MRL – Memristor Ratioed Logic

Shahar Kvatinsky, Nimrod Wald, Guy Satat,
Avinoam Kolodny, and Uri C. Weiser
Department of Electrical Engineering
Technion – Israel Institute of Technology
Haifa 32000 ISRAEL
{skva@tx, guys@tx, kolodny@ee,
uri.weiser@ee}.technion.ac.il

Eby G. Friedman
Department of Electrical and Computer Engineering
University of Rochester
Rochester, New York 14627 USA
friedman@ece.rochester.edu

***Abstract*— Memristive devices are novel structures, developed primarily as memory. Another interesting application for memristive devices is logic circuits. In this paper, MRL (Memristor Ratioed Logic) - a hybrid CMOS-memristive logic family - is described. In this logic family, OR and AND logic gates are based on memristive devices, and CMOS inverters are added to provide a complete logic structure and signal restoration. Unlike previously published memristive-based logic families, the MRL family is compatible with standard CMOS logic. A case study of an eight-bit full adder is presented and related design considerations are discussed.**

I. INTRODUCTION

Memristors [1] and memristive devices [2] are novel structures, useful in many applications. This device is basically a resistor with varying resistance, dependent on the history of the device. It can be used for memory, where the data is stored as a resistance. While memory is the common application for memristive devices, additional applications can also use memristive devices as building blocks, such as analog circuits, neuromorphic systems, and logic circuits.

This paper is focused on bipolar memristive devices [3], such as TiO_2 memristive devices and STT-MRAM (Spin Transfer Torque Magnetoresistance Random Access Memory). In bipolar memristive devices, the resistance of the device increases due to current flow in one direction, and decreases due to current flow in the other direction. The symbol and polarity of a memristive device are shown in Figure 1. Several memristive device models have been developed. In this paper, the TEAM model [4] is used since this model can fit any memristive device.

Practical memristive devices are nonvolatile and compatible with standard CMOS technology [5]. These devices are fabricated in the metal layers of an integrated circuit, where the memristive effects occur in the oxide between the metal layers (*e.g.*, in TiO_2) or within the metal layers (*e.g.*, in STT-MRAM). Memristive devices can therefore be fabricated above the CMOS transistors. The size of a typical memristive device is relatively small, since the fabrication process is similar to the processing of a via between metal layers. Hence,

Figure 1. Memristive device symbol. The thick black line on the left side of the device represents the polarity of the device. If the current flows into the device, the resistance of the device decreases. If the current flows out of the device, the resistance increases.

memristive-based circuits may be smaller than transistor-only CMOS circuits. Memristive devices therefore exhibit high density and good scalability. The read and write time for these devices can be as fast as one nanosecond [6]. Currently, except for STT-MRAM, memristive devices suffer from endurance limitations, where the number of allowed writes per cell is approximately 10^{10} [7]. It is believed however that this limit will increase to at least 10^{15} [8]. Memristive devices may therefore solve many major problems in the semiconductor industry, providing nonvolatile, dense, fast, and power efficient memory.

Integrating memristive devices and CMOS for performing logical operations may be beneficial. Since memristive devices are fabricated within the metal layers, the integration saves physical area and therefore increases the logic density. Furthermore, with deeply scaled CMOS, CMOS logic suffers from problems such as leakage current, requiring novel logic structures.

Logic operations with memristive devices open opportunities for novel functionality. Although the use of memristive devices as logic gates is early, several approaches have been proposed, mainly for logic gates designed within the structure of a crossbar array originally targeted for memory [9]. Logic in a memristive-based crossbar opens an opportunity to explore advanced computer architectures different from the classical Von Neumann architecture. In these architectures, the memory can perform logic operations on the same devices that store data. The decision regarding which elements act as logic gates and which act as memory

This work was partially supported by Hasso Plattner Institute and by an Intel grant, "Heterogeneous Computing, the Inevitable Solution: Power Management, Scheduling and ISA," Grant no. 864-737-13.

cells can be done dynamically during the operation of the memory [10].

Material implication (IMPLY logic gate) [11] is one option for logic inside a memristive-based crossbar. The IMPLY logic gate is extended in [12] to a NOR logic gate. Another logic family within a crossbar is MAGIC [13]. In MAGIC, all basic Boolean functions can be produced, *e.g.*, AND, NAND, NOR, and OR logic gates. All of these logic gates require a sequencer to operate the logic gate, *i.e.*, any basic Boolean function requires more than one clock cycle to execute the computation. The logic within a crossbar is therefore relatively slow. These logic gates also suffer from state drift and lack signal restoration [14].

Memristive-based logic families within a crossbar cannot be easily integrated with standard CMOS logic. In these logic families, a resistance, rather than a voltage, represents the logical state. To integrate memristive devices with CMOS for logic circuits, several requirements need to be fulfilled: the technology of the memristive devices needs to be compatible with a standard CMOS process, the logical state, used for input and output signal transfer between the logic gates, needs to be converted from a resistance into a voltage, and the interface between the memristive device layers and the CMOS layer should require minimal additional circuitry. To integrate these logic families with standard voltage based CMOS logic, a conversion mechanism is required. This mechanism includes a sense amplifier as well as additional components. The additional required circuitry reduces the efficiency of integrating CMOS and memristive-based logic families within a memory [10].

In this paper, MRL (Memristor Ratioed Logic) for integration with CMOS is described. This logic family uses the programmable resistance of memristive devices for computation of Boolean AND/OR functions with voltage as the state variable, hence it avoids the drawbacks described above. Design principles and constraints of this logic family are discussed in Section II. A case study of an eight-bit full adder is used to demonstrate the MRL design process in Section III. The dependence of the MRL gates on the behavior of the memristive device, as well as several tradeoffs in the design procedure is discussed in Sections III and IV.

II. MEMRISTOR RATIOED LOGIC (MRL) FAMILY

An interesting method for integrating memristive devices with standard CMOS logic is using memristive devices as computational elements, OR and AND logic gates [15]. Since these functions are non-inverting logic gates, a complete logic structure can be achieved by adding a standard CMOS inverter. In this logic family, the logical is represented as a voltage, consistent with CMOS. The memristive devices are utilized solely for logic computation and not for storing a logical state. The computational result is independent of the initial state of the memristive devices, and the initial state only affects the computational time. Unlike other logic methods (such as IMPLY), the computational process is composed of only a single step. Similar to standard combinatorial logic using CMOS, the topology of the circuit determines the logical function.

A. Description of Logic Gates

Both OR and AND logic gates consist of two memristive devices connected in series with opposite polarity, as shown in Figures 2a and 2b. The output node is the common node of the memristive devices, while the signals on the other terminal of each memristive device are the inputs of the logic gate.

Due to the polarity of the memristive devices, in an OR logic gate, when current flows into the logic gate through one of the inputs, the resistance of this memristive device decreases. Similarly, in an AND logic gate, the opposite polarity is used, and the resistance of the memristive device increases when current flows into the device.

Both the OR and AND logic gates react similarly to identical inputs (where either both inputs are logical 1 or both are logical 0). For identical inputs, the voltage drop between inputs is zero; hence no current flows within the circuit. The output voltage is therefore equal to the input voltage. For the case where both inputs are logical zero (one), the ground (supply) voltage is at the inputs, the output voltage is ground (supply) and the logical state of the output is logical zero (one).

For the case where the inputs are different, *i.e.*, one input is logical one and the other input is logical zero, current flows from the high voltage (the terminal of the memristive device where the input is logical one) to the low voltage (the terminal of the memristive device where the input is logical zero), thus changing the resistance of both memristive devices. This case for an OR logic gate is illustrated in Figure 2c. The resistance of the memristive device connected to the logical one input R_1 is lower, and the resistance of the memristive device R_2 is higher, as shown in Figure 2e. At the end of the computational process, the resistance of both memristive devices is approximately R_{ON} and R_{OFF}, respectively, the minimum and maximum resistance of the device. Assuming $R_{OFF} >> R_{ON}$, the output voltage of the logic gate is determined by the voltage divider across both of the memristive devices,

$$V_{out,OR} = \frac{R_{off}}{R_{off} + R_{on}} V_{high} \approx V_{high}. \quad (1)$$

In the AND logic gate, the opposite polarity, as compared to the OR logic gate, is used. For the case where the inputs are different, the resistance of the memristive devices is the opposite of the resistance of the OR logic gate. This behavior is illustrated in Figures 2d and 2f. The output voltage of the AND logic gate in this case is therefore

$$V_{out,AND} = \frac{R_{on}}{R_{off} + R_{on}} V_{high} \approx 0. \quad (2)$$

Note that the initial resistance of both memristive devices has no effect on the result of the computation. The only effect

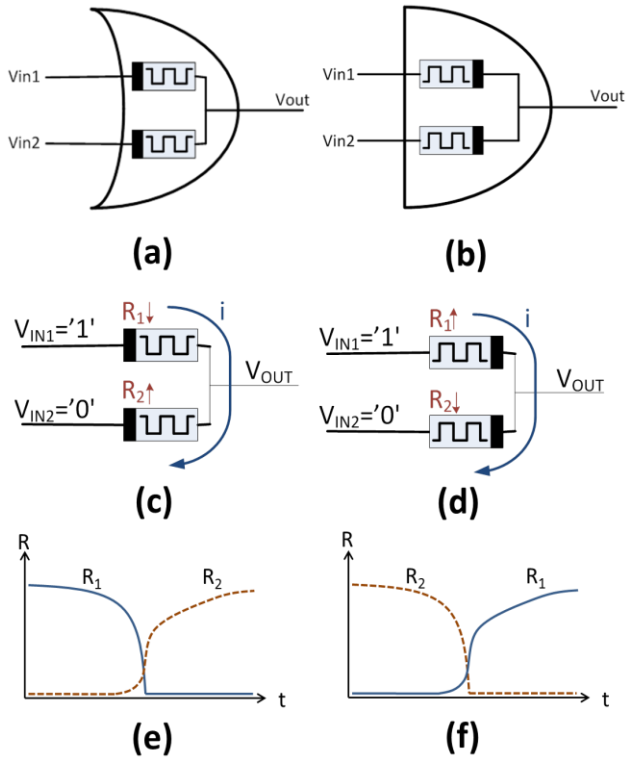

Figure 2. Schematic and behavior of MRL gates. (a) The schematic of an OR logic gate, and (b) an AND logic gate. Both logic gates consists of two memristive devices where the polarity of the memristive devices is the only structural difference. The behavior of (c) an OR logic gate, and (d) an AND logic gate when V_{IN1} = '1' and V_{IN2} = '0'. The current flows from V_{IN1} to V_{IN2} and the resistance of the memristive devices changes for the (e) OR, and (f) AND logic gates. The continuous and dashed lines are, respectively, the resistance of R_1 and R_2.

of the initial resistance on the behavior of the logic gate is the delay time of the execution for the case where the inputs are different, *i.e.*, the time required to change the resistance of both memristive devices to either the maximum or minimum resistance. The delay time is also dependent on the voltage level. A relatively low voltage level increases the delay time. It is possible that the memristive devices do not fully switch and achieve the maximum and minimum resistance since the input voltages are not applied for a sufficiently long time or the input voltage is too low. In this case, it would be difficult to distinguish between the different output levels. The MRL family is inspired by Diode Logic [16] and shares some characteristics, such as both logic families are non-inverting and non-restoring [17]. The number of inputs for both MRL gates can be extended in a similar way as diode logic, as shown in Figures 3a and 3b.

To provide a complete logic family, an inverter is needed in addition to OR and AND logic gates. Furthermore, memristive devices are passive elements and therefore cannot amplify signals. The MRL OR and AND logic gates therefore lack signal restoration, *i.e.*, the output voltage levels degrade, as expressed by (1) and (2). These logic gates cannot therefore be cascaded for too many stages before signal amplification is required. CMOS logic, alternatively, exhibits signal restoration. Since the logical state of the input and output in MRL OR and AND logic gates is represented as a voltage, these logic gates can be integrated with standard CMOS inverters. To provide a complete logic structure and signal restoration, the addition of a CMOS inverter to the MRL family is therefore proposed. The schematic of a two input MRL NAND and NOR is shown in Figures 3c and 3d.

B. General design considerations

In the design process of an MRL gate, several issues need to be considered. When the input changes from one input case to another input case, *i.e.*, changing the inputs from (0,1) to (1,0) and vice versa, the output produces a dynamic hazard until the switching process is completed. Another issue may occur when both initial resistances are high (approximately R_{OFF}). In this case, the current through the logic gate is relatively small, and the settling time is therefore relatively long, also producing a dynamic hazard. The dynamic behavior of the OR and AND logic gates is illustrated in Figures 4a and 4b.

Power consumption is another issue. When both inputs are identical, no current flows in the circuit and the power is zero. If the inputs are different, current flows and power is consumed. The power consumed during the switching of the memristive devices is dependent on the resistance of both memristive devices and changes during the computational process. Generally, the power consumption of an MRL gate for these input cases is

$$P(t) = \frac{V_{high}^{\ 2}}{R_1(t) + R_2(t)}, \tag{3}$$

where V_{high} is the voltage of logical one and is assumed to be constant, and $R_1(t)$ and $R_2(t)$ are the resistance of the memristive devices, which change during the computational process. The value of $R_1(t)$ and $R_2(t)$ is dependent on the initial states and the value of V_{high}. For the case of different inputs, a constant current flows from one input to the other input, even after the resistance of the memristive devices reaches the desired magnitude and the output becomes stable. The static power consumed in these cases is approximately

$$P_{static} = \frac{V_{high}^{\ 2}}{R_{on} + R_{off}}. \tag{4}$$

The power consumption for all input cases is illustrated in Figure 4c. The output voltage is dependent on the voltage divider across the two memristive devices. This voltage divider degrades the output signal. Although the degradation is minor when $R_{OFF} >> R_{ON}$, for cascaded logic gates, this degradation accumulates and may become significant. This phenomenon can be avoided by occasionally amplifying the signal by CMOS inverters or buffers. Integrating a CMOS inverter into an MRL OR or AND logic gate however adds capacitance to the circuit. The delay time of the logic gates is dependent on the CMOS gate capacitance and therefore needs to be optimized. The delay of the logic gates is the time required for the memristive devices to be fully switched, and

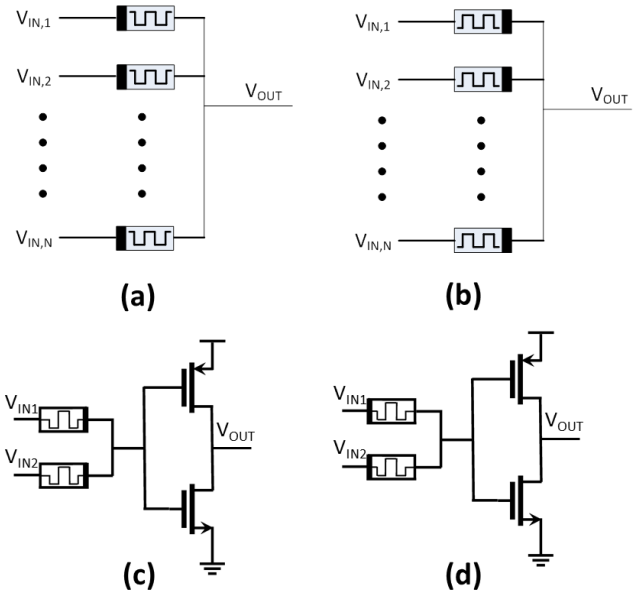

Figure 3. Schematic of an (a) N-input MRL OR, (b) N-input MRL AND, (c) two-input MRL NAND, and (d) two-input MRL NOR.

is dependent on the determined by the case of different inputs.

The MRL logic gates can be inserted into a standard cell library as in standard CMOS logic. These standard cell libraries can consist of OR and AND logic gates. Alternatively, NOR and NAND logic gates, consisting of a memristive-based OR (AND), and a CMOS inverter, can produce the functionality of a NOR (NAND) logic gate.

III. EIGHT-BIT FULL ADDER CASE STUDY

An eight-bit full adder is considered as a case study for the MRL family. Five different parameter sets of memristive devices are chosen to evaluate a variety of memristive devices. The primary parameters are the linearity coefficient and the current threshold (respectively, α, i_{on}, and i_{off} in the TEAM model [4]). All other parameters are chosen to exhibit a hysteretic behavior. The parameters for the memristive devices are listed in Table 1.

To provide a standard cell design methodology, the standard cell is a NAND (NOR) logic gate, as described in Section IIB. No current flows from the output node in steady state since the output node of the AND (OR) logic gate is connected to an MOS gate. In this approach, every standard cell requires two connections between the CMOS and memristive layers, one for the middle stage transition and one for the output. This approach is robust, albeit inefficient in terms of power consumption and area as compared to an optimized circuit, where the CMOS inverter is only applied when signal restoration is needed or when the logic function requires signal inversion. In this case study, the optimized approach is used.

For the optimized approach, when connecting cascaded memristive-based MRL gates, current can flow from the output node into the input of the next logic gate. In this case, the current flowing through the two memristive devices of one gate is not equal, and the smaller current may drop below the current threshold of the memristive devices, causing the logic gate to partially switch. This phenomenon can degrade the output voltage, and may perhaps cause the logic to fail after a single logic stage.

TABLE 1. DIFFERENT PARAMETERS OF THE MEMRISTIVE DEVICES USED IN THE CASE STUDY

Device / Parameter	Linear with no current thresh-old	Linear with current thresh-old	Low non-linearity	Non-linear	Highly non-linear
Parameter set number	1	2	3	4	5
α	1	1	3	5	10
i_{on}	-100 fA	-20 µA	-5 µA	-5 µA	-10 µA
i_{off}	100 fA	20 µA	5 µA	5 µA	10 µA
k_{on}	$-5\cdot 10^{-8}$	-10	-0.1	-0.01	-0.001
k_{off}	$5\cdot 10^{-8}$	10	0.1	0.01	0.001
R_{on}	1 kΩ				
R_{off}	100 kΩ				

One approach to eliminate a possible logic failure is to increase the voltage of the high logical state to ensure that all currents in the circuit are greater than the current threshold of the devices. The increase in voltage is limited by the CMOS process, since high voltages may cause breakdown in the CMOS transistors (*e.g.*, gate induced drain leakage [18]), and also dissipate more power.

Another approach to eliminate logic failure is to amplify the signal with CMOS logic gates, preventing steady state current leakage and performing signal restoration. In this case study, both approaches are used. The voltage is increased and signal restoration is achieved through a CMOS inverter. The behavior of an MRL XOR logic gate is shown in Figure 5 to demonstrate the signal degradation. Note that these signal degradation issues are circuit dependent, *i.e.*, the degree of signal degradation is dependent on the logic circuit structure as well as the parameters of the memristive devices. A schematic of the one-bit full adder used in this case study is shown in Figure 6.

The design of the eight-bit full adder in this case study is achieved using eight cascaded one-bit full adders. A tradeoff between signal integrity and minimizing the number of vias is the primary issue. To maintain a distinct value for the output of the eight-bit full adder (S_i for i = 1, …, 8 and C_{OUT}), a set of CMOS buffers is added to the circuit to amplify the output signal. For the intermediate signals ($C_{OUT} \rightarrow C_{IN}$), no constraint is placed on the strength of the signal other than to maintain the correct logical polarity. A lower signal strength requires fewer CMOS gates and hence less area and power consumption. The required number of CMOS buffers is dependent on the signal degradation along the logic path.

For parameter sets 1, 3 and 4 (memristive devices with a relatively low current threshold), the one-bit full adder shown

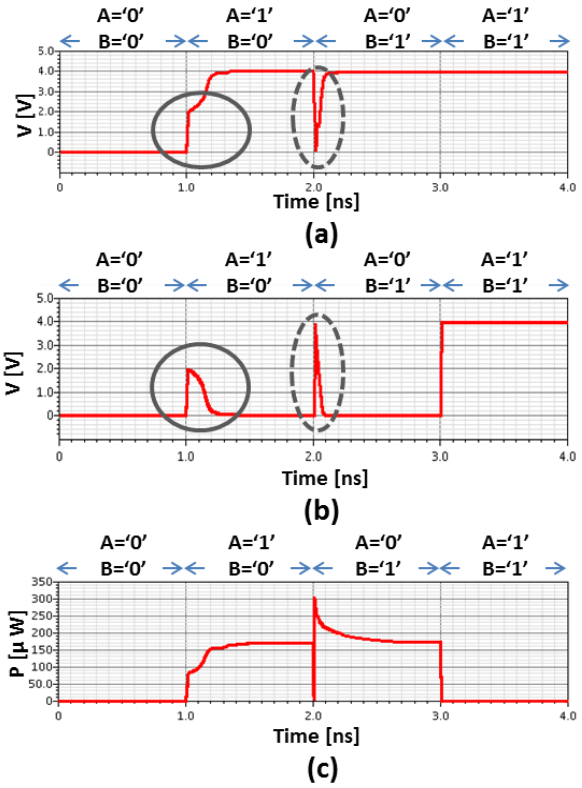

Figure 4. Dynamic behavior of MRL gates. Waveforms of (a) an OR logic gate, and (b) an AND logic gate. The output voltage is shown for different input states. Dynamic hazards occur when the input changes to ('0', '1') or ('1', '0'), which is marked by an oval. (c) The power consumption for both logic gates is identical. For the cases where the input states are different ('0', '1') or ('1', '0'), static power is consumed after the output is stable.

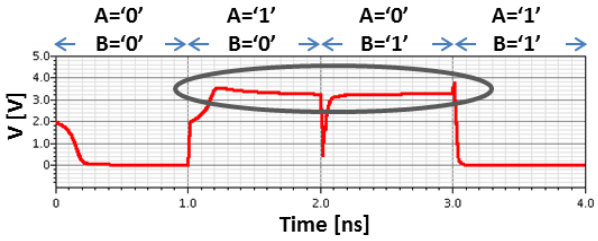

Figure 5. Dynamic behavior of an MRL XOR logic gate. The high voltage is 4 Volts. The output voltage degrades by approximately 15% for the input cases of ('1', '0') and ('0', '1').

in Figure 6 exhibits correct logic functionality, which requires amplifying the signal between different bit levels. Parameter sets 2 and 5 demonstrate a high current threshold and are therefore more sensitive to signal degradation due to partial switching. For these parameter sets, the circuit fails for all CMOS compatible voltages. For parameter sets 2 and 5, buffers have been added to the one-bit full adder circuit to ensure correct logic behavior. The required voltage levels and number of buffers for each parameter set are listed in Table 2, total number of devices is listed in Table 3, and normalized power consumption[1] for each parameter set is listed in Table 4.

[1] The power is normalized since the parameter set of the memristive devices is not correlated to the CMOS process.

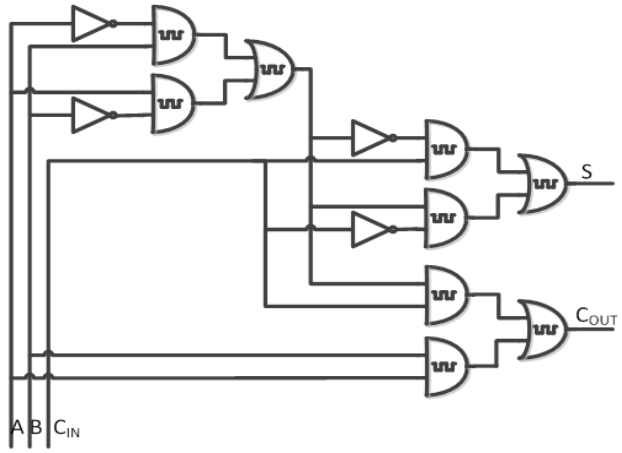

Figure 6. Schematic of an MRL one bit full adder (S = XOR[A, B, C_{IN}], $C_{OUT} = A \cdot B + C_{IN} \cdot$XOR[$A$, B]) for the optimized method used in the case study. The one-bit full adder consists of six memristive-based OR logic gates, three memristive-based AND logic gates, and four CMOS inverters. In this circuit, 18 memristive devices and eight CMOS transistors are used.

Note from the data listed in Tables 3 and 4 that unlike most digital applications [4], a linear memristive device with no threshold (as in parameter set no. 1) is preferable to minimize the number of connections between the CMOS and memristive layers, and to reduce power. The optimized approach consumes less dynamic power but more total energy, as compared to a standard cell methodology, since the static power is non-zero. Since decreasing the operating voltage requires additional CMOS buffers, the number of CMOS buffers in parameter set no. 3 (a high voltage of 3 Volts) is lower than in parameter set no. 1. The high voltage used in parameter sets number 2 and 5 significantly increases the power consumption.

IV. CONCLUSIONS

Memristor Ratioed Logic (MRL), a hybrid CMOS-memristive logic family, is described in this paper. This logic family uses less die area as compared to CMOS logic. It is possible to reduce the design effort of an MRL circuit by using standard library cells composed of only NOR and NAND logic gates. Standard cells however limit the flexibility of the design process and restrict the opportunity to save area. Other optimization criteria are also possible, such as increasing the operating voltage and minimizing the number of connections between the CMOS and memristive layers.

An eight-bit full adder is presented as a case study. This full adder is optimized for minimum CMOS/memristive connections and saves approximately 50% in area as compared to CMOS logic, while requiring 44% fewer connections and 30% less power as compared to an MRL standard cell library.

It is also shown that a linear memristive device with no current threshold is preferable for the MRL logic family, unlike other digital applications, where a threshold and nonlinearity are desirable. MRL gates based on linear memristive devices are faster, smaller, and consume less power as compared to nonlinear memristive devices.

The Memristor Ratioed Logic family opens an opportunity for additional memristive/CMOS integrated circuits and increases logic density. This enhancement can provide greater computational abilities to processor and other computational circuits.

TABLE 2. VOLTAGE LEVEL AND NUMBER OF BUFFERS FOR EACH PARAMETER SET IN THE CASE STUDY

Parameter set	Supply voltage	Number of buffers needed			
		Inside each one bit full adder	Between each C_{OUT} and C_{IN}	After last stage C_{OUT}	After each S_i
1	1V	0	2	2	1
2	6.5V	2	1	2	2
3	3V	0	1	1	1
4	4V	0	2	2	1
5	6.5V	2	2	2	2

TABLE 3. SUMMARY OF CASE STUDY

Parameter set	Number of memristors	Number of CMOS transistors	Number of vias	Supply voltage
CMOS – based	-	288	-	1 V
Standard cell approach	144	144	144	1 V
1	144	160	80	1 V
2	144	228	96	6.5 V
3	144	128	80	3 V
4	144	160	80	4 V
5	144	256	96	6.5 V

TABLE 4. POWER CONSUMPTION AND ENERGY FOR CASE STUDY

Parameter set	Average power [normalized]	Total energy [normalized]
Standard cell approach (for parameter set 1)	1	1
1	0.72	5.02
2	386.4	2035.1
3	22.5	167.9
4	60.8	499.2
5	354.95	2004.5

REFERENCES

[1] L. O. Chua, "Memristor – The Missing Circuit Element," *IEEE Transactions on Circuit Theory*, Vol. 18, No. 5, pp. 507-519, September 1971.

[2] L. O. Chua and S. M. Kang, "Memristive Devices and Systems," *Proceedings of the IEEE*, Vol. 64, No. 2, pp. 209- 223, February 1976.

[3] P. Vontobel, W. Robinett, J. Straznicky, P. J. Kuekes, and R. S. Williams, "Writing to and Reading from a Nano-Scale Crossbar Memory Based on Memristors," *Nanotechnology*, Vol. 20, No. 42, pp. 1-21, October 2009.

[4] S. Kvatinsky, E. G. Friedman, A. Kolodny, and U. C. Weiser, "TEAM - ThrEshold Adaptive Memristor Model," *IEEE Transactions on Circuits and Systems I: Regular Papers*, April 2012 (in press).

[5] J. Borghetti, Z. Li, J. Strasnicky, X. Li, D. A. A. Ohlberg, W. Wu, D. R. Stewart, and R. S. Williams, "A Hybrid Nanomemristor/Transistor Logic Circuit Capable of Self-Programming," *Proceedings of the National Academy of Sciences*, Vol. 106, No. 6, pp. 1699-1703, February, 2009.

[6] A. C. Torrezan, J. P. Strachan, G. Medeiros-Ribeiro, and R. S. Williams, "Sub-Nanosecond Switching of a Tantalum Oxide Memristor," *Nanotechnology*, Vol. 22, No. 48, pp. 1-7, November 2011.

[7] J. J. Yang *et al*, "High Switching Endurance in TaOx Memristive Devices," *Applied Physics Letters*, Vol. 97, No. 23, pp. 1-3, December 2010.

[8] J. Nickel, "Memristor Materials Engineering: From Flash Replacement towards a Universal Memory," *Proceedings of the IEEE IEDM Advanced Memory Technology Workshop*, December 2011.

[9] G. Snider, "Computing with Hysteretic Resistor Crossbars," *Applied Physics A*, Vol. 80, No. 6, pp. 1165-1172, March 2005.

[10] S. Kvatinsky, E. G. Friedman, A. Kolodny, and U. C. Weiser, "Design Principles and Methodologies for Integrated Memristor Memory and Memristor Logic," unpublished.

[11] 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, pp. 873-876, April 2010.

[12] S. Shin, K. Kim, and S.-M. Kang, "Reconfigurable Stateful NOR Gate for Large-Scale Logic-Array Integrations," *IEEE Transactions on Circuits and Systems II: Express Briefs*, Vol. 58, No. 7, pp. 442-446, July 2011.

[13] S. Kvatinsky *et al*, "MAGIC – Memristor Aided LoGIC," unpublished.

[14] S. Kvatinsky, E. G. Friedman, A. Kolodny, and U. C. Weiser, "Memristor-based IMPLY Logic Design Procedure," *Proceedings of the IEEE International Conference on Computer Design*, pp. 142-147, October 2011.

[15] K. Eshraghian, course notes on "Memristive Circuits and Systems," Technion, June 2011.

[16] R. H. Wilkinson, "A Method of Generating Functions of Several Variables Using Analog Diode Logic," *IEEE Transactions on Electronic Computers*, Vol. EC-12, No. 2, pp. 112-129, April 1963.

[17] G. G. Langdon Jr., *Logic Design – A Review of Theory and Practice*, Academic Press, 1974.

[18] T. Y. Chan, J. Chen, P. K. Ko, and C. Hu, "The Impact of Gate-Induced Drain Leakage Current on MOSFET Scaling," *Proceedings of the IEEE International Electron Devices Meeting*, pp. 718-721, December 1987.