Logic Operation Implementation Method with Single-Level Cell NAND Flash

Ji-Hoon Ahn[1], Minsuk Koo[2*], Yoon Kim[1,a*]
[1]School of Electrical and Computer Engineering, University of Seoul
[2]Dep. Of Computer Science & Engineering, Incheon National University
E-mail : jihoon1994@uos.ac.kr

***Abstract* - With the advent of the big data era, the memory wall has become a dominant issue in computer hardware design. To overcome the memory wall issue, processing-in-memory (PIM) technology has been actively researched with various types of memory devices including static random-access memory (SRAM), dynamic random-access memory (DRAM), resistive random-access memory (RRAM) and charge trap flash (CTF). However, a logic operation implementation methodology for single-level cell (SLC) NAND flash, has yet to be investigated. This paper proposes and validates a method for implementing logic operations on SLC NAND flash. The proposed logic operation implementation method was demonstrated by Sentaurus TCAD mixed-mode simulations.**

***Keywords*— processing-in-memory, memory, non-volatile memory, NAND flash, single-level cell NAND flash**

I. INTRODUCTION

Modern computers are designed based on the von Neumann architecture, where processing units and memory units are separated. The processing unit handles tasks such as logical operations and instruction execution, and the memory unit store large amount of data. The operation of a von Neumann architecture-based computer relies on the seamless exchange of data between these two units.

With the advancement of semiconductor technology, the traditional von Neumann architecture is facing a memory bottleneck issue. The data processing speed of the processing unit is continually improving, while the data access performance of the memory unit is relatively slow.

To complement this issue, processing-in-memory (PIM) technology is being developed to implement logic operations directly in the memory unit. PIM technology enables data processing within the memory unit, reducing data transfer delay and power between the processing unit and the memory unit.

As a form of PIM technology, there is a method that implementing logic operations on non-volatile memory arrays, and the results are directly stored in the memory during the computation. Research about implementing logic operations on non-volatile memory has been conducted on several types of memory devices like resistive random-access memory (RRAM) [1-2], charge trap flash (CTF) [3], and others. However, a logic operation implementation methodology for single-level cell (SLC) NAND flash, has yet to be investigated.

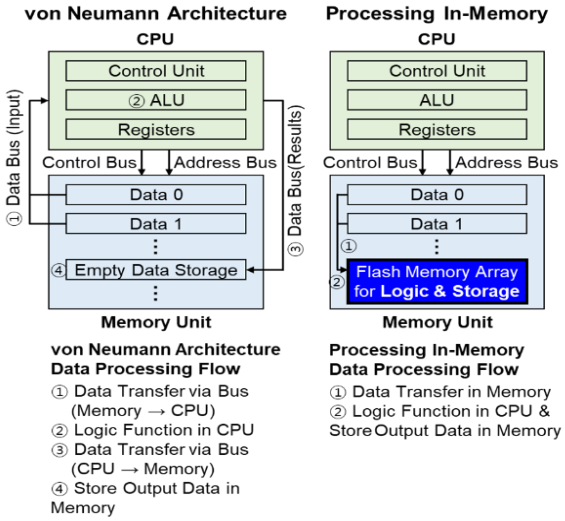


Figure 1. Data processing flow of von Neumann architecture and processing in-memory

In this paper, a method for implementing logic operations on SLC NAND flash is proposed and demonstrated. The proposed logic operation implementation method was verified by implementing XOR operations through Sentaurus TCAD mixed-mode simulations.

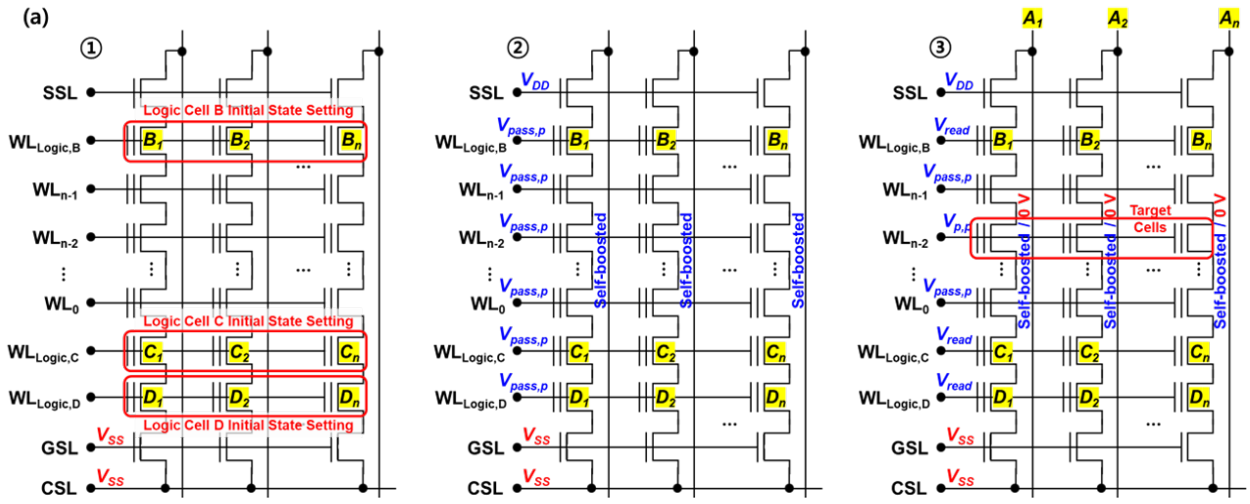
II.METHODS

A. Processing-in-memory

PIM technology performs logic operations within the memory unit to complement memory bottleneck from conventional von Neumann structure-based computer system. In the von Neumann architecture, the processing unit and memory unit are separate entities. When considering the process of performing processing based on data stored in the memory unit and storing the results back in the memory unit, all of the four operations are required (Fig. 1).

a. Corresponding author; yoonkim82@uos.ac.kr

Manuscript Received Jun. 14, 2023, Revised Oct. 24, 2023, Accepted Oct. 25, 2023



(b)

Boolean Logic	*A*	*B*	*C*	*D*
0	0	1	0	p
1	p	0	0	q
p	p	1	0	q
q	q	1	0	p
$\bar{p}$	0	p	0	q
$\bar{q}$	0	q	0	p
$p+q$	q	$\bar{p}$	0	0
$\bar{p}+\bar{q}$	$\bar{q}$	p	0	0
$p\cdot q$	p	$\bar{p}$	$\bar{q}$	$\bar{q}$
$\bar{p}\cdot\bar{q}$	0	p	q	1
$p+\bar{q}$	p	q	0	0
$\bar{p}+q$	q	p	0	0
$p\cdot\bar{q}$	p	$\bar{p}$	q	1
$\bar{p}\cdot q$	0	p	$\bar{q}$	$\bar{q}$
$p\cdot q+\bar{p}\cdot\bar{q}$	q	p	q	$\bar{p}$
$\bar{p}\cdot q+p\cdot\bar{q}$	$\bar{q}$	p	$\bar{q}$	$\bar{p}$

$[R = B' \cdot C' + B' \cdot D' + B \cdot C' + A \cdot D']$

(c)

Logic Variable	Variable Value	1
A	0	BL → V_{SS}
	1	BL → V_{DD}
B	0	Logic. cell B → Init. state '0'
	1	Logic. cell B → Init. state '1'
C	0	Logic. cell C → Init. state '0'
	1	Logic. cell C → Init. state '1'
D	0	Logic. cell D → Init. state '0'
	1	Logic. cell D → Init. state '1'

Figure 2. (a) Logic operation implementation method on NAND flash. (b) Logic variable allocation for 16 logic operations. (c) Logic variable utilization method

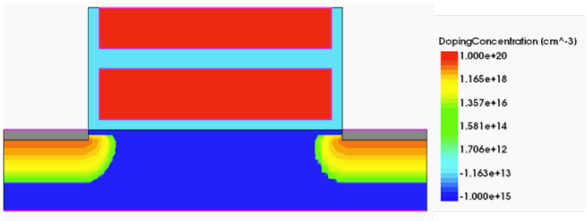
① The information stored in the memory unit is transferred to the processing unit via the data bus. ② The processing unit performs the necessary computational operations on the received information. ③ The results of the operations are transmitted back to the memory unit via the data bus. ④ Finally, the memory unit stores the data. Since the processing unit and memory unit are separate, there is a need for data transfer via the data bus. The bandwidth of the data bus is limited, which means that data transfer operations need to be repeated proportionally to the size of the data. In contrast, when using the proposed PIM architecture, only two operations are required: ① First, data is read from the memory unit. ② Logic operations are performed based on the read data. Simultaneously, data is stored in the memory, eliminating the need for separate additional data storage operations.

B. Logic operation on SLC NAND flash

To implement logic operations on SLC NAND flash, some flash cells within the NAND string are defined as logic cells, and these cells are utilized for logic operations. The topmost flash cell in the NAND string is defined as logic cell B, and the bottom two flash cells, in proximity to logic cell B, are respectively defined as logic cell C and logic cell D.

Logic operations on SLC NAND flash consist of two main actions. First, the logic cells within the NAND flash string are programmed to specific states depending on the logic variable B, C, D (Fig. 2(a)). Subsequently, logic operation and storing operation are performed on a NAND flash string. Initially, V_{SS} is applied to GSL, V_{DD} to SSL, V_{DD} to all BLs, and $V_{pass,p}$ to all WLs. Through this process, the channel voltage of all strings is self-boosted. Following this, the WL voltage of logic cells is lowered to V_{read}, and BL voltage is set to V_{DD} or V_{SS} based on the type of logic operation and input data. As a result of this action, if the operation result is '0', the channel is connected to V_{SS} and falls to 0 V, while if the operation result is '1', the channel maintains the self-boosted initial state. Finally, $V_{p,p}$ is applied to the target cell's WL to perform the program/inhibit operation. This action stores the logic operation result in the target cell.

Through this operation, 16 logic operations can be performed on a NAND flash string. The logic variables are assigned to specific values depending on the type of logic operation and input data values. Fig. 2(b) represents the assigned logic variable values based on the desired operation. For example, to perform XOR operation, logic variable A, B, C, and D should be assigned input data values q, p, q, and p'. Depending on the values of logic variables B, C, and D, the initial setting state of logic cell B, C, and D is determined (Fig. 2(c)). If the logic variable value is '0', the initial cell state is also '0'. Conversely, if the logic variable value is '1', the initial cell state is also '1'. Logic variable A determines the BL voltage during program/erase operations. When the logic variable value is '0', V_{SS} is applied, and when the logic variable value is '1', V_{DD} is applied.



Simulation Parameters	
Channel Length (L_g)	180 nm
Channel Width (W)	1 (AreaFactor)
Blocking Oxide Thickness	10 nm
Tunneling Oxide Thickness	5 nm
Gate Oxide	SiO_2
Channel Doping (N_c)	1×10^{15}
Drain Doping (N_d)	1×10^{19}
Source Doping (N_s)	1×10^{19}

Figure 3. Logic variable allocation for logic operation on single poly 4T eFlash cell

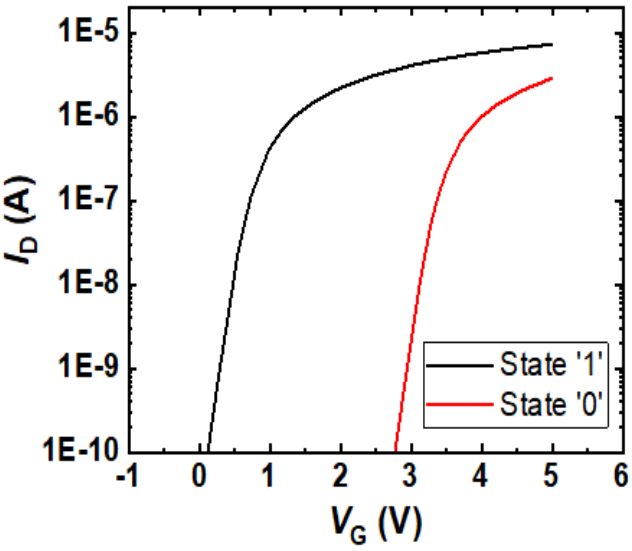


Figure 4. TCAD simulation parameters

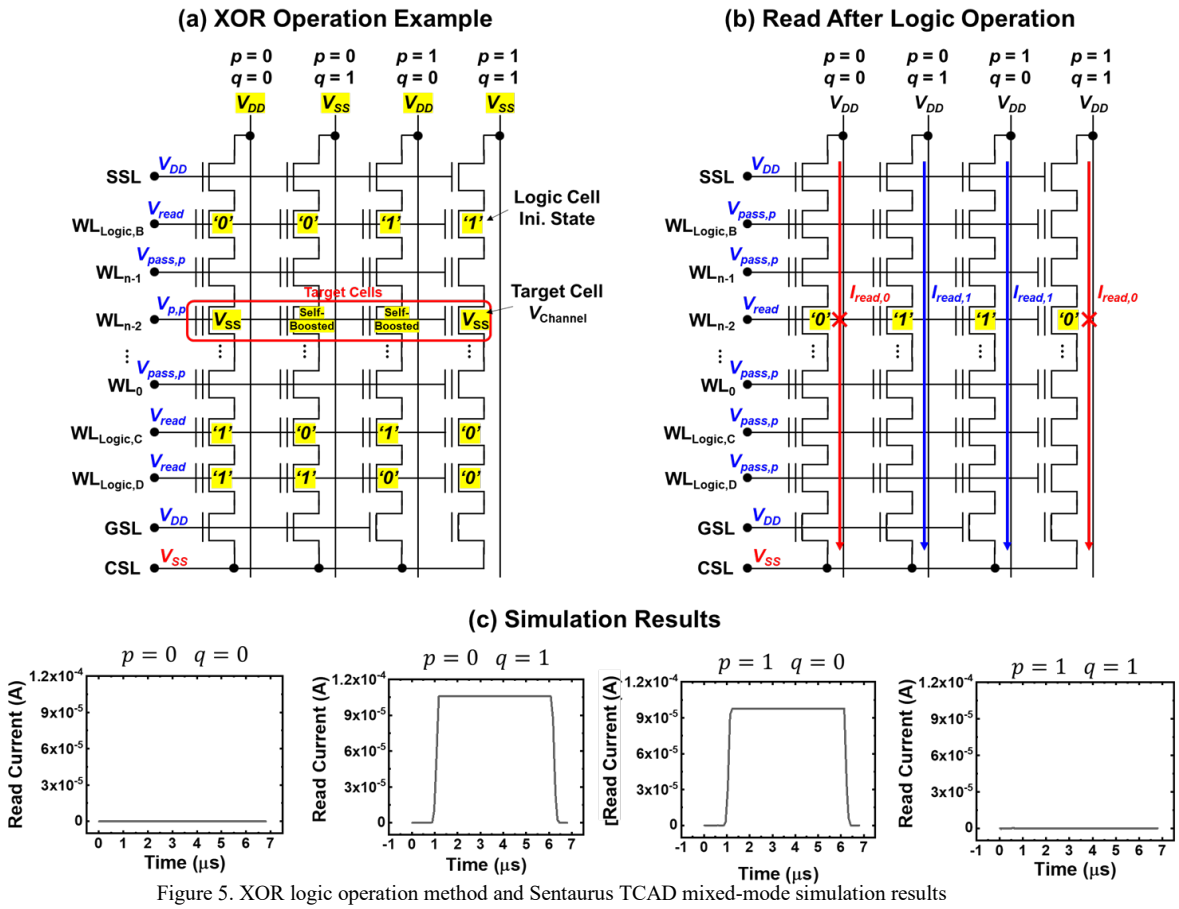


Figure 5. XOR logic operation method and Sentaurus TCAD mixed-mode simulation results

III. SIMULATION

To validate the implementation of logic operations on SLC NAND flash, a flash device was modeled using Sentaurus TCAD, and XOR logic operation behavior was confirmed through mixed-mode simulation. The flash device was modeled with a channel length of 180 nm, a width of 1 area factor, a blocking oxide thickness of 10 nm, and a tunneling oxide thickness of 5 nm (Fig. 3). Based on simulation, a Vth (threshold voltage) shift of approximately 2.7 V was observed (Fig. 4).

By connecting flash devices in series and modeling a NAND string using mixed-mode simulation, XOR logic operations were performed. Logic variable A, B, C, and D were assigned input data values q, p, q, and p' to perform XOR logic operations. The initial state of logic cells B, C, and D was set up as 011, 001, 110, and 100, respectively, based on input values 00, 01, 10, and 11 (Fig. 5(a)). In the subsequent program/erase operations, BL voltage was applied as V_{SS}, V_{DD}, V_{SS}, and V_{DD} for inputs 00, 01, 10, and 11, respectively. Ultimately, during the NAND string read operation (Fig. 5(b)), it was confirmed that for inputs 00, 01, 10, and 11, current values of approximately 0 μA, ~ 10 μA, ~ 10 μA, and ~ 0 μA were obtained (Fig. 5(c)) and this represents that the state of the target cell was programmed as 0, 1, 1, 0 for inputs 00, 01, 10, and 11, respectively. This corresponds to the XOR operation results for inputs 00, 01, 10, and 11, validating the implementation of logic operations on the NAND string.

IV. CONCLUSION

This research proposes a method for implementing logic operations on Single-Level Cell (SLC) NAND flash, which is one of the representative memory types. Based on the proposed method, the top flash cell and the bottom two flash cells in a NAND flash string are designated as logic cells. The initial state of these logic cells is set based on the type of operation and input data values, and the BL voltage is controlled to directly store the desired logic operation results in the flash cell. This proposed technology has been validated through Sentaurus TCAD mixed-mode simulations and can be utilized as a form of Processing-In-Memory (PIM) technology

ACKNOWLEDGMENT

This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (2021-0-01764, Charge-Storage-Memory-Based PIM Development, 50 %) and by National R&D Program through the National Research Foundation of Korea (NRF) funded by Ministry of Science and ICT(2021M3F3A2A01037928, 50 %). The Chip fabrication and EDA Tool were supported by the IC Design Education Center.

REFERENCES

[1] E. Linn, R. Rosezin, S. Tappertzhofen, U. Böttger, and R. Waser,“Beyond von Neumann—Logic operations in passive crossbar arrays alongside memory operations,” Nanotechnol., vol. 23, no. 30, Jul. 2012, Art. no. 305205. doi: 10.1088/0957-4484/23/30/305205.

[2] Z.-R. Wang, Y.-T. Su, Y. Li, Y.-X. Zhou, T.-J. Chu, K.-C. Chang, T.-C. Chang, T.-M. Tsai, S. M. Sze, and X. -S. Miao, “Functionally complete Boolean logic in 1T1R resistive random-access memory,” IEEE Electron Device Lett., vol. 38, no. 2, pp. 179–182, Feb. 2017. doi: 10. 1109/LED.2016.2645946.

[3] J.H. Lee, B.-G. Park, and Y. Kim, "Implementation of boolean logic functions in charge trap flash for in-memory computing." IEEE Electron Device Letters., vol. 40, no. 9, pp. 1358-1361, Jun. 2019.

.

JI-HOON AHN received the B.S. degree and M.S. degree in electrical and computer engineering from University of Seoul, Seoul, South Korea, in 2020 and 2022, respectively. He is currently pursuing a Ph.D. degree in electrical and computer engineering from University of Seoul, Seoul, South Korea.

MINSUK KOO received the B.S. degree in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST) in 2007, the M.S. degree in electrical engineering from Seoul National University in 2009, and the Ph. D degree in Electrical and Computer Engineering from Purdue University in 2020. From 2009 to 2012, he was a Senior Engineer with RadioPulse Inc., Seoul, South Korea, where he was involved in the development of ZigBee transceiver and SoC products. Since 2020, he has been with the Department of Computer Science and Engineering at Incheon National University. His research interests are in the hw/sw co-design for neuromorphic computing system including neural networks.

YOON KIM received the B.S. and Ph.D. degrees in electrical engineering from Seoul National University, Seoul, South Korea, in 2006 and 2012, respectively. From 2012 to 2015, he was a Senior Engineer with Samsung Electronics Company, South Korea. In 2015, he joined Pusan National University, Busan, South Korea, as an Assistant Professor. In 2018, he joined the University of Seoul and in 2020 became an Associate Professor.