Realization of Logical NOT Based on Standard DRAM Cells for security-centric Compute-in-Memory applications

Zheng Qiao [1,2], Jie Li[1,2], Chunyang Liu[1,2], Lei Guo[1,2], Pengpeng Ren[1,2], Sheng Ye[1,2], Bo Zhou[3], Jianfu Zhang[3], Zhigang Ji*[1], Junhua Liu*[1,4], Runsheng Wang[1,4] and Ru Huang[1,4]

[1]National Key Laboratory of Science and Technology on Micro/Nano Fabrication, Shanghai Jiao Tong university, Shanghai, China and Peking University, Beijing, China, [2]Department of Micro/Nano Electronics, School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong university, Shanghai, 200240, China, [3]School of Engineering, Liverpool John Moores University, Liverpool, UK, [4]Institute of Microelectronics, Peking University, Beijing, P.R. China

Email: zhigangji@sjtu.edu.cn, junhua.liu@pku.edu.cn

Abstract

In this study, a novel method, Bi-mode Activation, is proposed to implement logical NOT. For the first time, this is achieved without the need in modifying Dynamic Random Access Memory (DRAM) cells and core peripheral circuits, making it possible to be adopted in existing commercial DRAMs. The proposed method is successfully validated through simulation. Combined with the recently-proposed OR operation, the XOR-based data encryption is demonstrated, which paves ways for future security-centric in-memory-computing.

(Keywords: Compute-in-memory, DRAM, bitwise-NOT, encryption)

Introduction

Bitwise operations are widely used in modern binary-code-based computers, especially in applications such as data search [1], encryption [2], and neuro-inspired computing [3]. What is in common is that they all involve large amount of data transfer between the storage and computing units, causing the well-known "memory wall" issue that leads to delays and power consumption [4]. Compute-in-memory (CIM) is a promising candidate to solve this problem and various memory technologies have been exploration for this concept. Among them, DRAM, as one of the mainstream memories, has attracted people's attention in recent years due to the following two reasons: 1) Compared to the emerging non-volatile memory technologies such as RRAMs [4], its technology is mature and ready for implementation; 2) Compared with another mainstream memory, SRAM, it exhibits much higher storage density [5].

At present, several bitwise-CIM using DRAM have been developed. Ambit proposed by Vivek Seshadri et al makes use of the charge sharing and achieved AND/OR operation, however, the modification of DRAM standard cells is required for achieving logic NOT [6]. Lei Jiang et al simply re-designed the standard cell by adding 22 extra transistors and implemented the XOR logic. However, the methods mentioned above all require modification of the storage unit. Faced with this problem, recently, researchers from Princeton University proposed the DRAM-based CIM with slight modification of the peripheral circuit and used the time violation [7]. This ensures the minimum impact on the DRAM cell, which is critical to the DRAM manufacturers. However, the proposed method still can only realize AND/OR.

In this work, for the first time, we propose a solution to perform the logic NOT, which do not need to change either the memory cell or the core peripheral circuits. Together with the AND/OR logic from the operations proposed by the predecessors, a logical-complete set can be constructed [6]. Based on this, we demonstrate the XOR logic and its adoptions for symmetric encryption and decryption. Therefore, the work paves ways to future data storage security for lightweight applications.

Bi-mode Activation for Logical NOT

The typical structure of a DRAM-based array contains a matrix of DRAM cells, a row of sense amplifiers (SA), and a row of address decoder. SA relies on the positive feedback mechanism that amplifies a small potential difference between the cell-controlled bitline (BL) and the reference /bitline (/BL) (**Fig. 1 (a)**). However, such positive feedback mechanism is opposite to what is required for the logic NOT, in which the BL needs to be inversed (i.e. pull up to V_{DD} if it is 0 initially). To overcome this, we propose the *Bi-mode Activation*: the use of both normal and reverse operation of the SA to enable the bitwise NOT operation. Unlike prior work, this method can be applied without changing the memory cell, nor the core part of the peripherals [8-10].

A. Reverse Operation on Sense Amplifier (RSA)

For normal operation of SA, SAN and SAP are applied with 0 and V_{DD} respectively. When swapped (i.e. V_{DD} and 0 is applied on SAN and SAP), we call it the reverse operation of SA. **Fig. 1 (b)** shows this can balance the potential between BL and /BL, making them approach to their averaged potential. As shown in **Fig. 2**, assuming BL = $1/2V_{DD}$, /BL = 0 in the initial state and SA is in its reverse operation mode, the transistor T_{n2} and T_{p1} turns on. BL discharges through SAP and /BL charges through SAN until their difference become negligible. In addition, four extra transistors are added to precharge circuit to ensure the voltage independent writing to BL and /BL. When the precharge circuit is off, the CSI signal selects BL or /BL for independent charging.

B. Operation of Logical NOT

The NOT operation involves two memory cells using as the input and output, as shown in **Fig. 3**. The empty state and the full-charged state of the cell capacitor are used to represent '0' and '1' respectively. Without loss of generality, in the following, we illustrated our NOT operation using the input of '1'. Thus, Cap1 is fully charged before the logical NOT operation is performed.

Step 1: The standard PRECHARGE operation is performed first (❶) in which the wordline (WL) is lowered, the SA is disabled, and the potential of BL and /BL are both charged to $V_{DD}/2$ by the equalization circuit.

Step 2: The ACTIVATE operation is then applied on the first column and the WL of Cap1 is raised (❷). Through charge sharing with the fully-charged Cap1, the potential of BL rises up to $V_{DD}/2 + \delta$. The activated SA amplifies the potential difference between BL and /BL until BL is raised to V_{DD} and the corresponding /BL lowered to 0 (❸). Then WL of Cap1 and SA are both turned off.

Step 3: BL is independently charged to $V_{DD}/2$ (❹). Therefore, /BL can be maintained at 0. Then, SA is reversely enabled and the equalization of the voltages on BL and /BL is completed (❺). This is the key to achieve the logical NOT.

Step 4: /BL is independently charged to $V_{DD}/2$, while BL is kept as $V_{DD}/2 - \delta$ (❻). Since the potential in BL is lower than /BL, after SA is enabled normally, BL is pulled down to 0 while /BL is pulled up to V_{DD} (❼).

Step 5: Finally, the ACTIVATE operation is applied. The Cap2 is charged up through charge sharing with the BL. A logic '0' is stored in the cell, which equals to /Cap1 (❽), i.e. a logic NOT operation is achieved.

Then successful NOT operation is also achievable with the input logic '0'. The only difference is that the potential of BL becomes $V_{DD}/2 + \delta$ after **Step 3**. When comparing with /BL at $V_{DD}/2$, SA pulled up BL to V_{DD} at **Step 4**, and a logic '1' is written back to Cap2 at **Step 5**, completing the logic NOT operation.

C. Validation

We use Cadence Spectre to simulate this scheme based on the 45nm process library, and the waveform is shown in **Fig. 4**. The bitwise NOT operation of the input Cap1 is perfectly completed when Cap1 is logical '1'. In the figure, the voltage fluctuation of Cap1 in (2) and (3) is resulted from the ACTIVATE process, i.e., it participates in SA amplification after charge sharing with BL. In (4) and (6), the independently write operations to BL and /BL are completed successfully. When SA is turned on reversely in (5), it can be seen that the voltage difference between BL and /BL decreases. The potential of Cap2 in (8) changes sharply, which indicates that we have obtained logical '0' expectedly.

XOR- Based Image Encryption

DRAM generally has inherent vulnerabilities such as "RowHammer," which may pose a threat to data security [11]. Due to the effectiveness and simplicity, XOR logic is widely used for symmetric encryption [12]. Based on the NOT logic proposed in this work together with the predecessors' bitwise AND/OR operation, the XOR logic can be directly achieved [6]. Therefore, it is possible that the data storage and encryption/decryption can be performed in-situ within DRAM: By writing the original data (plaintext) and bitwise XORing with the private key, the plaintext can be stored and encrypted simultaneously. After receiving the ciphertext, the decryption process is similar with the same key. In addition, plaintext and key are overwritten directly during encryption/decryption (**Fig. 6 (a)**).

Fig. 5 shows a mapping scheme using the 8*15 DRAM array to encrypt and decrypt 5*3 pixel data. The plaintext (a_0-a_{14}) and the key (b_0-b_{14}) are placed in two rows respectively, and the ciphertext (d_0-d_{14}) is read from another row. During XOR calculation, the plaintext and key are corrupted directly, which provides enough security. All rows are calculated in parallel. The decryption process can also be completed through this mapping scheme. **Fig. 6 (b)** shows the plaintext in in-situ XOR encryption experiment based on DRAM. In **Fig. 6 (c)**, "S," "J," "T" and "U" cannot be identified, which proves that the image encryption is successful. Continue to perform the bitwise XOR operations on the encrypted data and the key, and the original "S," "J," "T," "U" graphics are restored, indicating that the data is successfully decrypted (**Fig. 6 (d)**).

Conclusion

This work proposes a new implementation of logical NOT based on unmodified DRAM cells and has passed the test verification of simulation. Moreover, the Bi-mode Activation method is successfully applied in XOR-based image encryption and decryption.

Acknowledgement

This work is partly supported by the National Key R&D Program of China (2019YFB2205005).

References

[1] H. Jee, et al., IJCSET, 2007. [2] A. K. Kurra, et al., IJEEI, 2019. [3] J. Sim, et al., ICCAD 2018. [4] S. Yu, et al., CICC, 2020. [5] H. Jiang, et al., IEEE Trans. Comput., 2019. [6] V. Seshadri, et al., MICRO, 2017. [7] F. Gao, et al., MICRO, 2019. [8] L. Jiang, et al., ISLPED, 2017. [9] Choi, H. et al., COOL CHIPS, 2020. [10] S. Li, et al., ACM, 2017. [11] A. Schaller et al., HOST, 2017. [12] Y. Song, et al., EDL, 2021.

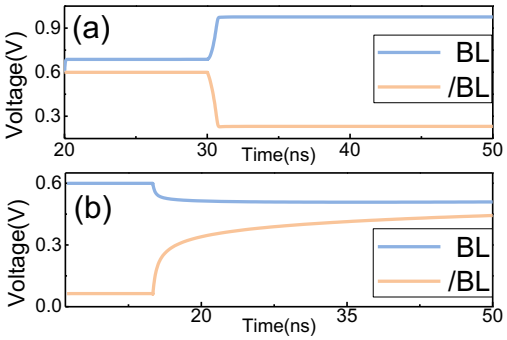


Fig. 1. (a) The waveform of normal operation on sense amplifier and (b) reverse operation on sense amplifier.

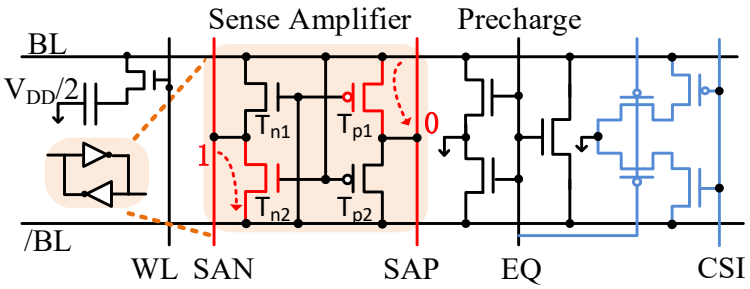


Fig. 2. Circuit for typical DRAM cell and peripherals while RSA is in progress. SAN and SAP are connected to logical '1' and '0' respectively. Four transistors (blue) are added to the precharge circuit for BLs' independent charging.

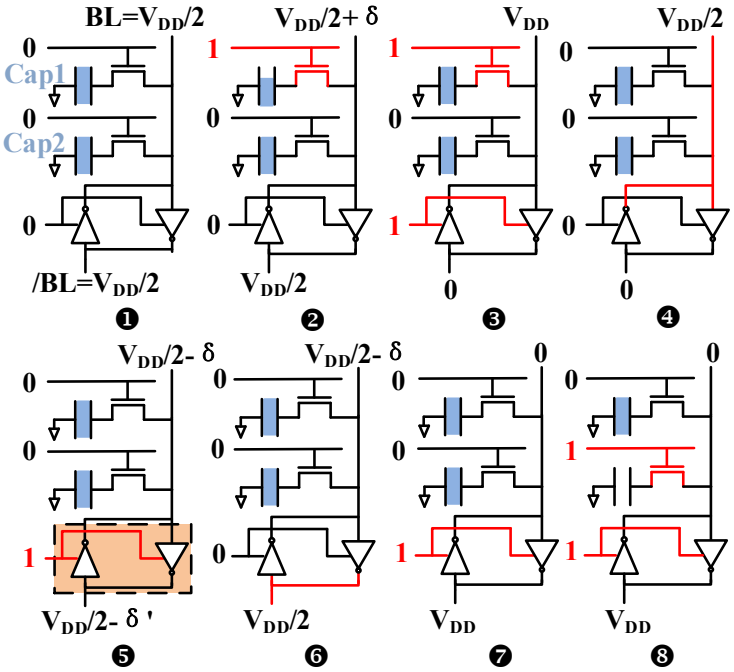


Fig. 3. The flowchart for implementation of logical NOT. Red represents the signal that changes during the operation. ❶PRECHARGE ❷~❸ ACTIVATE Cap1 ❹charge BL ❺RSA ❻charge /BL ❼enable SA normally ❽ACTIVATE Cap2.

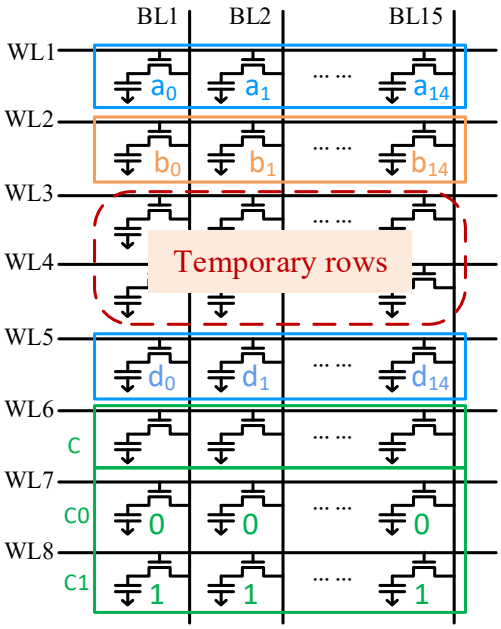


Fig. 5. A parallel mapping scheme for encryption and decryption of 5*3 pixel data. Two temporary rows are reserved for computing and operating row (C) is used to choose AND/OR operations.

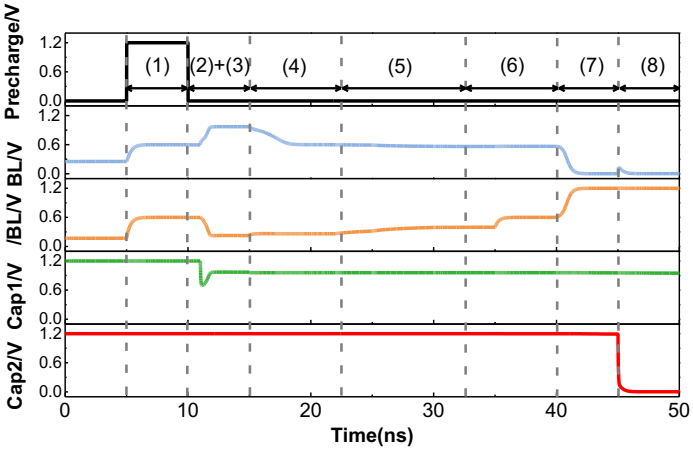
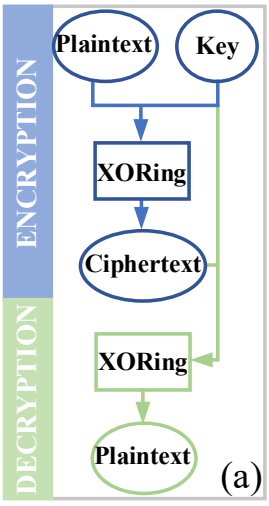


Fig. 4. Cadence simulation results based on 45 nm process library. (1) ~ (8) correspond to the operations in Fig. 3.



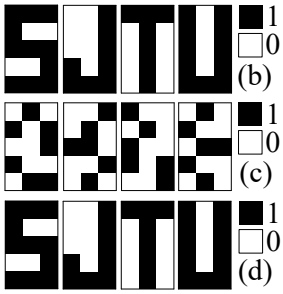


Fig. 6. (a) The process of encrypting and decrypting applications in DRAM. (b) Plaintext (c) Ciphertext (d) Decrypted data.