency ratio, and enable the implementation of complete Boolean logic. There are various logic implementations [49], e.g., material implication (IMPLY), memristor-aided logic (MAGIC), linear threshold gate (LTG), memristors-as-drivers (MAD) gate, and complementary resistive switch (CRS). By combining basic logic gates with logical operations, these implementations can accomplish complex Boolean logic. Based on these approaches, recent research has proposed a series of improvement measures and ideas to overcome the shortcomings of different logic implementations, with optimized designs represented by adders and multipliers being introduced continuously.

There have been numerous studies conducted on IMPLY operations. However, the many disadvantages of IMPLY logic, such as multiple operation steps, complex control processes, weak concurrency control, long calculation cycles, and high consistency requirements for memristors, pose significant challenges to IMPLY-based circuit design. To address these issues, Shaltoot et al. [30] proposed a memristive carry lookahead adder that simplifies IMPLY logic to optimize logical expressions. This adder outperforms traditional carry lookahead adders as the input bit width increases. Guckert et al. [50] proposed a memristor-based ripple carry adder that required $2n + 19$ steps and $7n + 1$ memristors. However, the study revealed that the delays of various devices differ, which means that the performance of the device cannot be solely judged by the number of steps. Ahmad et al. [51] considered that the ripple carry adder proposed by Guckert et al. consumes too much area, and they proposed new eight-bit memristive full adder architectures, as shown in Figure 4. The parallel-serial structure requires $4n + 1$ memristors and $5n + 16$ steps for implementation, while the serial structure only needs $2n + 3$ memristors and 21n steps for implementation. Rohani et al. [52] proposed a semi-parallel full-adder architecture that implements an adder in 17n steps using $2n + 3$ memristors. This architecture achieves parallel operation in a one-bit adder by utilizing a parallel architecture for reference. However, due to the complicated operation steps of IMPLY logic, improving the process speeds of these adders is challenging, and the leakage currents of IMPLY operations persist. Hence, nonvolatile adders based on other operations and memristor-CMOS hybrid adders have been the focus of recent research.

Siemon et al. [53] proposed two types of CRS-based serial adders, which require $2n + 2$ memristors for implementation in $2n + 4$ steps and $n + 2$ memristors for implementation in $4n + 5$ steps. Since CRS logic can be performed in a passive memristive crossbar array and has no serious leakage current or signal degradation like IMPLY logic, CRS-based circuit design is an effective way to fulfill in-memory computing. Talati et al. [54] proposed a MAGIC-based memristive nonvolatile adder, which outperforms IMPLY logic in terms of speed and power consumption. Guckert et al. [33] proposed MAD gates based on IMPLY logic and implemented an n-bit ripple carry adder in $n + 1$ steps using 8n memristors. Wang et al. [55] proposed a one-bit full adder based on MIG logic, which can accomplish nondestructive readout in four steps with only three memristors and bit extension in twosteps. Cui et al. [56] proposed four-step RRAM-based logic gates to implement the pipeline structure of logic gates. An n-bit adder constructed from the gates has a good time sequence and is implemented in $2n + 2$ steps. Furthermore, hybrid approaches have been studied for adder design. Guckert et al. [50] proposed an MRL-based memristor-CMOS hybrid adder that uses 14N memristors and 12N MOSFETs to implement the delay in $3n + 4$ steps. The study shows that the switch time of the memristor-CMOS hybrid circuit is shorter than that of the traditional CMOS circuit. Since the delay of level logic is only affected by the propagation delay of level signal, the process speed of level logic circuits is much higher than that of voltage-controlled nonvolatile logic circuits. Rajendran et al. [46] proposed a

memristive threshold logic circuit. By comparing with the CMOS capacitor threshold logic gate and four-bit CMOS look-at-table, the advantages of the design have been proven in power consumption, delay, and the number of transistors used. Maan et al. [57] proposed a half adder and a four-bit carry lookahead adder based on voltage-controlled memristive threshold logic gates, which have lower power consumption and area compared to the traditional CMOS circuits. In addition, numerous studies have shown that memristor-based logic circuits have more advantages in circuit design, such as subtractors, comparators, and multipliers.



(**a**) (**b**)

Figure 4. Memristor-based full adder architecture [51]. (**a**) Schematic of a serial 8-bit full adder based on memristors. (**b**) Schematic of an 8-bit parallel-serial memristive full adder.

As key components of digital signal processors (DSPs), central processing units (CPUs), and communication systems, multipliers consist of adder accumulators, shifters, etc. Memristive multipliers offer new ideas and perspectives for multiplier design to address the shortcomings of traditional logic implementations. Many optimized multiplier designs have been proposed as a result.

Logic operations based on IMPLY logic require plenty of memristors and steps. Several approaches have been proposed to optimize IMPLY-based multipliers. Guckert et al. [58] proposed a design that requires $7N + 1$ memristors and $2N^2 + 21N$ steps for implementation. Haghiri et al. [59] optimized the logic expression to reduce the operation steps of a two-bit multiplier to just eight steps. Radakovits et al. [60] proposed an IMPLY logic multiplier based on a half serial adder, which allowed for inter-bit parallelism in additive operation. This design uses $2N^2 + N + 2$ memristors and can accomplish operations in $\log_2 N(10N + 2) + 4N + 2$ steps. Lee et al. [61] proposed an MRL-CMOS hybrid reconfigurable multiplier to address the limitations of MRL-based logic operations. The circuit is relatively simple and has been shown to reduce area and power consumption compared to traditional multipliers under the same delay. Teimoory et al. [62] developed a two-bit multiplier based on MRL compound logic gates. Their study demonstrated that a two-bit multiplier can be implemented in a single step by using 16 memristors and 8 transistors, with an optimized circuit structure. Chen et al. [63] proposed a ternary multiplier based on ternary memristors, where the inputs are square signals represented by threelevel states and the outputs are ternary level signals.

The features of the recently published memristor-based circuit architecture mentioned above are shown in Table 2. The studies mentioned above have highlighted the short-

comings of the main proposed designs for memristor-based logic circuits. However, these approaches have provided valuable design ideas and proved the tremendous potential of memristor-based nonvolatile logic circuits. Unlike CMOS operations, memristor logics are based on time and can achieve different operations in the same circuit structure by varying the time. This feature gives memristor logics high reconfigurability and nonvolatility, thereby improving the flexibility of operations and saving the required area. Although the process speed of CMOS circuits is determined by the RC delay, memristor logics are limited by the number of operation steps required to achieve a specific function. As the function complexity increases, the number of steps also increases, resulting in lower process speeds. Therefore, designing memristive nonvolatile operations based on time can enhance the operation reconfigurability, but sacrifice some computational efficiency at the same time. Memristor-based nonvolatile operations can accomplish the integration of computing and memory in several steps, which leads to low process speeds and restricts the use of memristors for complex logic functions. Consequently, research efforts have focused on reducing the number of operation steps and improving the process speeds of memristor logics.

Table 2. Features of the recently published memristor-based circuit architecture.

Reference	Logic Implementations	Total Number of Memristors	Number of CMOS	Total Number of Steps	Circuit Designs
[50]	IMPLY	$7n+1$	-	$2n+19$	Parallel adder
[50]	Boolean	14n	12n	$3n+4$	Hybrid-CMOS adder
[51]	IMPLY	$2n+3$	-	21n	Serial full adder
[51]	IMPLY	4n+	-	$5n+16$	Parallel full adder
[52]	IMPLY	$2n+3$	-	17n	Semi-parallel fulladder
[53]	CRS	$2n+2$	-	$2n+4$	Serial adder
[53]	CRS	$n+2$	-	$4n+5$	Serial adder
[33]	IMPLY	8n	-	$n+1$	Ripple carry adder
[54]	MAGIC	$22n-3$	-	$15n+1$	Scheme-1 of adder
[54]	MAGIC	$13n-3$	-	$10n+3$	Scheme-2 of adder
[55]	Majority-inverter graph (MIG)	3	-	4	1-bit full adder
[58]	IMPLY	$7n+1$	-	$2N^2+21N$	Multiplier
[59]	IMPLY	20	-	8	Binary multiplier
[60]	IMPLY	$2N^2+N+2$	-	$\log_2 N(10N+2)+4N+2$	Multiplier
[62]	MRL	16	8	1	2-bit multiplier

In memristor logics, there are various ways to perform logic operations, but it is essential to improve the sensitivity to changes in logical states when using memristive operations. The key factor in memristor operations is the state transition, which occurs under particular conditions. Since the state transitions are achieved under many given conditions, the flexibility of logical operations is increased, resulting in functionally complete logics and improved process speeds. However, this can also lead to insensitivity to changes in logical states. To address this issue, circuits based on hybrid memristor–CMOS technology with strong sensitivity to logical states require further research and development as a promising way for future development.

As a promising device for computing and memory integration, the memristor provides a new research idea for conventional logic circuit structure and architecture innovation, thanks to its non-volatility, scalability, low power consumption, fast switching speed, etc. [64]. However, the logic operations based on memristors still face many problems. Firstly, the resistance of a memristor is not stable enough. The ions in a memristor migrate under the influence of any electric field, causing the drift of memristor resistance over time [46]. Secondly, the integration of memristive devices presents challenges. The sneak path problem in a memristor crossbar array leads to large leakage currents and high power

consumption [65,66]. The half-select problem caused by the random variation of memristor resistance also results in the deterioration of device performance [65]. Furthermore, there are other issues to consider, e.g., hybrid memristor-CMOS-based multi-bit comparators that suffer from voltage decay [42]. Additionally, memristor-based threshold logic gates require long delay times [44] and lack effective tools for practical applications in very large-scale integration.

In addition to the applications in analogy logic operations, memristive devices have also found wide use in artificial neural networks, signal processing, image processing, field programmable gate arrays (FPGAs), etc. [40,46,67–69]. In neuromorphic computing systems, memristor-based logic circuits require fewer components compared to traditional CMOS-based logic circuits, resulting in reduced power consumption and area. Furthermore, due to their resistive and nonvolatile properties, memristors show performance similar to that of neural synapses, making the combination of memristor arrays with CMOS circuits a research hotspot with extensive potential applications in the future. In image storage and processing, studies of image processing algorithms based on memristive crossbar arrays are currently in their initial stages, primarily focused on image recognition and video analysis. As one of the main programmable logic devices, FPGAs have a large number of programming circuits designed to accomplish different functions. Therefore, the technology of programming circuits is crucial to FPGAs. Currently, SRAM-based programming technology is the mainstream of FPGA programming technology [70]; however, its volatility is the biggest drawback. To maintain configuration data, additional off-chip nonvolatile memory is required, which not only increases circuit area but also poses security issues such as data loss. Fortunately, memristor-based FPGA designs offer non-volatility characteristics and are compatible with CMOS technology. Moreover, they have higher density and lower power consumption, making them a promising candidate to overcome the limitations of SRAM in FPGA programming technology [71]. In future, memristors can not only develop new circuit structures but also combine with other physical characteristics to accomplish multi-port memristive devices, enhancing their controllability, practicability, and finally their application inreal-life scenarios.

4. Conclusions

As CMOS devices approach their physical size limitations, memristor-based logic gates are becoming a promising technology for computing and memory integration that can meet the numeracy required for processing big data. Logical applications need abundant computations, and memristor devices need to meet the increasing demand for higher computing speeds, durability, and uniformity. Therefore, the optimization of memristor uniformity has become a research hotspot. In circuit integration, the high-density integration of memristors can be achieved through the use of three-dimensional memristor crossbar arrays.

To date, research on memristor-based analog logic operations has mainly focused on the optimization of basic logical operations, algorithms, and operation processes in small-scale integration circuits. However, there is a lack of research on in-memory computing architectures based on memristors. In the future, research will shift towards the logic implementation of large-scale memristor crossbar arrays, developing new integrated in-memory computing architectures, and exploring engineering practices for logic circuits based on in-memory computing. Although memristor-based circuit structures and architectures have shown potential in the field of memristor computing, there are still significant technical challenges that need to be addressed. These include problems with leakage current, reliability, and memory in circuit integration. In the short term, memristor-based circuits cannot replace traditional CMOS circuits in computing systems, and it remains difficult to put idealized memristive devices into practice. Overall, there are numerous technical issues in circuit architectures that need to be tackled in order to fully exploit the potential of memristor-based computing.

Author Contributions: Conceptualization, C.C.; funding acquisition, C.C. and Y.Y.; methodology, C.C., Y.H. and Y.Y.; formal analysis, Y.H. and S.L.; investigation, S.L. and Y.Y.; writing—original draft preparation, Y.H. and S.L.; writing—review and editing, C.C. and Y.Y.; supervision, C.C.; project administration, Y.H. and S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Xiamen Youth Innovation Fund Project (3502Z20206074) and Major science and technology projects of Xiamen (3502Z20221022).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chua, L. Memristor-The Missing Circuit Element. *IEEE Trans. Circuit Theory* **1971**, *18*, 507–519. [CrossRef]
2. Strukov, D.B.; Snider, G.S.; Stewart, D.R.; Williams, R.S. The missing memristor found. *Nature* **2008**, *453*, 80–83. [CrossRef]
3. Geer, D. Chip makers turn to multicore processors. *Computer* **2005**, *38*, 11–13. [CrossRef]
4. Kuzum, D.; Jeyasingh, R.G.D.; Lee, B.; Wong, H. Nanoelectronic Programmable Synapses Based on Phase Change Materials for Brain-inspired Computing. *Nano Lett.* **2011**, *12*, 2179–2186. [CrossRef] [PubMed]
5. Wang, L.; Wang, X.; Duan, S.; Li, H. A Spintronic Memristor Bridge Synapse Circuit and the Application in Memrisitive Cellular Automata. *Neurocomputing* **2015**, *167*, 346–351. [CrossRef]
6. Cho, K.; Lee, S.-J.; Eshraghian, K. Memristor-CMOS logic and digital computational components. *Microelectron. J.* **2015**, *46*, 214–220. [CrossRef]
7. Singh, A. Design and Analysis of Memristor-based Combinational Circuits. *IETE J. Res.* **2018**, *66*, 182–191. [CrossRef]
8. Ha, S.D.; Ramanathan, S. Adaptive oxide electronics: A review. *J. Appl. Phys.* **2011**, *110*, 14. [CrossRef]
9. Xu, R.; Jang, H.; Lee, M.H.; Amanov, D.; Cho, Y.; Kim, H.; Park, S.; Shin, H.-J.; Ham, D. Vertical MoS_2 Double-layer Memristor with Electrochemical Metallization as an Atomic-scale Synapse with Switching Thresholds Approaching 100 mV. *Nano Lett.* **2019**, *19*, 2411–2417. [CrossRef]
10. Yang, C.H.; Seidel, J.; Kim, S.Y.; Rossen, P.B.; Yu, P.; Gajek, M.; Chu, Y.H.; Martin, L.W.; Holcomb, M.B.; He, Q.; et al. Electric Modulation of Conduction in Multiferroic Ca-doped $BiFeO_3$ Films. *Nat. Mater.* **2009**, *8*, 485. [CrossRef]
11. Li, Y.; Fang, P.; Fan, X.; Pei, Y. NiO-based Memristor with Three Resistive Switching Modes. *Semicond. Sci. Technol.* **2020**, *35*, 55004. [CrossRef]
12. Li, Y.; Chu, J.; Duan, W.; Cai, G.; Fan, X.; Wang, X.; Wang, G.; Pei, Y. Analog and Digital Bipolar Resistive Switching in Solution-combustion-processed NiO Memristor. *ACS Appl. Mater. Interfaces* **2018**, *10*, 24598–24606. [CrossRef] [PubMed]
13. Driscoll, T.; Kim, H.T.; Chae, B.-G.; Di Ventra, M.; Basov, D.N. Phase-transition Driven Memristive System. *Appl. Phys. Lett.* **2009**, *95*, 43503. [CrossRef]
14. Zhang, L.; Chen, Z.; Yang, J.J.; Wysocki, B.; McDonald, N.; Chen, Y. A Compact Modeling of TiO_2-TiO_{2-x} Memristor. *Appl. Phys. Lett.* **2013**, *102*, 153503. [CrossRef]
15. Dongale, T.; Desai, N.D.; Khot, K.; Volos, C.K.; Bhosale, P.N.; Kamat, R.K. An Electronic Synapse Device Based on TiO_2 Thin Film Memristor. *J. Nanoelectron. Optoelectron.* **2018**, *13*, 68–75. [CrossRef]
16. Zhou, Z.; Yan, X.; Zhao, J.; Lu, C.; Ren, D.; Lu, N.; Wang, J.; Zhang, L.; Li, X.; Wang, H.; et al. Synapse Behavior Characterization and Physical Mechanism of a TiN/SiOx/p-Si Tunneling Memristor Device. *J. Mater. Chem. C* **2019**, *7*, 1561–1567. [CrossRef]
17. Ryu, J.H.; Mahata, C.; Kim, S. Long-term and Short-term Plasticity of Ta_2O_5/HfO_2 Memristor for Hardware Neuromorphic Application. *J. Alloys. Compd.* **2021**, *850*, 156675. [CrossRef]
18. Kim, T.H.; Nili, H.; Kim, M.H.; Min, K.K.; Park, B.-G.; Kim, H. Reset-voltage-dependent Precise Tuning Operation of TiOx/Al_2O_3 Memristive Crossbar Array. *Appl. Phys. Lett.* **2020**, *117*, 152103. [CrossRef]
19. Zhou, Y.; Liu, D.; Wang, J.; Cheng, Z.; Liu, L.; Yang, N.; Liu, Y.; Xia, T.; Liu, X.; Zhang, X. Black Phosphorus Based Multicolor Light-modulated Transparent Memristor with Enhanced Resistive Switching Performance. *ACS Appl. Mater. Interfaces* **2020**, *12*, 25108–25114. [CrossRef]
20. Yan, X.; Wang, X.; Xing, B.; Yu, Y.; Yao, J.; Niu, X.; Li, M.; Sha, J.; Wang, Y. Resistive Memory Based on Single-crystalline Black Phosphorus Flake/HfOx structure. *AIP Adv.* **2020**, *10*, 75013. [CrossRef]
21. Di, J.; Du, J.; Lin, Z.; Liu, S.; Ouyang, J.; Chang, J. Recent Advances in Resistive Random Access Memory Based on Lead Halide Perovskite. *InfoMat* **2021**, *3*, 293–315. [CrossRef]
22. Chen, S.; Lou, Z.; Chen, D.; Shen, G. An Artificial Flexible Visual Memory System Based on an UV-Motivated Memristor. *Adv. Mater.* **2018**, *30*, 1705400. [CrossRef]
23. Chen, J.; Li, J.; Li, Y.; Miao, X. Multiply accumulate operations in memristor crossbar arrays for analog computing. *J. Semicond.* **2021**, *42*, 13104. [CrossRef]
24. Jain, S.; Ranjan, A.; Roy, K.; Raghunathan, A. Computing in Memory with Spin-transfer Torque Magnetic RAM. *IEEE Trans. Very Large Scale Integr. Syst.* **2018**, *26*, 470–483. [CrossRef]
25. Kim, H.; Chen, Q.; Yoo, T.; Kim, T.T.-H.; Kim, B. A 1-16b Precision Reconfigurable Digital In-Memory Computing Macro Featuring Column-MAC Architecture and Bit-Serial Computation. In Proceedings of the ESSCIRC 2019—IEEE 45th European Solid State Circuits Conference (ESSCIRC), Cracow, Poland, 23–26 September 2019; pp. 345–348. [CrossRef]

26. Wang, J.; Wang, X.; Eckert, C.; Subramaniyan, A.; Das, R.; Blaauw, D.; Sylvester, D. A 28-nm Compute SRAM With Bit-Serial Logic/Arithmetic Operations for Programmable In-Memory Vector Computing. *IEEE J. Solid-State Circuits* **2019**, *55*, 76–86. [CrossRef]
27. Whitehead, A.N.; Russell, B. *Principia Mathematica*; Cambridge University Press: Cambridge, UK, 1912.
28. Borghetti, J.; Snider, G.S.; Kuekes, P.J.; Yang, J.J.; Stewart, D.R.; Williams, R.S. 'Memristive' Switches Enable 'Stateful' Logic Operations via Material Implication. *Nature* **2010**, *464*, 873–876. [CrossRef]
29. Bickerstaff, K.; Swartzlander, E.E. Memristor-based arithmetic. In Proceedings of the 2010 Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 7–10 November 2010.
30. Shaltoot, A.H.; Madian, A.H. Memristor based carry lookahead adder architectures. In Proceedings of the 2012 IEEE 55th International Midwest Symposium on Circuits and Systems, Boise, ID, USA, 5–8 August 2012.
31. Xie, L.; Nguyen, H.A.D.; Taouil, M.; Hamdioui, S.; Koen, B. A Mapping Methodology of Boolean Logic Circuits on Memristor Crossbar. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **2018**, *37*, 311–323. [CrossRef]
32. Zha, X.; Xia, Y. Defect-tolerant mapping of CMOL circuits with delay optimization. In Proceedings of the ACM Great Lakes Symposium on VLSI, Online, China, 7–9 September 2020.
33. Guckert, L.; Swartzlander, E.E. MAD Gates Memristor Logic Design Using Driver Circuitry. *IEEE Trans. Circuits Syst. II* **2017**, *64*, 171–175. [CrossRef]
34. Kvatinsky, S.; Satat, G.; Wald, N.; Friedman, E.G.; Kolodny, A.; Weiser, U.C. Memristor-Based Material Implication (IMPLY) Logic: Design Principles and Methodologies. *IEEE Trans. Very Large Scale Integr. Syst.* **2013**, *22*, 2054–2066. [CrossRef]
35. Chakraborty, A.; Rahaman, H. Implementation of combinational circuits via material implication using memristors. In Proceedings of the 2016 IEEE Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER), Mangalore, India, 14–16 August 2016; pp. 67–72. [CrossRef]
36. Xie, L.; Du Nguyen, H.; Yu, J.; Kaichouhi, A.; Taouil, M.; AlFailakawi, M.; Hamdioui, S. Scouting Logic: A Novel Memristor-Based Logic Design for Resistive Computing. In Proceedings of the 2017 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Bochum, Germany, 3–5 July 2017; pp. 176–181. [CrossRef]
37. Kvatinsky, S.; Belousov, D.; Liman, S.; Satat, G.; Wald, N.; Friedman, E.G.; Kolodny, A.; Weiser, U.C. MAGIC—Memristor-aided Logic. *IEEE Trans. Circuits Syst. II* **2014**, *61*, 895–899. [CrossRef]
38. Thangkhiew, P.L.; Gharpinde, R.; Datta, K. Efficient Mapping of Boolean Functions to Memristor Crossbar Using MAGIC NOR Gates. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2018**, *65*, 2466–2476. [CrossRef]
39. Kvatinsky, S.; Wald, N.; Satat, G.; Koldony, A.; Weiser, U.C.; Friedman, E.G. MRL—Memristor ratioed logic. In Proceedings of the International Workshop on Cellular Nanoscale Networks and their Applications, Turin, Italy, 29 August 2012.
40. Vinukollu, L.; Kesari, P. Area efficient carry look ahead adder based on memristor ratioed logic. In Proceedings of the 2021 2nd International Conference on Communication, Computing and Industry 4.0, Bangalore, India, 16–17 December 2021.
41. Wang, Z.; Wang, L.; Duan, S. Memristor Ratioed Logic Crossbar-based Delay and Jump-key Flip-flops Design. *Int. J. Circuit Theory Appl.* **2022**, *50*, 353–1364. [CrossRef]
42. Paramasivam, K.; Nithya, N.; Nepolean, A. A Novel Hybrid CMOS-Memristor Based 2-Bit Magnitude Comparator using Memristor Ratioed Logic Universal Gate for Low Power Applications. In Proceedings of the 2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), Coimbatore, India, 8–9 October 2021; pp. 1–5. [CrossRef]
43. Lageweg, C.; Cotofana, S.; Vassiliadis, S. A linear threshold gate implementation in single electron technology. In Proceedings of the IEEE Computer Society Workshop on VLSI 2001. Emerging Technologies for VLSI Systems, Orlando, FL, USA, 19–20 April 2001; p. 93. [CrossRef]
44. Bahrepour, D.; Sharifi, M.J. High Speed Full Adder Based on Modified Linear Threshold Gate and Its Application to a 4-2 Compressor. *J. Comput. Theor. Nanosci.* **2013**, *10*, 2527–2535. [CrossRef]
45. Xie, L. Hybrid threshold-Boolean logic mapped on memristor crossbar. In Proceedings of the 2016 12th Conference on Ph.D. Research in Microelectronics and Electronics (PRIME), Lisbon, Portugal, 25 July 2016; pp. 1–4. [CrossRef]
46. Rajendran, J.; Manem, H.; Karri, R.; Rose, G.S. An Energy-Efficient Memristive Threshold Logic Circuit. *IEEE Trans. Comput.* **2012**, *61*, 474–487. [CrossRef]
47. Danaboina, Y.K.Y.; Samanta, P.; Datta, K.; Chakrabarti, I.; Sengupta, I. Design and implementation of threshold logic functions using memristors. In Proceedings of the 2019 32nd International Conference on VLSI Design and 2019 18th International Conference on Embedded Systems (VLSID), New Delhi, India, 5–9 January 2019.
48. Traeff, H. Novel Approach to High Speed CMOS Current Comparators. *Electron. Lett.* **1992**, *28*, 310–312. [CrossRef]
49. Testa, E.; Soeken, M.; Zografos, O.; Soeken, M.; Zografos, O.; Amaru, L.; Raghavan, P.; Lauwereins, R.; Gaillardon, P.-E.; De Micheli, G. Inversion optimization in majority-inverter graphs. In Proceedings of the 2016 IEEE/ACM International Symposium on Nanoscale Architectures, Beijing, China, 18–20 July 2016.
50. Guckert, L.; Swartzlander, E. Optimized memristor-based ripple carry adders. In Proceedings of the Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, 6–9 November 2016.
51. Ahmad, K.; Abdalhossein, R. Novel Design for A Memristor-based Full Adder Using A New IMPLY Logic Approach. *J. Comput. Electron.* **2018**, *17*, 1303–1314.

52. Rohani, S.; Taherinejad, N.; Radakovits, D. A Semiparallel Full-adder in IMPLY Logic. *IEEE Trans. Very Large Scale Integr. Syst.* **2020**, *28*, 297–301. [CrossRef]
53. Siemon, A.; Menzel, S.; Waser, R.; Linn, E. A Complementary Resistive Switch-based Crossbar Array Adder. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2015**, *5*, 64–74. [CrossRef]
54. Talati, N.; Gupta, S.; Mane, P.; Kvatinsky, S. Logic Design Within Memristive Memories Using Memristor-Aided loGIC (MAGIC). *IEEE Trans. Nanotechnol.* **2016**, *15*, 635–650. [CrossRef]
55. Wang, Y. A Few-step and Low-cost Memristor Logic Based on MIG Logic for Frequent-off Instant-on Circuits in IoT Applica-tions. *IEEE Trans. Circuits Syst. II* **2019**, *66*, 662–666. [CrossRef]
56. Cui, X.; Ma, X.; Lin, Q.; Li, X.; Zhou, H.; Cui, X. Design of High-Speed Logic Circuits with Four-Step RRAM-Based Logic Gates. *Circuits Syst. Signal Process.* **2019**, *39*, 2822–2840. [CrossRef]
57. Maan, A.K.; James, A.P. Voltage controlled memristor threshold logic gates. In Proceedings of the 2016 IEEE Asia Pacific Conference on Circuits and Systems, Jeju, Republic of Korea, 25–28 October 2016.
58. Guckert, L.; Swartzlander, E. Optimized Memristor-based Multipliers. *IEEE Trans. Circuits Syst.* **2017**, *64*, 373–385. [CrossRef]
59. Haghiri, S.; Nemati, A.; Feizi, S.; Amirsoleimani, A.; Ahmadi, A.; Ahmadi, M. A memristor based binary multiplier. In Proceedings of the Canadian Conference on Electrical and Computer Engineering, Windsor, ON, Canada, 30 April—3 May 2017.
60. Radakovits, D.; Taherinejad, N.; Cai, M.; Mirabbasi, S. A Memristive Multiplier Using Semi-serial IMPLY-based Adder. *IEEE Trans. Circuits Syst. I* **2020**, *67*, 1495–1506. [CrossRef]
61. Lee, S.; Park, B.; Cho, S.; Cho, K.; Eshraghian, K. Memristor-CMOS reconfigurable multiplier architecture. In Proceedings of the 2014 14th International Workshop on Cellular Nanoscale Networks and their Applications (CNNA), Notre Dame, IN, USA, 29–31 July 2014.
62. Teimoory, M.; Amirsoleimani, A.; Ahmadi, A.; Ahmadi, M. A hybrid memristor-CMOS multiplier design based on memristive universal logic gates. In Proceedings of the 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), Boston, MA, USA, 6–9 August 2017.
63. Chen, Q.-L.; Liu, G.; Tang, M.-H.; Chen, X.-H.; Zhang, Y.-J.; Zheng, X.-J.; Li, R.-W. A univariate ternary logic and three-valued multiplier implemented in a nano-columnar crystalline zinc oxide memristor. *RSC Adv.* **2019**, *9*, 24595–24602. [CrossRef]
64. Liu, R.; Mahalanabis, D.; Barnaby, H.J.; Yu, S. Investigation of Single-Bit and Multiple-Bit Upsets in Oxide RRAM-Based 1T1R and Crossbar Memory Arrays. *IEEE Trans. Nucl. Sci.* **2015**, *62*, 2294–2301. [CrossRef]
65. Pal, S.; Bose, S.; Ki, W.-H.; Islam, A. Design of Power- and Variability-Aware Nonvolatile RRAM Cell Using Memristor as a Memory Element. *IEEE J. Electron Devices Soc.* **2019**, *7*, 701–709. [CrossRef]
66. Lastras-Montaño, M.A.; Cheng, K.-T. Resistive random-access memory based on ratioed memristors. *Nat. Electron.* **2018**, *1*, 466–472. [CrossRef]
67. Papandroulidakis, G.; Serb, A.; Khiat, A.; Merrett, G.V.; Prodromakis, T. Practical Implementation of Memristor-Based Threshold Logic Gates. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2019**, *66*, 3041–3051. [CrossRef]
68. Yang, L.; Wang, Y.; Wu, Z.; Wang, X. FPGA Implementation of Threshold-Type Binary Memristor and Its Application in Logic Circuit Design. *Micromachines* **2021**, *12*, 1344. [CrossRef]
69. Papandroulidakis, G.; Khiat, A.; Serb, A.; Stathopoulos, S.; Michalas, L.; Prodromakis, T. Metal oxide-enabled reconfigurable memristive threshold logic gates. In Proceedings of the IEEE International Symposium on Circuits and Systems, Florence, Italy, 27–30 May 2018.
70. Kuon, I.; Rose, J. Measuring the Gap Between FPGAs and ASICs. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **2007**, *26*, 203–215. [CrossRef]
71. Wang, W.; Jing, T.T.; Butcher, B. FPGA Based on Integration of Memristors and CMOS Devices. In Proceedings of the ISCAS 2010—2010 IEEE International Symposium on Circuits and Systems: Nano-Bio Circuit Fabrics and Systems, Paris, France, 30 May–2 June 2010.