

# Stateful Logic Using Phase Change Memory

BARAK HOFFER<sup>1</sup>, NICOLÁS WAINSTEIN<sup>1</sup> (Member, IEEE), CHRISTOPHER M. NEUMANN<sup>2</sup>,  
ERIC POP<sup>1,2</sup> (Fellow, IEEE), EILAM YALON<sup>1</sup> (Member, IEEE),  
and SHAHAR KVATINSKY<sup>1</sup> (Senior Member, IEEE)

<sup>1</sup>The Andrew and Erna Viterbi Faculty of Electrical and Computer Engineering, Technion–Israel Institute of Technology, Haifa 3200003, Israel

<sup>2</sup>Department of Electrical Engineering, Stanford University, Stanford, CA 94305 USA

CORRESPONDING AUTHOR: B. HOFFER (barakhoffer@campus.technion.ac.il)

This work was supported in part by the European Research Council through the European Union's Horizon 2020 Research and Innovation Program under Grant 757259 and in part by the European Research Council through the European Union's Horizon Europe Research and Innovation Program under Grant 101069336.

This article has supplementary downloadable material available at <https://doi.org/10.1109/JXDC.2022.3219731>, provided by the authors.

**ABSTRACT** Stateful logic is a digital processing-in-memory (PIM) technique that could address von Neumann memory bottleneck challenges while maintaining backward compatibility with standard von Neumann architectures. In stateful logic, memory cells are used to perform the logic operations without reading or moving any data outside the memory array. Stateful logic has been previously demonstrated using several resistive memory types, mostly resistive RAM (RRAM). Here, we present a new method to design stateful logic using a different resistive memory-phase change memory (PCM). We propose and experimentally demonstrate four logic gate types (NOR, IMPLY, OR, NIMP) using commonly used PCM materials. Our stateful logic circuits are different than previously proposed circuits due to the different switching mechanisms and functionality of PCM compared to RRAM. Since the proposed stateful logic forms a functionally complete set, these gates enable the sequential execution of any logic function within the memory, paving the way to PCM-based digital PIM systems.

**INDEX TERMS** Phase-change-memory (PCM), processing-in-memory (PIM), stateful logic.

## I. INTRODUCTION

FOR the last 75 years, computers have been typically designed in the von Neumann architecture, which separates the memory from the processing units. While their programming model is simple, incessant data movement limits system performance because memory access time is often substantially longer than the computing time. This bottleneck has worsened over the years since CPU speed has improved more than memory speed and bandwidth (the so-called “memory wall”) [1]. One attractive approach to deal with this problem is processing-in-memory (PIM), which suggests adding computation capabilities to the memory. PIM reduces the need for costly (in terms of processing speed, bandwidth, and energy) chip-to-chip transfers, thus yielding higher performance and energy efficiency [2].

An increasing number of applications from high-performance computing (HPC) to databases, data analytics, and deep neural networks require higher memory capacity to meet the needs of workloads with large datasets. DRAM scaling has slowed down in the last years, and it has become

a Herculean task to improve its capabilities further [3], [4]. Thus, new technologies, such as resistive random access memory (RRAM), conductive-bridge RAM (CBRAM), and phase-change memory (PCM), are being explored [5]. These technologies offer increased memory capacity and add non-volatility, enabling persistent memory. These types of persistent memories are usually referred to as storage class memory (SCM) [6], [7], as they combine both storage and memory characteristics. With SCM, applications stand to benefit from the availability of large-capacity memory, but the performance will still be limited by the incessant data movement between the CPU and memory.

*Stateful logic* [8], [9] is a PIM technique based on memristive memory technologies, for example, RRAM or CBRAM. In stateful logic gates, the input and output are represented in the form of resistance, and the result is written during the computation directly to the output memory cell without reading the input cells beforehand or moving any data outside the memory array [10]. When the stateful logic gates are functionally complete (e.g., NOR gates), any desired function

can be computed using a sequence of stateful logic operations [11], [12], [13]. Stateful logic enables PIM architectures such as the *memristive memory processing unit* (mMPU) [14] and *resistive accelerated computation for energy reduction* (RACER) [15] that offer massive intrinsic parallelism, high-performance, and energy-efficient processing, while maintaining backward compatibility with von Neumann architectures.

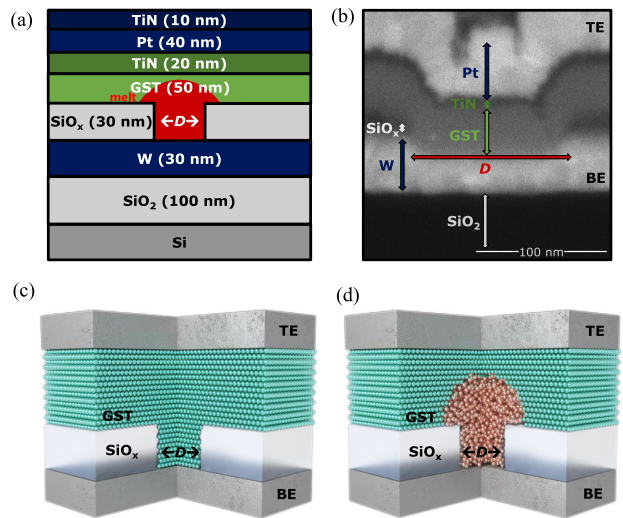
Prior studies on stateful logic mostly focused on bipolar RRAM devices [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], which still suffer from reliability and variability issues [7], [27] and are unavailable commercially in large scale. PCM is a more mature resistive technology that offers fast operation speed, low power, good reliability, and high-density integration, while being already used commercially [6], [28], [29], [30], for example, in the Intel Optane technology [31]. However, previous studies of computation using PCM mainly focused on analog neuromorphic computation [32], [33], [34], [35], [36], [37], [38], [39] or nonstateful binary logic operations [40], [41], [42], [43]. Since the *switching mechanism of PCM is unipolar and completely different than RRAM*, to achieve stateful logic using PCM devices, new circuit topologies and voltage schemes are needed. To the best of our knowledge, only a single stateful logic method was previously proposed for PCM, which was mentioned and demonstrated for a single test cycle in [44]. In this method, called input–output transfer, three sequential voltage pulses are required to perform a single AND operation.

In this article, we present and experimentally demonstrate a new method to perform stateful logic operations using PCM in a single step. We demonstrate four different logic gates (NOR, IMPLY, OR, and NIMP) with robust and repeatable results. Our logical set is functionally complete, enabling sequential execution of any logic function in memory. The gates are compatible with the memory crossbar structure and can be applied in parallel on multiple rows [14].

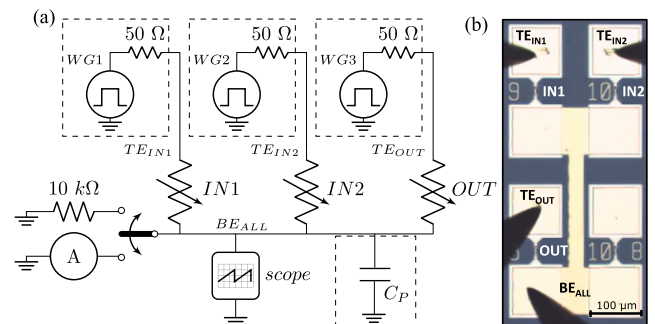
## II. STATEFUL LOGIC WITH PCM

### A. PCM DEVICES

PCM exploits the behavior of certain chalcogenide materials that can be switched rapidly and repeatedly between amorphous and crystalline phases [45]. These materials are typically compounds of germanium, antimony, and tellurium ( $\text{Ge}_x\text{Sb}_y\text{Te}_z$ , GST). The amorphous phase presents a high electrical resistivity, while the crystalline phase exhibits low resistivity. A PCM device consists of a certain volume of this phase change material sandwiched between two electrodes (see Fig. 1). Applying pulses to a PCM device results in Joule heating, which alters the phase (state) of the material. A reset pulse is used to melt a significant portion of the phase change material. When the pulse is stopped abruptly, the molten material quenches into the amorphous phase. Following the reset pulse, the device will be in a high resistive state (HRS). When a slower set pulse, with an amplitude above a threshold voltage ( $V_{th}$ ) [46] is applied to a PCM device in the HRS,



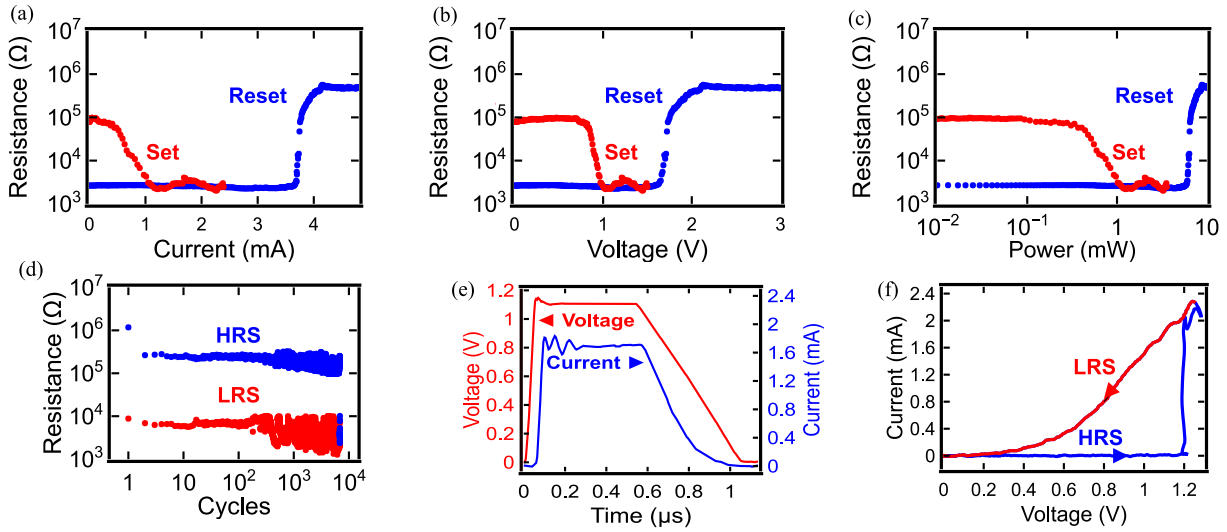
**FIGURE 1.** Confined PCM cell used in this work. (a) Cross section schematic. (b) Cross section scanning electron microscopy (SEM). (c) and (d) Three-dimensional cartoon of the crystalline and amorphous states. Evaporation and etching of tungsten (W) are used to form the BE. Sputtering and liftoff are used to pattern the GST layer, the TiN/Pt TE, and contact pads. GST is patterned into a confined area with a diameter  $D \sim 125$  nm.



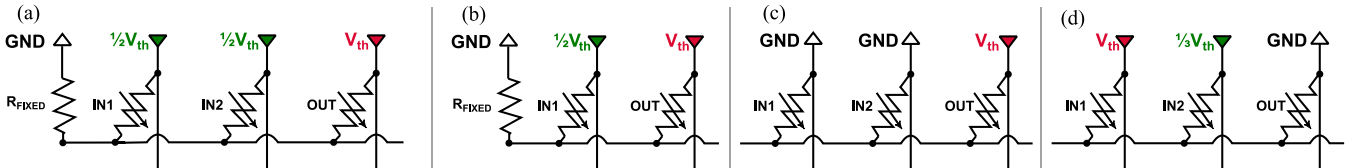
**FIGURE 2.** Experimental setup. (a) Schematic and (b) optical top view. Three cells are connected to a shared BE. Three waveform generators are connected to the top electrode of each cell. The BE can be switched between floating, grounded, and grounded using a 10-k $\Omega$  resistor.  $C_p$  marks the total parasitic capacitance at the shared node caused by the pads and probes.

the amorphous region crystallizes. After the SET pulse, the device will be in a low resistive state (LRS). The resistance state can be read by biasing the device with a low read voltage that does not change the phase configuration.

Our setup (see Fig. 2) includes three PCM cells with a shared bottom electrode (BE) and enables programming and reading each cell as well as performing the logic operations. A write-verify scheme is used to probe the maximum cycle count of a single device and characterize its switching behavior. For set and reset, we use 30/500/500 ns and 30/50/30 ns rise/width/fall pulses, respectively. The resistance is measured with a 0.2 V, 1  $\mu\text{s}$  read pulse. We consider resistance higher than 100 k $\Omega$  as HRS (logical ‘0’) and resistance lower than 10 k $\Omega$  as LRS (logical ‘1’). The current, voltage, and power required to set and reset the devices are depicted in Fig. 3(a)–(c). Here, multiple pulses are used, while varying



**FIGURE 3.** In-house fabricated PCM device characteristics. DC read resistance during the set and reset versus (a) programming current, (b) applied voltage, and (c) power. (d) Endurance data for a representative PCM device. The device was cycled using a write-verify scheme. The voltages for set and reset are 1.2 and 3 V, respectively. The rise/width/fall of the pulses for set and reset are 30/500/500 ns and 30/50/30 ns, respectively. (e) Current and voltage across the PCM cell during set operation. Threshold switching occurs within  $\sim 100$  ns and the voltage is kept high for the remaining time to complete the crystallization process. (f)  $I$ - $V$  transition of the device from the amorphous state (HRS) to the crystalline state (LRS) showing its threshold switching voltage, 1.2 V.  $I$ - $V$  was measured using a triangular voltage ramp with 1  $\mu$ s rise and fall times.



**FIGURE 4.** Schematics for PCM stateful logic gates mapping to the crossbar structure. (a) NOR gate. (b) IMPLY gate. (c) OR gate. (d) NIMP gate.

the voltage, until the set or reset event is encountered. Our endurance test shows that a device can maintain a  $10\times$  resistance window for almost  $10^4$  cycles, with some degradation in the resistance distribution after several hundred cycles [see Fig. 3(d)]. The current and voltage waveforms across the PCM cell during a typical set operation are shown in Fig. 3(e). The set operation is the basis for our proposed logic operations, where threshold switching occurs within  $\sim 100$  ns. Finally, the current-voltage ( $I$ - $V$ ) transition of the device from the amorphous state to the crystalline state is shown in Fig. 3(f).

## B. PCM STATEFUL LOGIC

The proposed PCM-based stateful logic gate consists of three PCM cells where two cells serve as inputs and the third device as the output (see Fig. 2). The output cell may also serve as an additional input at the cost of losing its stored data, that is, a destructive operation. A grounded fixed resistor (10 k $\Omega$  in our configuration) can be connected to the shared node as well (similar to the material implication in RRAM [8]). A logic operation is achieved by applying voltage pulses to the top electrode (TE) of the input cells, causing a conditional output switching, depending on the resistive states of the inputs. Note that the cells for IN1, IN2, and OUT are

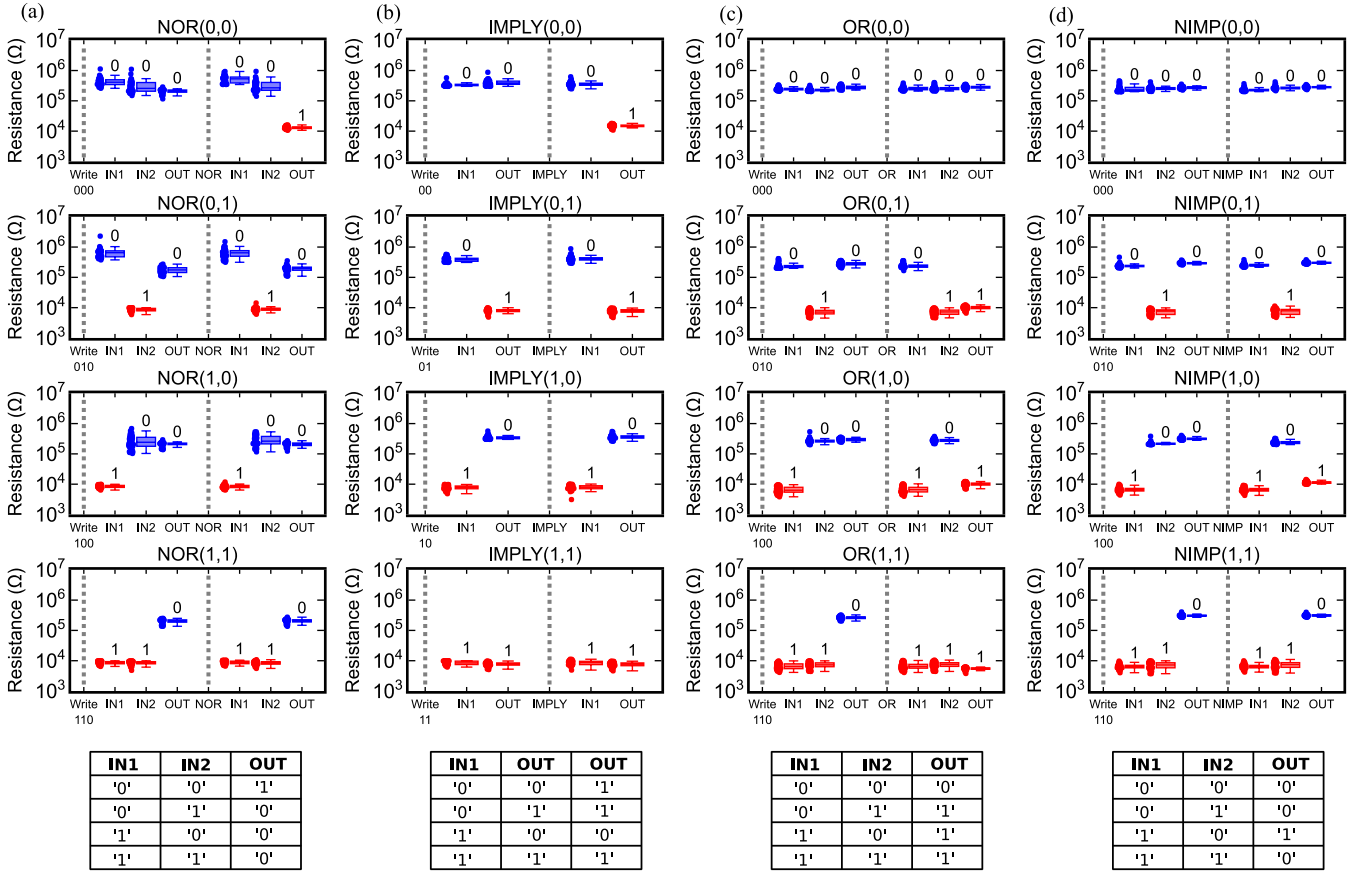
**TABLE 1.** Summary of the applied voltages and configurations at each node to realize the logic gates.

Gate	Voltage/Configuration			
	TE <sub>IN1</sub>	TE <sub>IN2</sub>	TE <sub>OUT</sub> *	BE <sub>ALL</sub> **
NOR	$\sim \frac{1}{2}V_{th}$	$\sim \frac{1}{2}V_{th}$	$\sim V_{th}$	R
IMPLY	$\sim \frac{1}{2}V_{th}$	F	$\sim V_{th}$	R
OR	0 V	0 V	$\sim V_{th}$	F
NIMP	$\sim V_{th}$	$\sim \frac{1}{3}V_{th}$	0 V	F

\* For the IMPLY gate, OUT serves also as input.

\*\* 'R' marks connection to the grounded fixed resistor, 'F' marks a node left floating.

interchangeable, as memory cells in the same row. The switching mechanism is based on the Ovonic threshold switching phenomenon [46] that occurs if the voltage across the output cell is above its threshold voltage,  $V_{th}$ , followed by the crystallization of the output cell. Since the logic operation is based on a switching event, the endurance data in

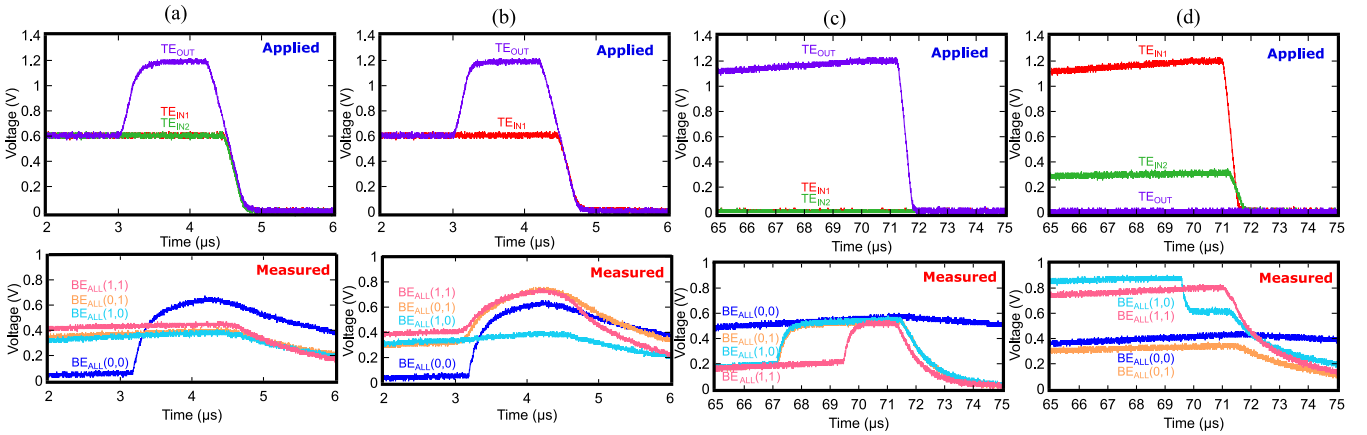


**FIGURE 5.** Experimental results of the proposed logic gates. Fifty iterations of (a) NOR, (b) IMPLY, (c) OR, and (d) NIMP gates. The  $x$ -axis is the operation or read cell, and the  $y$ -axis is the measured resistance plotted as a scatter and a median box. For the IMPLY gate, OUT is used also as input. All iterations of all the gates show correct logic operation and exhibit input stability.

Fig. 3(d) represents the degradation of the output device. The voltage selection and design methodology are similar to what is common in RRAM-based stateful logic [47], where voltage divider expressions are used to characterize the voltage distribution across the output and input cells. Furthermore, this design principle is compatible with the crossbar memory structure commonly used for PCM [38], [48]. Here, we propose four different logic functions (i.e., NOR, IMPLY, OR, and NIMP) based on this principle. Note that the fixed resistor required by some gates can be implemented as part of the peripheral circuitry for each word line in the crossbar. Fig. 4 and Table 1 summarize the voltages and configurations used to realize the different logic gates and their mapping to the crossbar structure. The crossbar structure is referred to as the general form of a crossbar memory with vertical bit lines and horizontal word lines. The same principles can be applied to state-of-the-art PCM crossbars [30] that mitigate the sneak-path and write-disturb phenomena by incorporating an Ovonic threshold selector (OTS) per cell, since the switching mechanism of an OTS is also based on a threshold voltage. However, in the presence of OTS, the distribution of voltages between the operating cells might change and affect the functionality of the gates. In future work, we plan

to further examine and verify stateful logic using PCM-OTS crossbars.

A NOR operation is realized by initializing the output cell to the HRS and grounding the shared BE using the fixed resistor. Applying  $\sim \frac{1}{2}V_{th}$  to  $TE_{IN1}$  and  $TE_{IN2}$ , and applying  $\sim V_{th}$  to  $TE_{OUT}$  causes the voltage across OUT to be approximately  $V_{th}$  only if both inputs are in HRS. Additionally, the state of the inputs remains unchanged because the maximum voltage across each input cell is  $\sim \frac{1}{2}V_{th}$ . Similarly, a destructive implication (IMPLY) operation is realized. Here,  $TE_{IN2}$  is kept floating, OUT serves as an input as well, and the voltage across OUT is  $\sim V_{th}$  only if IN1 is in HRS. An OR operation is realized by keeping the shared BE floating, grounding  $TE_{IN1}$ ,  $TE_{IN2}$ , and applying  $V_{th}$  to  $TE_{OUT}$ . This causes the voltage across the output cell to be approximately  $V_{th}$  only if at least one input is in the LRS. Similarly, a not implication (NIMP) operation is realized by keeping the shared BE floating, grounding  $TE_{OUT}$ , and applying  $\sim V_{th}$  to  $TE_{IN1}$  and  $\sim \frac{1}{3}V_{th}$  to  $TE_{IN2}$ . Here, the voltage across the output cell is approximately  $V_{th}$  only if IN1 is in LRS and IN2 is in HRS. As described in previous works [24], an XOR gate can be performed in two steps by running the NIMP operation twice on the same output with alternating inputs. Our logic



**FIGURE 6.** Voltages applied to the TE of the input–output cells to evaluate (a) NOR, (b) IMPLY, (c) OR, and (d) NIMP gates. The measured voltage on the BE shared node is plotted for each input case. (a) NOR gate; if at least one of the inputs is in LRS, the voltage at the shared node follows the constant voltage dictated by the inputs, thus keeping the voltage across OUT below the set threshold. (b) IMPLY gate; the voltage at the shared node changes according to the resistive state of the inputs. (c) OR gate; the voltage across OUT is higher than the set threshold if at least one input is in LRS, causing a switch event noticeable by the voltage change at the shared node. (d) NIMP gate; the voltage across OUT is higher than the set threshold if IN1 is in LRS and IN2 is in HRS, causing a switch event noticeable by the voltage change at the shared node.

**TABLE 2.** Experimental demonstrations of stateful logic.

	Technology	Functions	No. Steps	No. Tests Reported
[8]	RRAM	IMPLY	1	1
[49]	RRAM	IMPLY	1	80
[16]	RRAM	NOR, NOT	1	50
[17]	RRAM	NOR, NOT	1	1
[50]	RRAM	IMPLY	1	1
[18]	RRAM	NOR, NOT	1	1
[19]	RRAM	NAND, NOR, NIMP, MAJORITY3, PARITY3	1	1
[51]	RRAM	IMPLY	1	1
[21]	RRAM	IMPLY, NIMP, AND, OR, NOR	1	1
[23]	RRAM	NAND, NOR	1	50
[24]	RRAM	OR, NIMP	1	50
[44]	PCM	NOT, AND	2, 3	1
<b>This Work</b>	PCM	NOR, IMPLY, OR, NIMP	1	50

set is functionally complete (NOR by itself is functionally complete), and synthesis tools can be used to determine the required execution steps of any desired logic function by applying sequential operations of these gates [11].

### III. EXPERIMENTS

#### A. DEVICE FABRICATION

PCM devices are fabricated as in [52], starting with the evaporation and etching of tungsten (W) to form the BEs. Next,  $\text{SiO}_x$  is deposited with plasma-enhanced chemical vapor deposition (PECVD) and patterns the confined vias using e-beam lithography. In situ Ar sputtering is used to make sure

the top of the BE is not oxidized. Then, sputtering and liftoff are used to pattern the GST layer with in situ TiN capping, and the final TiN/Pt TEs and contact pads.

#### B. ELECTRICAL MEASUREMENTS

The measurement setup is shown in Fig. 2; it includes three PCM cells and enables programming and reading each cell, as well as performing the logic operations. Electrical measurements are performed on-wafer using a Keysight B1500A with four B1530 WGF MU channels and a Keysight MSOX3104T oscilloscope.

#### C. EXPERIMENTAL DEMONSTRATION

We measured the functionality and robustness of the proposed gates on the fabricated devices. In each test iteration, we examine the four input combinations for the tested gate. Each experiment includes: 1) a write-verify procedure to initialize the inputs and output to the desired states; 2) applying the voltage pulses required to evaluate the logic function; and 3) cells reading to examine the output result and to verify the stability of the inputs. The write-verify procedure includes using the voltage pulses to set the cells to the desired state, verifying the result using a read pulse, and applying the voltage pulses again until the resistance is in the desired range. The voltage pulses and resistance ranges are as described in Section II-A. A stateful operation is evaluated not only by the correct logical result, but also by the stability of the inputs. This is not always trivial, as reported by previous RRAM works [23], [24], [27], [53]. Results of 50 iterations for: 1) NOR; 2) IMPLY; 3) OR; and 4) NIMP logic gates are shown in Fig. 5. For each gate, all test iterations were performed sequentially on the same couple/triplet of devices. The results show successful logic operation for all iterations on all gates. Additionally, the inputs remain stable, without any meaningful change in their resistance. Therefore,

the input degradation is negligible. However, we note that our work is a proof-of-concept and to have conclusive results regarding degradation, additional measurements on larger arrays are required. Table 2 compares previous experimental demonstrations of stateful logic, primarily using RRAM, with our demonstration using PCM. Note that we do not compare energy and latency numbers, since the experimental demonstrations are proofs-of-concept and usually use unoptimized, university-fabricated devices that are hard to compare. Furthermore, the different measurement setups in each work may affect the results. Previous PCM stateful logic work requires two to three steps for computation, while the proposed PCM stateful logic uses a single step for the execution of different functions. We do not count output initialization as a computation step in this comparison since all methods require it. Compared to RRAM stateful logic, the complexity of our operations is similar, and the actual difference in terms of latency and energy lies in the different device properties and switching mechanisms. In future work, we plan to further examine the differences between RRAM and PCM-based stateful logic methods using experimental measurements.

In the NOR gate test, we apply 0.6 V for 3  $\mu$ s on all TEs, then we increase the voltage on TE<sub>OUT</sub> to 1.2 V for 1  $\mu$ s, while keeping the voltage on TE<sub>IN1</sub> and TE<sub>IN2</sub> at 0.6 V. We apply the same voltage scheme for the IMPLY gate test, but TE<sub>IN2</sub> is kept floating. The value for the fixed resistor in the NOR and IMPLY gates was selected as 10 k $\Omega$ , the max value for LRS, to assure the current passing through the output is high enough for complete crystallization during output switching, while keeping the voltage across the output lower than the threshold voltage for the nonswitching cases. In the OR gate test, we keep the shared BE floating, ground TE<sub>IN1</sub> and TE<sub>IN2</sub>, and apply 1.2 V on TE<sub>OUT</sub> with a rise time of 70  $\mu$ s and pulselength of 1  $\mu$ s. Similarly, for the NIMP gate, we keep the shared BE floating, ground TE<sub>OUT</sub>, and apply 1.2 V on TE<sub>IN1</sub> and 0.35 V on TE<sub>IN2</sub> with a rise time of 70  $\mu$ s and pulselength of 1  $\mu$ s.

The shape of the voltage pulses used for the demonstration of the NOR and IMPLY gates has two parts, whereas the pulses for the OR and NIMP gates have relatively long rise time. These pulses were selected to compensate for the long  $RC$  delays to the shared node in our experimental setup and will be substantially shorter in an integrated design. The  $RC$  delays are caused by a combination of the large parasitic capacitance of the pads and probes, and the effective resistance that drives the shared node differs between gates, due to their different circuits. In the OR and NIMP gates, this problem is most prominent in the IN<sub>1</sub> = IN<sub>2</sub> = '0' input case, since the impedance that drives the shared node is relatively high (the inputs and the output are in HRS). Without using a long rise time, the output might switch regardless of the state of the inputs or the inputs might change, since it takes time for the voltage at the shared node to update to its steady-state voltage. For the NOR and IMPLY gates, this issue is less distinct, since the circuit uses a small fixed resistor that connects the shared node to the ground, which decreases the

value of  $RC$ . Nevertheless, we chose to use a pulse with two parts to deal with the  $RC$  delay for the NOR and IMPLY gates, since it is still meaningful. Although the  $RC$  delay here is significant since it is considerably large in the experimental setup, the results indicate that even in an integrated setup, parasitic capacitance might be a limiting factor that can affect crossbar size selection and the performance of the logic gates.

The applied voltage pulses to implement the gates and the measured voltage at the shared BE, marked as BE<sub>ALL</sub>, for each input case are depicted in Fig. 6. In the NOR and IMPLY tests, if one of the inputs is in LRS, the voltage at BE<sub>ALL</sub> follows the constant voltage dictated by the inputs, thus keeping the voltage across the output below the set threshold. Otherwise, the voltage at BE<sub>ALL</sub> remains at 0 V, and the output is switched once the voltage on its TE is above  $V_{th}$ . In the OR and NIMP tests, the voltage at BE<sub>ALL</sub> follows a constant trend for all nonswitching cases, keeping the voltage across the output cell below  $V_{th}$ . In the cases where the output is switched, a meaningful change in the voltage at BE<sub>ALL</sub> is noticeable, caused by the resistance change of the output.

#### IV. CONCLUSION

To tackle the incessant data movement between the CPU and memory, we propose adding computation capabilities to PCM technology, inspired by previously proposed stateful logic for RRAM. Since the PCM switching mechanism is fundamentally different than RRAM, new circuits are required. We experimentally demonstrate a new method to perform four stateful logic gates using PCM (NOR, IMPLY, OR, and NIMP) in a single step. The measured results show correct and robust logic operation with 50 test iterations demonstrated for each gate. The proposed gates are crossbar compatible, functionally complete, and can be executed simultaneously on multiple rows. This may reignite scientific interest in PCM technology, which was almost completely disregarded for stateful logic, and paves the path toward PCM-based digital PIM architectures.

#### ACKNOWLEDGMENT

Fabrication was performed at the Stanford Nanofabrication Facility (SNF), Stanford, CA, USA, and the Technion Micro-Nano Fabrication Unit (MNFU).

#### REFERENCES

- [1] J. L. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitative Approach*, 6th ed. Burlington, MA, USA: Morgan Kaufmann, 2017.
- [2] O. Mutlu, S. Ghose, J. Gómez-Luna, and R. Ausavarungnirun, "A modern primer on processing in memory," in *Emerging Computing: From Devices to Systems: Looking Beyond Moore and Von Neumann* (Computer Architecture and Design Methodologies), M. M. s. Aly and A. Chattopadhyay, Eds. Singapore: Springer, 2023.
- [3] J. A. Mandelman et al., "Challenges and future directions for the scaling of dynamic random-access memory (DRAM)," *IBM J. Res. Develop.*, vol. 46, nos. 2–3, pp. 187–212, Mar. 2002.
- [4] S. Shiratake, "Scaling and performance challenges of future DRAM," in *Proc. IEEE Int. Memory Workshop (IMW)*, May 2020, pp. 1–3.
- [5] S. Yu and P.-Y. Chen, "Emerging memory technologies: Recent trends and prospects," *IEEE Solid-State Circuits Mag.*, vol. 8, no. 2, pp. 43–56, Mar. 2016.

- [6] S. W. Fong, C. M. Neumann, and H.-S.-P. Wong, "Phase-change memory—Towards a storage-class memory," *IEEE Trans. Electron Devices*, vol. 64, no. 11, pp. 4374–4385, Nov. 2017.
- [7] Y. Chen, "ReRAM: History, status, and future," *IEEE Trans. Electron Devices*, vol. 67, no. 4, pp. 1420–1433, Apr. 2020.
- [8] J. Borghetti, G. S. Snider, P. J. Kuekes, J. J. Yang, D. R. Stewart, and R. S. Williams, "Memristive switches enable stateful logic operations via material implication," *Nature*, vol. 464, no. 7290, pp. 873–876, Apr. 2010.
- [9] S. Kvatinsky et al., "MAGIC—Memristor-aided logic," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 61, no. 11, pp. 895–899, Nov. 2014.
- [10] J. Reuben et al., "Memristive logic: A framework for evaluation and comparison," in *Proc. IEEE Int. Symp. Power Timing Modeling, Optim. Simulation*, Sep. 2017, pp. 1–8.
- [11] R. Ben-Hur et al., "SIMPLER MAGIC: Synthesis and mapping of in-memory logic executed in a single row to improve throughput," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 10, pp. 2434–2447, Oct. 2020.
- [12] R. Ronen et al., "The bitlet model: A parameterized analytical model to compare PIM and CPU systems," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 18, no. 2, pp. 1–29, Apr. 2022.
- [13] O. Leitersdorf, R. Ronen, and S. Kvatinsky, "MultiPIM: Fast stateful multiplication for processing-in-memory," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 3, pp. 1647–1651, Mar. 2022.
- [14] A. Haj-Ali, R. Ben-Hur, N. Wald, R. Ronen, and S. Kvatinsky, "Not in name alone: A memristive memory processing unit for real in-memory processing," *IEEE Micro*, vol. 38, no. 5, pp. 13–21, Sep. 2018.
- [15] M. S. Q. Truong et al., "RACER: Bit-pipelined processing using resistive memory," in *Proc. 54th Annu. IEEE/ACM Int. Symp. Microarchitecture (MICRO)*. New York, NY, USA: Association for Computing Machinery, Oct. 2021, pp. 100–116.
- [16] B. C. Jang et al., "Zero-static-power nonvolatile logic-in-memory circuits for flexible electronics," *Nano Res.*, vol. 10, no. 7, pp. 2459–2470, Jul. 2017.
- [17] H. Bae et al., "Functional circuitry on commercial fabric via textile-compatible nanoscale film coating process for fibertronics," *Nano Lett.*, vol. 17, no. 10, pp. 6443–6452, Oct. 2017.
- [18] B. C. Jang et al., "Memristive logic-in-memory integrated circuits for energy-efficient flexible electronics," *Adv. Funct. Mater.*, vol. 28, no. 2, Jan. 2018, Art. no. 1704725.
- [19] Z. Sun, E. Ambrosi, A. Bricalli, and D. Ielmini, "Logic computing with stateful neural networks of resistive switches," *Adv. Mater.*, vol. 30, no. 38, Sep. 2018, Art. no. 1802554.
- [20] W. Shen et al., "Stateful logic operations in one-transistor-one-resistor resistive random access memory array," *IEEE Electron Device Lett.*, vol. 40, no. 9, pp. 1538–1541, Sep. 2019.
- [21] K. M. Kim and R. S. Williams, "A family of stateful memristor gates for complete cascading logic," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 11, pp. 4348–4355, Nov. 2019.
- [22] N. Xu et al., "A stateful logic family based on a new logic primitive circuit composed of two antiparallel bipolar memristors," *Adv. Intell. Syst.*, vol. 2, no. 1, Jan. 2020, Art. no. 1900082.
- [23] Y. S. Kim et al., "Stateful in-memory logic system and its practical implementation in a TaO<sub>x</sub>-based bipolar-type memristive crossbar array," *Adv. Intell. Syst.*, vol. 2, no. 3, Mar. 2020, Art. no. 1900156.
- [24] B. Hoffer, V. Rana, S. Menzel, R. Waser, and S. Kvatinsky, "Experimental demonstration of memristor-aided logic (MAGIC) using valence change memory (VCM)," *IEEE Trans. Electron Devices*, vol. 67, no. 8, pp. 3115–3122, Aug. 2020.
- [25] Y. S. Kim, M. W. Son, and K. M. Kim, "Memristive stateful logic for edge Boolean computers," *Adv. Intell. Syst.*, vol. 3, no. 7, Jul. 2021, Art. no. 2000278.
- [26] L. Liu et al., "Low-power memristive logic device enabled by controllable oxidation of 2D HfSe<sub>2</sub> for in-memory computing," *Adv. Sci.*, vol. 8, no. 15, Aug. 2021, Art. no. 2005038.
- [27] J. H. In et al., "A universal error correction method for memristive stateful logic devices for practical near-memory computing," *Adv. Intell. Syst.*, vol. 2, no. 9, Sep. 2020, Art. no. 2000081.
- [28] H. Y. Cheng et al., "Novel fast-switching and high-data retention phase-change memory based on new Ga-Sb-Ge material," in *IEDM Tech. Dig.*, Feb. 2015, pp. 3.5.1–3.5.4.
- [29] H. Y. Cheng et al., "Si incorporation into AsSeGe chalcogenides for high thermal stability, high endurance and extremely low V<sub>th</sub> drift 3D stackable cross-point memory," in *Proc. IEEE Symp. VLSI Technol.*, Jun. 2020, pp. 1–2.
- [30] N. Gong et al., "A no-verification multi-level-cell (MLC) operation in cross-point OTS-PCM," in *Proc. IEEE Symp. VLSI Technol.*, Jun. 2020, pp. 1–2.
- [31] J. Choe. (2017). Intel 3D XPoint memory die removed from Intel Optane™ PCM. Tech Insights. [Online]. Available: <https://www.techinsights.com/blog/intel-3d-xpoint-memory-die-removed-intel-optanetm-pcm-phase-change-memory>
- [32] M. Suri et al., "Phase change memory as synapse for ultra-dense neuromorphic systems: Application to complex visual pattern extraction," in *IEDM Tech. Dig.*, Dec. 2011, pp. 4.4.1–4.4.4.
- [33] G. W. Burr et al., "Experimental demonstration and tolerancing of a large-scale neural network (165 000 synapses) using phase-change memory as the synaptic weight element," *IEEE Trans. Electron Devices*, vol. 62, no. 11, pp. 3498–3507, Nov. 2015.
- [34] A. Sebastian et al., "Temporal correlation detection using computational phase-change memory," *Nature Commun.*, vol. 8, no. 1, pp. 1–10, Dec. 2017.
- [35] I. Boybat et al., "Neuromorphic computing with multi-memristive synapses," *Nature Commun.*, vol. 9, no. 1, pp. 1–12, Jun. 2018.
- [36] V. Joshi et al., "Accurate deep neural network inference using computational phase-change memory," *Nature Commun.*, vol. 11, no. 1, pp. 1–13, Dec. 2020.
- [37] S. R. Nandakumar et al., "Mixed-precision deep learning based on computational memory," *Frontiers Neurosci.*, vol. 14, p. 406, May 2020.
- [38] A. Sebastian, M. Le Gallo, and E. Eleftheriou, "Computational phase-change memory: Beyond von Neumann computing," *J. Phys. D, Appl. Phys.*, vol. 52, no. 44, Aug. 2019, Art. no. 443002.
- [39] M. Xu et al., "Recent advances on neuromorphic devices based on chalcogenide phase-change materials," *Adv. Funct. Mater.*, vol. 30, no. 50, Dec. 2020, Art. no. 2003419.
- [40] Y. Li, Y. P. Zhong, Y. F. Deng, Y. X. Zhou, L. Xu, and X. S. Miao, "Nonvolatile 'AND,' 'OR,' and 'NOT' Boolean logic gates based on phase-change memory," *J. Appl. Phys.*, vol. 114, no. 23, Dec. 2013, Art. no. 234503.
- [41] B. Lu, X. Cheng, J. Feng, X. Guan, and X. Miao, "Logic gates realized by nonvolatile GeTe/Sb<sub>2</sub>Te<sub>3</sub> super lattice phase-change memory with a magnetic field input," *Appl. Phys. Lett.*, vol. 109, no. 2, Jul. 2016, Art. no. 023506.
- [42] D. Loke et al., "Ultrafast phase-change logic device driven by melting processes," *Proc. Nat. Acad. Sci. USA*, vol. 111, no. 37, pp. 13272–13277, Sep. 2014.
- [43] I. Giannopoulos, A. Singh, M. Le Gallo, V. P. Jonnalagadda, S. Hamdioui, and A. Sebastian, "In-memory database query," *Adv. Intel. Syst.*, vol. 2, no. 12, 2020, Art. no. 2000141.
- [44] M. Cassinerio, N. Ciochini, and D. Ielmini, "Logic computation in phase change materials by threshold and memory switching," *Adv. Mater.*, vol. 25, no. 41, pp. 5975–5980, Nov. 2013.
- [45] S. Raoux, F. Xiong, M. Wuttig, and E. Pop, "Phase change materials and phase change memory," *MRS Bull.*, vol. 39, no. 8, pp. 703–710, Aug. 2014.
- [46] S. R. Ovshinsky, "Reversible electrical switching phenomena in disordered structures," *Phys. Rev. Lett.*, vol. 21, no. 20, pp. 1450–1453, Nov. 1968.
- [47] N. Wald and S. Kvatinsky, "Design methodology for stateful memristive logic gates," in *Proc. IEEE Int. Conf. Sci. Electr. Eng. (ICSEE)*, Nov. 2016, pp. 1–5.
- [48] D. Ielmini and A. L. Lacaita, "Phase change materials in non-volatile storage," *Mater. Today*, vol. 14, no. 12, pp. 600–607, 2011.
- [49] G. C. Adam, B. D. Hoskins, M. Prezioso, and D. B. Strukov, "Optimized stateful material implication logic for three-dimensional data manipulation," *Nano Res.*, vol. 9, no. 12, pp. 3914–3923, Dec. 2016.
- [50] L.-J. Yu et al., "Stateful logic operations implemented with graphite resistive switching memory," *IEEE Electron Device Lett.*, vol. 39, no. 4, pp. 607–609, Apr. 2018.
- [51] Z.-Y. He et al., "Atomic layer-deposited HfAlO<sub>x</sub>-based RRAM with low operating voltage for computing in-memory applications," *Nanoscale Res. Lett.*, vol. 14, no. 1, p. 51, Dec. 2019.
- [52] C. M. Neumann, H.-S. P. Wong, E. Pop, and K. Saraswat. (2019). *The Effect of Interfaces on Phase Change Memory Switching*. [Online]. Available: <http://purl.stanford.edu/rq703nq8223>
- [53] A. Siemon, D. Wouters, S. Hamdioui, and S. Menzel, "Memristive device modeling and circuit design exploration for computation-in-memory," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2019, pp. 1–5.