Multistate Register Based on Resistive RAM

Ravi Patel, *Student Member, IEEE*, Shahar Kvatinsky, *Student Member, IEEE*,
Eby G. Friedman, *Fellow, IEEE*, and Avinoam Kolodny, *Senior Member, IEEE*

***Abstract*— In recent years, memristive technologies, such as resistive random access memory (RRAM), have emerged. These technologies are usually considered as alternates for static RAM, dynamic RAM, and Flash. In this paper, a novel digital circuit, the multistate register, is proposed. The multistate register is different from conventional types of memory, and is used to store multiple data bits, where only a single bit is active and the remaining data bits are idle. The active bit is stored within a CMOS flip flop, while the idle bits are stored in an RRAM crossbar co-located with the flip flop. It is demonstrated that additional states require an area overhead of 1.4% per state for a 64-state register. The use of multistate registers as pipeline registers is demonstrated for a novel multithreading architecture—continuous flow multithreading (CFMT), where the total area overhead in the CPU pipeline is only 2.5% for 16 threads compared with a single thread CMOS pipeline. The use of multistate registers in the CFMT microarchitecture enables higher performance processors (40% average performance improvement) with relatively low energy (6.5% average energy reduction) and area overhead.**

***Index Terms*— Flip flop, memristive device, memristor, multithreading, resistive random access memory (RRAM).**

I. Introduction

MEMRISTIVE technologies [1]–[3] have been proposed to augment existing state-of-the-art CMOS circuits. One interesting memristive technology is resistive random access memory (RRAM) [5]–[9]. RRAM-based memories can be integrated with existing digital circuits to increase functionality and system throughput. RRAM is a two-terminal device that exhibits the properties of nonvolatility and high density. Unlike charge-based memories, information in an RRAM is stored by modulating the material state. An RRAM cell dissipates no static power to store a state and provides immunity to radiation- and noise-induced soft errors. Fabrication of these devices generally requires deposition of a thin-film material. The integration of these devices with CMOS is constrained primarily by lithographic patterning limits. Thus memristors scale with existing CMOS technologies.

The traditional approach of increasing CPU clock frequency has abated owing to constraints on power consumption and density. To increase performance with each CMOS generation, thread-level parallelism is exploited with multicore processors [10]. This approach uses an increasing number of CMOS transistors to support additional cores on the same die, rather than increase the frequency of a single processor. This larger number of cores, however, dissipates greater static power. Multithreading is an approach to enhance the performance of an individual core by increasing logic utilization [11], without consuming additional static power. Handling each thread, however, requires duplication of resources (e.g., register files, flags, pipeline registers). This added overhead increases the area, power, and complexity of the processor, potentially increasing on-chip signal delays. The thread count is therefore typically limited to two to four threads per core in modern general purpose processors [12].

The high density, nonvolatility, and soft error immunity exhibited by resistive random access memory enables novel tradeoffs in digital circuits, allowing new mechanisms to increase thread count without increasing the static power. These tradeoffs support innovative memory structures for novel microarchitectures. In this paper, a memristive multistate pipeline register (MPR) is proposed that exploits these properties to enable higher throughput computing. The MPR is compatible with existing digital circuits while leveraging RRAM devices to store multiple machine states within a single register. This behavior enables an individual logic pipeline to be densely integrated with memory while retaining state information for multiple independent ongoing operations. The state information for each operation is stored within a local memory and recalled at a later time, allowing the computation to resume without flushing the pipeline.

This functionality is useful in multithreaded processors to store the state of different threads. This situation is demonstrated in the case study of a novel microarchitecture—continuous flow multithreading (CFMT) [13]. It is shown that including an RRAM MPR within the CFMT microarchitecture enhances the performance of a processor, on average, by 40%, while reducing the energy, on average, by 6.5%. The proposed MPR circuit can also be used as a multistate register for applications other than pipeline registers.

Background of RRAM and crosspoint style memories is reviewed in Section II. The operation of the multistate register is presented in Section III. The simulation setup and circuit

Manuscript received December 17, 2013; revised May 6, 2014; accepted August 3, 2014. Date of publication September 18, 2014; date of current version August 21, 2015. This work was supported in part by Hasso Plattner Institute, Potsdam, Germany; in part by the Advanced Circuit Research Center, Technion-Israel Institute of Technology, Haifa, Isreal; in part by the Binational Science Foundation under Grant 2012139; in part by the National Science Foundation under Grant CCF-1329314; and in part by Research Grants through the Qualcomm Inc., San Diego, CA, USA, Cisco Systems Inc., San Jose, CA, USA, and Samsung, Ridgefield Park, NJ, USA.

R. Patel and E. G. Friedman are with the Department of Electrical Engineering and Computer Engineering, University of Rochester, Rochester, NY 14627 USA (e-mail: rapatel0@gmail.com; friedman@ece.rochester.edu).

S. Kvatinsky and A. Kolodny are with the Department of Electrical Engineering, Technion-Israel Institute of Technology, Haifa 32000, Israel (e-mail: skva@tx.technion.ac.il; kolodny@ee.technion.ac.il).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVLSI.2014.2347926

evaluation process are described in Section IV. A case study examining the multistate register as a pipeline register within a CPU is presented in Section V, followed by some concluding remarks in Section VI.

II. Background

Memristors and memristor-based arrays behave differently from standard CMOS SRAM memory arrays owing to the different properties of RRAM devices. Operation of memristive devices and memristor-based crosspoint structures is described in the following section.

A. Background of Memristor and RRAM

Memristors [14] and memristive devices [15] behave as resistors, where the resistance is modulated by an applied bias. Positive and negative biases increase or decrease, respectively, the resistance of the device. In general, a bias applied for a longer duration produces a greater change in resistance. A larger voltage will generally increase the speed of the change in resistance. The device may also exhibit a threshold voltage or current, such that the resistance will change only if the bias exceeds the threshold specific to the device technology [16]–[18]. Once the bias is removed, the final resistance of the memristor is retained without dissipating any power.

One interesting memristive technology is RRAM, where oxide-based materials (e.g., TaO, TiO, SiO) [19], [20] rely on the migration of dopants to switch the resistance of a tunnel barrier. Dopant chains form through the oxide and reduce the thickness of the tunneling gap. An increase in the gap thickness gives rise to an increase in the resistance of the device whereas a decrease reduces the resistance. Currently, RRAM is considered a good candidate to replace Flash memory and is being widely investigated both in industry and academia.

The exact behavior of RRAM devices varies for different oxide materials. To simulate the behavior of memristive circuits, a general device model is used—the threshold adaptive memristor TEAM model [20]. In the TEAM model, the behavior of the resistive device is represented by the following expressions:

$$\frac{dx(t)}{dt} = \begin{cases} k_{\mathrm{OFF}} \cdot \left(\frac{i(t)}{i_{\mathrm{OFF}}} - 1\right)^{\alpha_{\mathrm{OFF}}} \cdot f_{\mathrm{OFF}}(x), & 0 < i_{\mathrm{OFF}} < i \quad (1a) \\ 0, & i_{\mathrm{ON}} < i < i_{\mathrm{OFF}} \quad (1b) \\ k_{\mathrm{ON}} \cdot \left(\frac{i(t)}{i_{\mathrm{ON}}} - 1\right)^{\alpha_{\mathrm{ON}}} \cdot f_{\mathrm{ON}}(x), & i < i_{\mathrm{ON}} < 0 \quad (1c) \end{cases}$$

$$v(t) = \left[R_{\mathrm{ON}} + \frac{R_{\mathrm{OFF}} - R_{\mathrm{ON}}}{x_{\mathrm{OFF}} - x_{\mathrm{ON}}}(x - x_{\mathrm{ON}}) \right] \cdot i(t) \quad (2)$$

where k_{OFF} and k_{ON} are fitting parameters, α_{ON} and α_{OFF} are adaptive nonlinearity parameters, i_{OFF} and i_{ON} are current threshold parameters, $f_{\mathrm{ON}}(x)$ and $f_{\mathrm{OFF}}(x)$ are window functions, R_{ON} and R_{OFF} are, respectively, the minimum and maximum resistance of the memristor, and x_{ON} and x_{OFF} are, respectively, the minimum and maximum allowed value of the internal state variable x. The window function returns a value between zero and one and describes the rate at which the change of the state variable becomes nonlinear near the minimum and maximum resistance of a memristor. A Joglekar window function is used with a p-coefficient of two [22]. An I–V curve of a memristive device based on the TEAM model is shown in Fig. 2(a), exhibiting a pinched hysteresis loop.

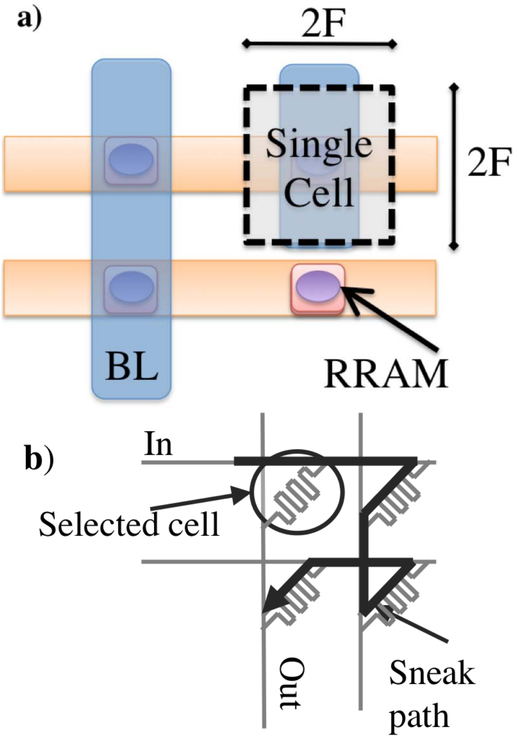

Fig. 1. RRAM crosspoint (a) structure and (b) example of a parasitic sneak path within a 2 × 2 crosspoint array.

B. Crosspoints and Nonlinearity

RRAM exhibits high density when used in a crosspoint array configuration. In this structure, a thin film is sandwiched between two sets of parallel interconnects. Each set of interconnects is orthogonal, allowing any individual memristive device to be selected by biasing one vertical and one horizontal metal line. In this configuration, the circuit density is only limited by the available metal pitch. The structure of a crosspoint is shown in Fig. 1(a).

Crosspoint arrays have the inherent problem of sneak path currents where currents propagate between two selected lines through unselected memristors. The sneak path phenomenon is shown in Fig. 1(b). The nonlinear I–V characteristic of certain memristive devices lessens the sneak path phenomenon [23]. This nonlinearity can be achieved by depositing additional materials above or below the memristive thin film. Depending on the material system used for RRAM, the nonlinearity can result from an insulator to metal transition or a negative differential resistance [23]. From a circuits perspective, the combined device can be modeled as a pair of cross-coupled diodes in series with a memristor, as shown in the inset of Fig. 2(b). Because the rectifying structure requires an

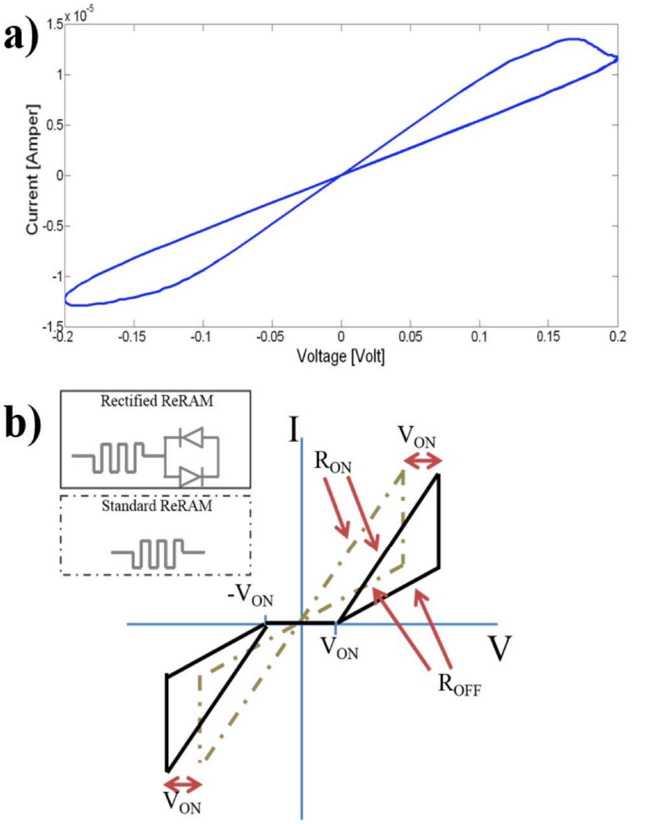

Fig. 2. I–V characteristic of a memristor for (a) TEAM model with a 0.2 V sinusoidal input operating at a frequency of 2 GHz and (b) resistive devices with and without ideal cross-coupled diodes. The parameters of the TEAM models are listed in Table II. V_{ON} is the ON-voltage of the diode, and R_{ON} and R_{OFF} are, respectively, the minimum and maximum resistance of the memristor.

TABLE I
COMPARISON OF DC ON/OFF CURRENT FOR 4 × 4 CROSSPOINT ARRAY

	I_{on} [mA]	I_{off} [mA]	Ratio	Average Active Power [mW]
Unrectified	2.3	0.132	1.7	1.45
Rectified	0.486	0.017	28.5	0.201

additional thin-film layer, there is no effect on the area of the crosspoint structure.

An I–V curve of a memristive device with cross-coupled diodes is shown in Fig. 2(b). The high resistance of the unselected devices reduces sneak currents and ensures that the leakage power of the array is relatively small. Reducing sneak currents ensures that the leakage power of the array is relatively small. A dc analysis of the on and off crosspoint currents is listed in Table I, where a 4 × 4 crosspoint array with RRAM devices is dc biased at 0.8 V. These RRAM devices exhibit an ON/OFF current ratio of 30. In an unrectified crosspoint, the observed current ratio drops to less than two. The rectified crosspoint displays a current ratio of 28.5, only 5% less than the ideal ratio of an RRAM device. Furthermore, the total power consumption is reduced by almost an order of magnitude.

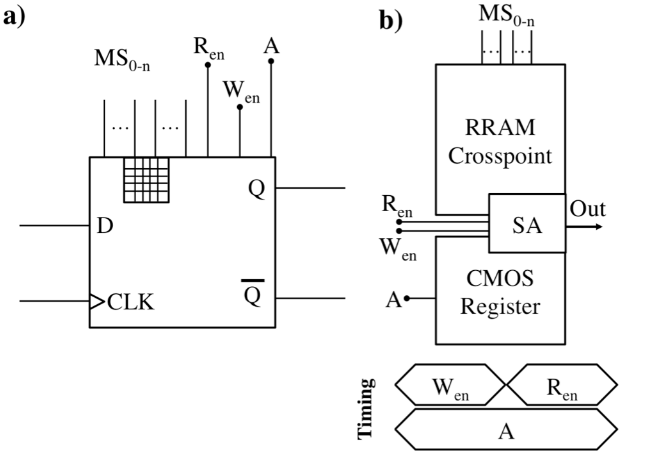

Fig. 3. Multistate register element (a) symbol of the multistate register and (b) block diagram with control signal timing. The symbol is similar to a standard CMOS D flip flop with the addition of a crosspoint array symbol.

III. RRAM MULTISTATE REGISTER

The multistate register is a novel circuit used to store multiple bits within a single logic gate. The multistate register is drop-in compatible with existing CMOS-based flip flops. The element uses a clocked CMOS register augmented by additional sense circuitry (SA) and global memristor select (MS) lines. The symbol and topology of the multistate register are shown in Fig. 3. Multistate registers can be used as pipeline registers within a processor pipeline, as shown in Fig. 4 and further explained in Section V.

The MS lines select individual RRAM devices within the crosspoint memory co-located with the CMOS register. A schematic of the proposed RRAM multistate register is shown in Fig. 5(a). The signals W_{en} and R_{en} are global control signals that, respectively, write and read within the local crosspoint memory. Signal A sets the CMOS register into an intermediate state that facilitates writes and reads from the crosspoint. An individual RRAM device is selected using a set of global MS lines. Local writes to the RRAM crosspoint array are controlled by the master stage within the register. The gates within the slave stage of the CMOS register are reconfigured to provide a built-in sense amplifier to read the RRAM crosspoint array [24]. The overhead of the additional circuitry (shown in Fig. 3) is relatively small (Section IV-B).

The multistate register primarily operates as a CMOS register. In this mode, the structure behaves as a standard D flip flop, where a single bit is stored and is active while the idle states are stored within the RRAM crosspoint array. When global control circuitry triggers a change of the pipeline state (e.g., for a pipeline stall or context switch), the circuit stores the current bit of the register and reads out the value of the next active bit from the internal RRAM-based storage. Switching between active bits consists of two phases. In the first half of the cycle, an RRAM write operation stores the current state of the register. During a write operation, the transmission gate A disconnects the first stage from the following stage, isolating the structure into two latches. The input latch stores the currently evaluated state, whereas the output latch stores the

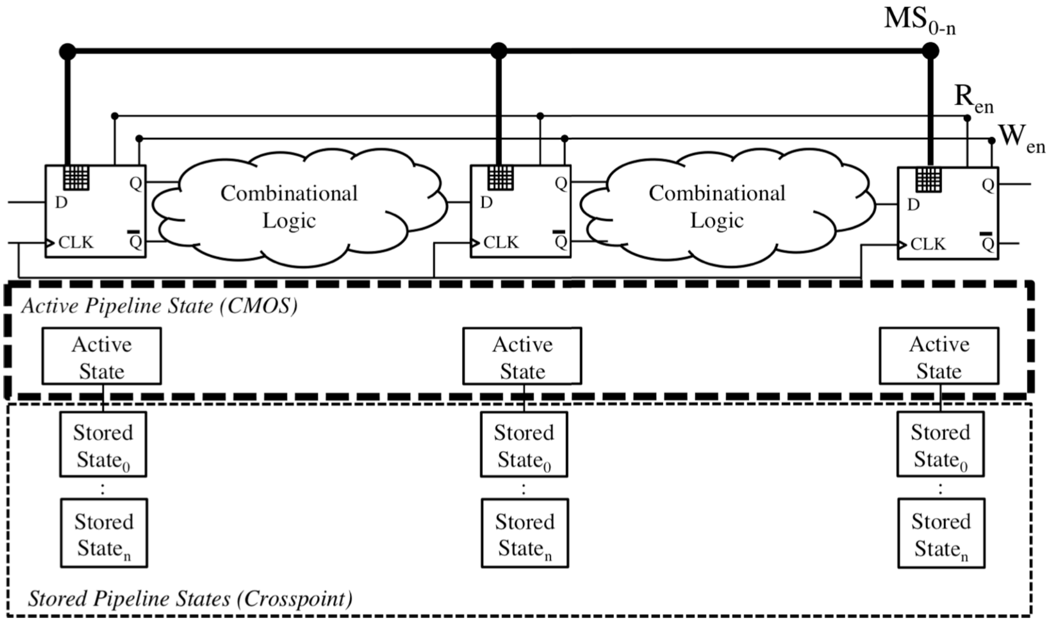

Fig. 4. MPR-based pipeline and logic diagram of active and stored pipeline states. The MPR replaces a conventional pipeline register and time multiplexes the stored states.

data of the previous state. Once W_{en} goes high, the input latch drives a pair of multiplexors that write the currently stored state into the RRAM cell selected by the global MS lines. The active devices during the write phase are shown in Fig. 5(b). The write phase may require more than half a cycle depending upon the switching time of the RRAM technology. During the second half of the clock cycle, the new active bit is selected within the resistive crosspoint array and sensed by the output stage of the CMOS D flip flop. During a read operation, the globally selected row is grounded through the common node N_{in}. The voltage on the common line N_{out} is set by the state of the RRAM cell. To bias the RRAM cell, the common line is connected through a pMOS transistor to the supply voltage V_{DD}. The voltage is sensed at the output of M_1. If R_{en} is set high, M_1 to M_5 reconfigure the last inverter stage as a single ended sense amplifier [13], and the crosspoint array is read. The active devices during the read phase are shown in Fig. 5(c).

The physical design of the multistate register can be achieved by two approaches. RRAM devices can be integrated between the first two metals, as shown in Fig. 6(a), or the RRAM can be integrated on the middle-level metal layers, as shown in Fig. 6(b). The middle metal layer approach allows the RRAM to be integrated above the CMOS circuitry, saving area. A standard cell floorplan is shown in Fig. 7(b), where a dedicated track is provided for the RRAM interface circuitry. This dedicated track runs parallel to the CMOS track. The addition of this track wastes area in those cases where multistate registers are sparsely located among the CMOS gates. Additional routing overhead increases the area required to pass signals around the crosspoint array.

The approach shown in Fig. 7(a), where the RRAM is integrated on the lower metal layers, requires slightly more area but is compatible with standard cell CMOS layout rules. Fabrication on the lower levels maintains standard routing conventions, where the lower metal layers are dedicated to routing within the gates, and the middle metal layers are used to route among the gates.

IV. Simulation Setup and Circuit Evaluation

The multistate register has been evaluated for use within a high-performance microprocessor pipeline. The latency, energy, and area of the register as well as the sensitivity to process variations are described in this section.

A. Latency and Energy

The latency and energy of an MPR are dependent on the parameters of an RRAM device and the CMOS sensing circuitry built into the MPR. The RRAM device is modeled using the TEAM model [21] based on the parameters listed in Table II. The parameters of the resistive device are chosen to incorporate device nonlinearity into the I–V characteristic, as shown in Fig. 2(b) and described in Section II-A. The multistate register is evaluated across a range of internal crosspoint array sizes (e.g., different number of states per register). The resistance of the device is extracted from [23]. The transistor and cell-track-sizing information is from the FREEPDK45 Standard Cell Library [25] and scaled to a 22-nm technology. Circuit simulations utilize the 22-nm PTM CMOS transistor model [26]. The RRAM and diode device parameters are listed in Table II. Standard CMOS timing information for the register is listed in Table III. The read operation requires 28.6 ps, equivalent to a 16 GHz clock frequency (the read operation is less than half a clock cycle). The register operates primarily as a CMOS register and only accesses the RRAM crosspoint array to switch between idle and active pipelines states. Note that the eight-row by eight-column crosspoint array is small compared with large-scale memory crosspoint arrays and therefore places a small electrical load on the

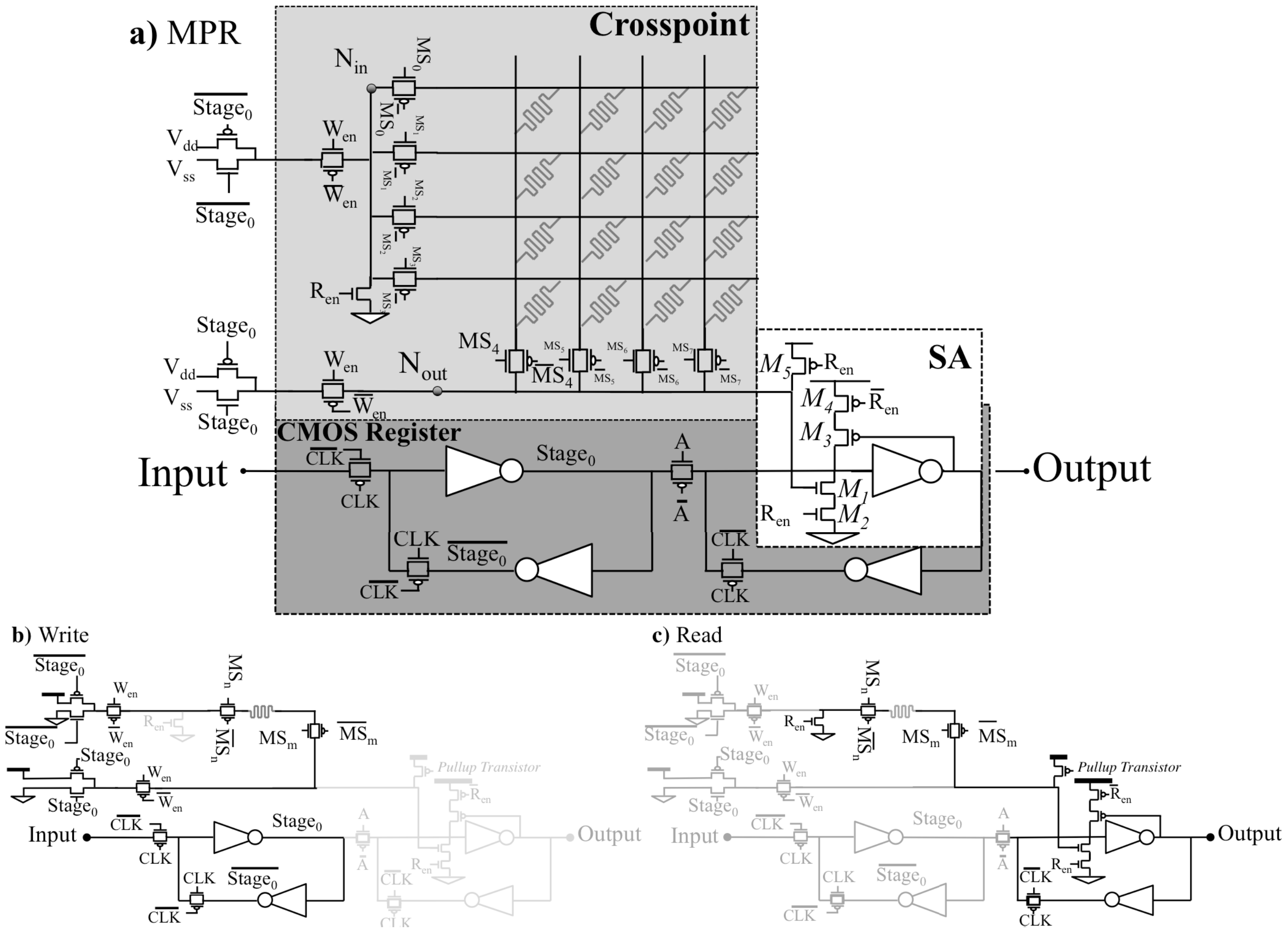

Fig. 5. Proposed RRAM MPR. (a) Complete circuit consists of a RRAM-based crosspoint array above a CMOS-based flip flop, where the second stage (the slave) also behaves as a sense amplifier. (b) Write and (c) read operations of the proposed circuit.

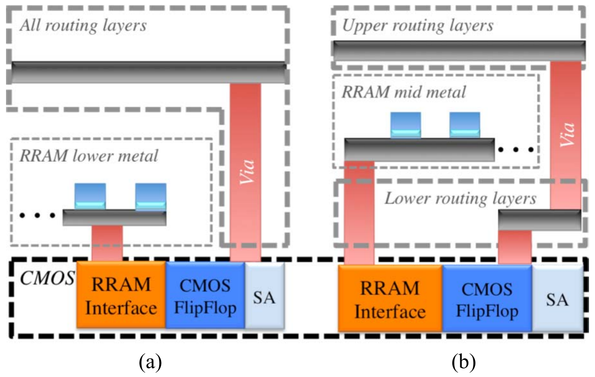

Fig. 6. Vertical layout of RRAM in MPR circuit for (a) lower level and (b) midlayer crosspoint RRAM array.

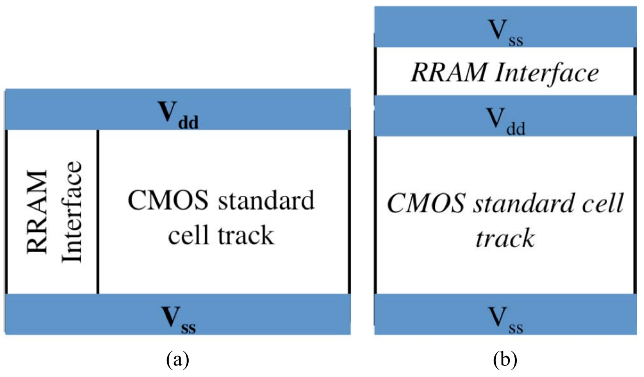

Fig. 7. Planar floorplan of MPR with lower metal and upper metal RRAM layers. The RRAM array is not marked in this figure since it is located above the CMOS layer and has a smaller area footprint. (a) Lower metal floorplan. (b) Mid metal floorplan.

sensing circuitry. Hence, the read operation is relatively fast and does not limit the operation of the multistate register.

The performance of the multistate register is limited by the switching characteristics of the RRAM device. To maintain high performance, the desired RRAM devices must be relatively fast [31]. These characteristics are chosen to achieve a target write latency of a 3 GHz CPU. As mentioned in Section II, the RRAM write operation occurs sequentially prior to the read operation. Because of the sequential nature of the multistate register access to the RRAM array, a half cycle is devoted to the read operation.

The energy of the multistate register depends upon the RRAM switching latency, as listed in Table IV. $E_{\text{Low-High}}$ and $E_{\text{High-Low}}$ are the energy required to switch, respectively,

TABLE II

MEMRISTOR AND DIODE PARAMETERS

R_{on} [kΩ]	0.5
R_{off} [kΩ]	30
k_{on}	-0.021-0.07
k_{off}	0.0021-0.007
$\alpha_{on,off}$	3
i_{on} [μA]	-1
i_{off} [μA]	1
V_{ON} (diode) [V]	0.5
R_{out} (diode) [Ω]	1

TABLE III

ACCESS LATENCY OF A 16-b MPR

Clock to Q [ps]	11.2
Setup Time [ps]	13.2
RRAM Read [ps]	28.6

TABLE IV

WRITE LATENCY AND ENERGY OF A 16-b MULTISTATE REGISTER

Write Time [cycles @ 3 GHz]	0.5	1.5	2.5	3.5	4.5
$E_{Low\text{-}High}$ [fJ]	2.24	5.26	8.3	10.49	13.23
$E_{High\text{-}Low}$ [fJ]	3.78	10.33	16.89	23.5	30.08

TABLE V

READ ACCESS ENERGY OF RRAM

States per Mutistate Register	4 States	16 States	64 States
$E_{read,Off}$ [fJ]	1.6	2.2	3.5
$E_{read,On}$ [fJ]	0.33	0.41	0.71

to R_{OFF} and R_{ON} for a single device write to the multistate register crosspoint array. Because the switching time of the memristor dominates the delay of a write to the multistate register, $E_{\mathrm{Low-High}}$ and $E_{\mathrm{High-Low}}$ increase linearly as the switching time increases. Note that the read energy only depends on R_{ON} and R_{OFF} and is therefore constant for different switching times. The read energy, however, depends on the size of the crosspoint array, as listed in Table V.

B. Layout and Physical Area

The energy and latency of an MPR are dependent on both the parameters of an RRAM device and on the CMOS sensing circuitry built into the MPR. An individual crosspoint RRAM cell is 0.001934 μm^2 ($4F^2$, where F is the feature size). The layout of the proposed RRAM multistate register is shown in Fig. 8. The layout of the multistate register is based on 45 nm design rules and scaled to the target technology of 22 nm. The number of RRAM devices within a crosspoint array is scaled from 4 to 64 devices. The MPR is evaluated for both the middle metal and lower metal approaches, as described in Section III. The physical area is listed in Table VI.

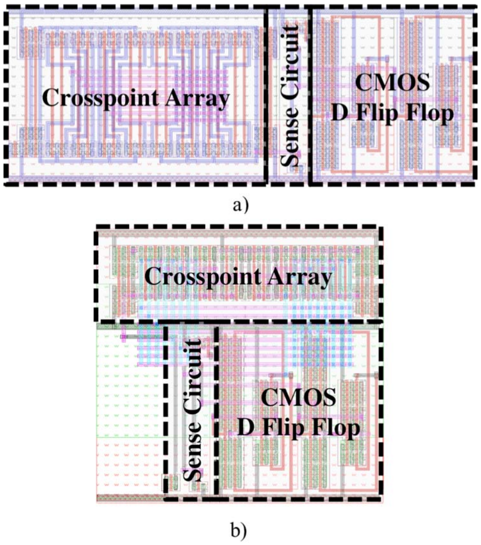

Fig. 8. Physical layout of 64-state MPR within the crosspoint array on (a) lower metal layers ($M1$ and $M2$) and (b) upper metal layers ($M2$ and $M3$) above the D flip flop.

TABLE VI

MPR AREA

		Area [μm^2]	Overhead [%]	Overhead per State [%]
	CMOS Register (1 state)	2.8	-	-
Lower Metal	MPR 4 states	5.5	96.2%	24%
	MPR 16 states	6.3	126.5%	8%
	MPR 64 states	8.1	192.5%	3%
Middle Metal	MPR 4 states	3.9	39.3%	9.8%
	MPR 16 states	4.3	53.6%	3.3%
	MPR 64 states	5.2	85.7%	1.3%

The transistors required to access the crosspoint, as shown in Fig. 8, dominate the area overhead of both the lower metal and middle metal multistate register. Because of the relatively small on-resistance of the RRAM devices, the access transistor needs to be sufficiently large to facilitate a write operation. In addition, CMOS transmission gates are used to ensure that there is no threshold voltage drop across the pass transistors. As a result, the area of the crosspoint memory array is only a small fraction of the area overhead of the multistate register. Note that alternative RRAM technologies with a higher R_{ON} supports smaller transistors and reduced area. Under these constraints, the most area efficient structure is a 64-b array, as the overhead per state is, respectively, 0.08 μm^2 for the lower metal approach and 3.75 μm^2 for the middle metal approach.

As shown, the middle metal register requires less area than a lower metal multistate register. As described in Section III and shown in Fig. 8(b), the middle metal register requires an additional track dedicated to the control transistors within the crosspoint array. Positioning the crosspoint array over the register also adds complexity as the upper metal layers can no longer be used to route signals above the multistate register.

C. Sensitivity and Device Variations

The built-in sense amplifier circuit senses the RRAM based on a threshold voltage. Any voltage above the threshold of the registers produces a logical zero at the output, and any voltage below the threshold produces a logical one. Similar to digital CMOS circuits, the structure is tolerant to variability in the RRAM resistance. To evaluate the sensitivity of the circuit to variations, the nominal R_{ON} is varied from 0.35 to 0.65 kΩ. This range produces a maximum and minimum change of ±2 mV in the voltage input of the sense amplifier. For 21 kΩ $>$ R_{OFF} $<$ 39 kΩ, a voltage ranging from −40 to +26 mV is produced. Both ranges represent a 30% variation in the device resistance of R_{ON} and R_{OFF}. In these cases, the correct output state is read out, indicating a high degree of tolerance to variations in the RRAM resistance.

The RRAM circuit can tolerate an R_{ON} of up to 12 kΩ before the circuit produces an incorrect output. In a 64-b multistate register, this behavior corresponds to an increase in the RRAM read delay from 78 to 476 ps. With increasing R_{ON}, the sense amplifier no longer generates a full range signal at the output, dissipating static energy. Much of this increased delay is due to the device operating near the switching threshold of the sense amplifier.

As R_{OFF} varies from 30 kΩ to 300 MΩ, the performance of the circuit improves owing to two effects. As the resistance increases, the voltage at the sense amplifier input also increases, placing the transistor into a higher bias state, which lowers the delay of the sense amplifier. Additionally, the large resistance of the sensed RRAM device prevents the sense line within the crosspoint array from dissipating charge, maintaining a high voltage at the input of the sense amplifier. Counterintuitively, this effect lowers the delay when R_{OFF} is greater than 30 MΩ. Because of the interplay of R_{ON} and R_{OFF}, a delay tradeoff exists between the average resistance of the RRAM technology and the resistive ratio of the device.

The gain and offset of the sense amplifier have a small effect on the circuit performance. A higher sense amplifier gain improves the tolerance of the sense circuit to variations of the RRAM device. An offset voltage shifts the reference threshold voltage, but must be comparable with the supply voltage (0.3 V_{DD} or more) before the circuit performance is affected.

V. MULTISTATE REGISTERS AS MPR FOR MULTITHREAD PROCESSORS—A TEST CASE

Replacing CMOS memory (e.g., register file and caches) with nonvolatile memristors significantly reduces power consumption. Multithreaded machines can exploit the high density and CMOS compatibility of memristors to store the state of the in-flight instructions with fine granularity within a CPU. Hence, using memristive technology can dramatically increase the number of threads running within a single core. This approach is demonstrated in this test case, where RRAM multistate registers store the state of multiple threads within a CPU pipeline.

TABLE VII
SoE MT AND CFMT PROCESSOR CONFIGURATIONS

	Switch on Event	RRAM-based CFMT
Number of pipeline stages	10	
CMOS process	22 nm	
Clock frequency [GHz]	3	
Switch penalty [cycles]	7	1 to 5
L1 read/write latency [cycles]	0	
L1 miss penalty [cycles]	200	
Data L1 cache configuration	32 kB, 4 way set associative	
Instruction L1 cache configuration	32 kB, 4 way set associative	
Branch predictor	Tournament , lshare 18kB/gshare 8kB	

TABLE VIII
PERFORMANCE SPEEDUP FOR DIFFERENT MPR WRITE LATENCIES COMPARED WITH SWITCH-ON-EVENT MULTITHREADING PROCESSOR FOR CPU SPEC 2006

Benchmark	**MPR Write Latency [clock cycles]**				
	1	**2**	**3**	**4**	**5**
libquantum	1.35	1.28	1.21	1.15	1.09
bwaves	1.22	1.15	1.08	1.04	1
milc	1.47	1.26	1.18	1.11	1.06
zeusmp	1.85	1.59	1.40	1.29	1.21
gromacs	1.53	1.32	1.21	1.17	1.14
leslie3d	1.67	1.48	1.33	1.22	1.15
namd	1.40	1.24	1.15	1.08	1.04
soplex.pds-50	1.35	1.28	1.21	1.16	1.1
lbm	1.5	1.31	1.2	1.12	1.08
bzip2.combined	1.13	1.1	1.08	1.05	1.03
gcc.166	1.35	1.28	1.21	1.15	1.09
gobmk.trevorc	1.3	1.24	1.19	1.14	1.09
h264ref.foreman_baseline	1.06	1.02	1	1	1
GemsFDTD	1.45	1.3	1.18	1.08	1.04
hmmer.nph3	1.18	1.14	1.11	1.07	1.04
soplex.ref	1.7	1.42	1.29	1.19	1.1
gcc.c-typeck	1.33	1.26	1.21	1.15	1.1
gobmk.trevord	1.29	1.23	1.18	1.13	1.08
Average	1.40	1.27	1.19	1.13	1.08

TABLE IX
ENERGY AND AREA EVALUATION FOR CFMT TEST CASE

	Switch on Event	**RRAM-based CFMT**	**Difference**
Thread switch energy [pJ]	109.9	9,1 @ 1 cycle penalty	-91.7%
		19.1 @ 2 cycle penalty	-82.6%
		29.2 @ 3 cycle penalty	-73.4%
		38.4 @ 4 cycle penalty	-65.1%
		48.2 @ 5 cycle penalty	-56.1%
Processor area [mm^2]	123.276	126.426	2.55%

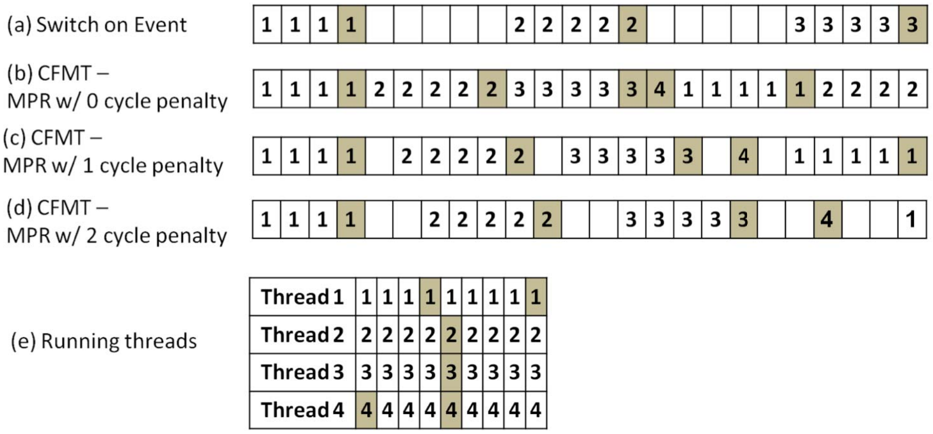
(a) Switch on Event
1 1 1 1 2 2 2 2 2 3 3 3 3 3
(b) CFMT – MPR w/ 0 cycle penalty
1 1 1 1 2 2 2 2 2 3 3 3 3 3 4 1 1 1 1 1 2 2 2 2
(c) CFMT – MPR w/ 1 cycle penalty
1 1 1 1 2 2 2 2 2 3 3 3 3 3 4 1 1 1 1 1
(d) CFMT – MPR w/ 2 cycle penalty
1 1 1 1 2 2 2 2 2 3 3 3 3 3 4 1
(e) Running threads
Thread 1 1 1 1 1 1 1 1 1 1
Thread 2 2 2 2 2 2 2 2 2 2
Thread 3 3 3 3 3 3 3 3 3 3
Thread 4 4 4 4 4 4 4 4 4 4

Fig. 9. Illustration of different multithreading techniques with four threads (marked by the number). All processors run the same four threads as shown in (e). The latency of the white and shaded instructions is, respectively, a single clock cycle and ten clock cycles. For (a) switch-on-event multithreaded processor, the long latency instruction that triggers a thread switch is the shaded instruction and the thread switch penalty is five clock cycles owing to the pipeline flush. For continuous flow multithreading, the thread switch penalty depends upon the read and write times of the multistate register, and is lower than traditional switch-on-event processors. The different thread switch penalties illustrated in this example are (b) zero for an ideal multistate register, (c) one clock cycle, and (d) two clock cycles. The performance (measured by the instructions per cycles) in this example is (a) 7/12, (b) 1 (71% improvement as compared with switch-on-event), (c) 0.83 (43% improvement), and (d) 0.67 (14% improvement) [12].

TABLE X
ENERGY PER INSTRUCTION FOR VARIOUS CPU SPEC 2006 BENCHMARK APPLICATIONS

Benchmark	SoE MT [pJ/inst.]	CFMT RRAM MPR – various thread switch latencies				
		1 cycle [pJ/inst.]	2 cycles [pJ/inst.]	3 cycles [pJ/inst.]	4 cycles [pJ/inst.]	5 cycles [pJ/inst.]
libquantum	15.17	14.12	14.29	14.46	14.63	14.80
bwaves	19.63	18.83	19.03	19.25	19.42	19.42
milc	24.51	22.61	23.23	23.47	23.74	24.11
zeusmp	21.10	18.04	18.62	19.19	19.18	19.95
gromacs	30.16	27.94	28.62	29.05	29.23	29.34
leslie3d	27.27	24.72	25.20	25.68	26.08	26.39
namd	22.90	21.42	21.91	22.21	22.50	22.65
soplex.pds-50	17.62	16.52	16.71	16.88	17.03	17.20
lbm	22.54	20.29	20.90	21.36	21.76	21.94
bzip2.combined	21.86	21.44	21.51	21.65	21.65	21.72
gcc.166	19.37	18.32	18.49	18.66	18.83	19.01
gobmk.trevorc	23.05	22.15	22.28	22.71	22.56	22.71
h264ref.foreman_baseline	25.95	25.27	25.35	25.50	25.69	25.76
GemsFDTD	23.89	21.88	22.43	22.99	23.36	23.49
hmmer.nph3	24.27	23.65	23.75	23.84	23.84	24.04
soplex.ref	21.92	19.47	20.04	20.44	20.80	21.17
gcc.c-typeck	19.94	19.16	19.12	19.27	19.43	19.58
gobmk.trevord	22.73	21.71	21.87	22.40	22.25	22.40
Average	22.44	20.97	21.30	21.61	21.78	21.98

In continuous flow multithreading [13], the multistate registers are used as MPRs to store the state of multiple threads. A single thread is active within the pipeline and the instructions from the other threads are stored within the MPRs. The MPRs therefore eliminate the need to flush instructions within the pipeline, significantly improving the performance of the processor, as shown in Fig. 9.

To exemplify this behavior, the performance and energy of a CFMT processor with the proposed RRAM-based MPRs have been evaluated [27]. To evaluate the performance, the GEM5 simulator [28] is extended to support CFMT. The energy has been evaluated by the McPAT simulator [29]. The simulated processor is a ten-stage single scalar ARM processor, where the execution stage operates at the eighth stage. The performance and energy of the CFMT processor are compared with a switch-on-event (SoE) multithreading processor [30], where a thread switch occurs for each long-latency instruction (e.g., L1 cache miss, floating point instructions), causing the pipeline to flush. The characteristics of the evaluated processors are listed in Table VII. The energy is compared with a 16 thread processor (i.e., with an MPR storing 16 states) which is a sufficient number of threads to achieve the maximum performance for most benchmark applications.

The performance of the processors is measured by the average number of instructions per clock cycle (IPC), as listed in Table VIII. The average speedup in performance is 40%.

A comparison of the thread switch energy is listed in Table IX. The average energy per instruction for various CPU SPEC 2006 benchmarks is listed in Table X, where the average reduction in energy is 6.5%. The area overhead for a 16-thread CFMT compared with an SoE is approximately 2.5%, as listed in Table IX.

For the CFMT configuration described herein, the simulations show that 16 threads are sufficient to achieve the maximum performance for the vast majority of SPEC CPU 2006 benchmarks. Alternate configurations with many long-latency events or different machines may benefit from additional states.

Physically, a linear increase in the number of rows and columns within a crosspoint array generates a quadratically increasing number of states and physical area, increasing the efficiency of the crosspoint array. A small increase in the number of rows and columns supports many more threads. However, as previously mentioned, 64 states is sufficient for most applications.

For the MPR to enhance performance, the cost of a thread switch must be smaller than the latency of a cache miss or other long-latency events. This situation is typical for all practical thread switching events.

VI. Conclusion

Emerging memory technologies, such as RRAM, are more than just a drop-in replacement to existing memory technologies. In this paper, an RRAM-based multistate register is proposed using an embedded array of memristive memory cells within a single flip flop. The multistate register can store additional data that is not conventionally contained within a computational pipeline.

The proposed multistate register is relatively fast owing to the physical closeness of the CMOS and RRAM devices. A 16-state multistate register requires only 54% additional area compared with a single-state standard register. The multistate register also exhibits relatively consumes low power because of the non-volatility of the resistive devices.

As an example, the proposed multistate register has been applied to a continuous-flow multithreaded processor, exhibiting a significant 40% performance improvement with low energy compared with a conventional switch-on-event processor. An RRAM-based MPR therefore enables novel microarchitectures, such as the CFMT. The proposed multistate register significantly improves performance and reduces energy with small area overhead.

Acknowledgment

The authors would like to thank Y. Etsion, Y. H. Nacson, and U. C. Weiser for their contributions.

References

[1] 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.

[2] M. Hosomi *et al.*, "A novel nonvolatile memory with spin torque transfer magnetization switching: Spin-RAM," in *Proc. IEEE Int. Electron Devices Meeting*, Dec. 2005, pp. 459–462.

[3] L. Chua, "Resistance switching memories are memristors," *Appl. Phys. A*, vol. 102, no. 4, pp. 765–783, Mar. 2011.

[4] E. Salman and E. G. Friedman, *High Performance Integrated Circuit Design*. New York, NY, USA: McGraw-Hill, 2012,

[5] H.-S. P. Wong *et al.*, "Metal-oxide RRAM," *Proc. IEEE*, vol. 100, no. 6, pp. 1951–1970, Jun. 2012.

[6] Y. Ho, G. M. Huang, and P. Li, "Nonvolatile memristor memory: Device characteristics and design implications," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, Nov. 2009, pp. 485–490.

[7] D. R. Lamb and P. C. Rundle, "A non-filamentary switching action in thermally grown silicon dioxide films," *Brit. J. Appl. Phys.*, vol. 18, no. 1, pp. 29–32, Jan. 1967.

[8] B. Govoreanu *et al.*, "10 × 10 nm^2 Hf/HfO$_x$ crossbar resistive RAM with excellent performance, reliability and low-energy operation," in *Proc. IEEE Int. Electron Devices Meeting*, Dec. 2011, pp. 31.6.1–31.6.4.

[9] J. J. Yang, M. D. Pickett, X. Li, D. A. Ohlberg, D. R. Stewart, and R. S. Williams, "Memristive switching mechanism for metal/oxide/metal nanodevices," *Nature Nanotechnol.*, vol. 3, no. 7, pp. 429–433, Jun. 2008.

[10] J. Li and J. F. Martinez, "Power-performance implications of thread-level parallelism on chip multiprocessors," in *Proc. IEEE Int. Symp. Perform. Anal. Syst. Softw.*, Mar. 2005, pp. 124–134.

[11] D. M. Tullsen, S. J. Eggers, and H. M. Levy, "Simultaneous multithreading: Maximizing on-chip parallelism," in *Proc. 22nd Annu. IEEE/ACM Int. Symp. Comput. Archit.*, May 1995, pp. 392–403.

[12] (Nov. 2, 2013). *Intel Ivy Bridge Specifications (Two Threads Per Core)*. [Online]. Available: http://ark.intel.com/

[13] S. Kvatinsky, Y. Nacson, Y. Etsion, E. G. Friedman, A. Kolodny, and U. Weiser, "Memristor-based multithreading," *IEEE Comput. Archit. Lett.*, to be published.

[14] L. O. Chua, "Memristor—The missing circuit element," *IEEE Trans. Circuit Theory*, vol. 18, no. 5, pp. 507–519, Sep. 1971.

[15] L. O. Chua and S. M. Kang, "Memristive devices and systems," *Proc. IEEE*, vol. 64, no. 2, pp. 209–223, Feb. 1976.

[16] T. Prodromakis, K. Michelakisy, and C. Toumazou, "Fabrication and electrical characteristics of memristors with TiO$_2$/TiO$_{2+x}$ active layers," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2010, pp. 1520–1522.

[17] 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.

[18] G. M. Ribeiro, J. J. Yang, J. Nickel, A. Torrezan, J. P. Strachan, and R. S. Williams, "Designing memristors: Physics, materials science and engineering," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2012, pp. 2513–2516.

[19] H. Y. Lee *et al.*, "Low power and high speed bipolar switching with a thin reactive Ti buffer layer in robust HfO$_2$ based RRAM," in *Proc. IEEE Int. Electron Devices Meeting*, Dec. 2008, pp. 1–4.

[20] Y.-F. Chang *et al.*, "Study of SiO$_x$-based complementary resistive switching memristor," in *Proc. 70th Annu. Device Res. Conf.*, Jun. 2012, pp. 49–50.

[21] S. Kvatinsky, E. G. Friedman, A. Kolodny, and U. C. Weiser, "TEAM: Threshold adaptive memristor model," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, no. 1, pp. 211–221, Jan. 2013.

[22] Y. N. Joglekar and S. J. Wolf, "The elusive memristor: Properties of basic electrical circuits," *Eur. J. Phys.*, vol. 30, no. 4, pp. 661–675, Jul. 2009.

[23] J. J. Yang *et al.*, "Engineering nonlinearity into memristors for passive crossbar applications," *App. Phys. Lett.*, vol. 100, no. 11, p. 113501, Mar. 2012.

[24] K. Pagiamtzis and A. Sheikholeslami, "Content-addressable memory (CAM) circuits and architectures: A tutorial and survey," *IEEE J. Solid-State Circuits*, vol. 41, no. 3, pp. 712–727, Mar. 2006.

[25] (Apr. 2011). *FreePDK45 User Guide*. [Online]. Available: http://www.eda.ncsu.edu/wiki/FreePDK45

[26] W. Zhao and Y. Cao, "New generation of predictive technology model for sub-45 nm early design exploration," *IEEE Trans. Electron Devices*, vol. 53, no. 11, pp. 2816–2823, Nov. 2006.

[27] S. Kvatinsky *et al.*, "Multithreading with emerging technologies—Dense integration of memory within logic," to be published.

[28] (May 2012). *The gem5 Simulator System*. [Online]. Available: http://www.m5sim.org/

[29] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi, "McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures," in *Proc. IEEE/ACM Int. Symp. Microarchit*, Dec. 2009, pp. 469–480.

[30] S. Kvatinsky, E. G. Friedman, A. Kolodny, and U. C. Weiser, "The desired memristor for circuit designers," *IEEE Circuits Syst. Mag.*, vol. 13, no. 2, pp. 17–22, May 2013.

[31] M. K. Farrens and A. R. Pleszkun, "Strategies for achieving improved processor throughput," in *Proc. ACM Int. Symp. Comput. Archit.*, May 1991, pp. 362–369.

Ravi Patel received the B.Sc. and M.Sc. degrees in electrical and computer engineering from the University of Rochester, Rochester, NY, USA, in 2008 and 2010, respectively, where he is currently pursuing the Ph.D. degree.

He was a Research Intern with Freescale Semiconductor, Tempe, AZ, USA, in 2008 and 2010. His current research interests include memristors, STT-MRAM, and high-performance memories.

Shahar Kvatinsky received the B.Sc. degree in computer engineering and applied physics, and the M.B.A. degree from the Hebrew University of Jerusalem, Jerusalem, Israel, in 2009 and 2010, respectively, and the Ph.D. degree from the Department of Electrical Engineering, Technion - Israel Institute of Technology, Haifa, Israel.

He was with Intel Corporation, Santa Clara, CA, USA, as a Circuit Designer.

Eby G. Friedman (F'00) received the B.S. degree from Lafayette College, Easton, PA, USA, in 1979, and the M.S. and Ph.D. degrees from the University of California at Irvine, Irvine, CA, USA, in 1981 and 1989, respectively, all in electrical engineering.

He was with Hughes Aircraft Company, Glendale, CA, USA, from 1979 to 1991, where he became the Manager of the Department of Signal Processing Design and Test, responsible for the design and test of high-performance digital and analog ICs. He has been with the Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY, USA, since 1991, where he is currently a Distinguished Professor, and the Director of the High-Performance VLSI/IC Design and Analysis Laboratory. He is also a Visiting Professor with the Technion - Israel Institute of Technology, Haifa, Israel. His current research interests include high-performance synchronous digital and mixed-signal microelectronic design and analysis with an application to high-speed portable processors and low-power wireless communications. He has authored over 400 papers and book chapters, 12 patents, and has authored and edited 16 books in the fields of high-speed and low-power CMOS design techniques, 3-D design methodologies, high-speed interconnect, and the theory and application of synchronous clock and power distribution networks.

Dr. Friedman is a Senior Fulbright Fellow. is the Editor-in-Chief of the *Microelectronics Journal*, an Editorial Board Member of the *Analog Integrated Circuits and Signal Processing* and *Journal of Low Power Electronics*, the Chair of the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION SYSTEMS steering committee, and a Technical Program Committee Member of numerous conferences. He was the Editor-in-Chief of the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION SYSTEMS, the Regional Editor of the *Journal of Circuits, Systems and Computers*, an Editorial Board Member of the PROCEEDINGS OF THE IEEE, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II: ANALOG AND DIGITAL SIGNAL PROCESSING, the IEEE JOURNAL ON EMERGING AND SELECTED TOPICS IN CIRCUITS AND SYSTEMS, and the *Journal of Signal Processing Systems*, a member of the Circuits and Systems Society Board of Governors, and the Program and Technical Chair of several IEEE conferences. He was a recipient of the IEEE Circuits and Systems 2013 Charles A. Dosoer Technical Achievement Award, the University of Rochester Graduate Teaching Award, and the College of Engineering Teaching Excellence Award.

Avinoam Kolodny (SM'11) received the Ph.D. degree in microelectronics from the Technion - Israel Institute of Technology, Haifa, Israel, in 1980.

He joined Intel Corporation, Santa Clara, CA, USA, where he was involved in research and development in the areas of device physics, VLSI circuits, electronic design automation, and organizational development. He has been a member of the Faculty of Electrical Engineering with Technion since 2000. His current research interests include interconnects in VLSI systems, at both physical and architectural level.