Fault Model Analysis of DRAM under Electromagnetic Fault Injection Attack

Qiang Liu*, Longtao Guo, Honghui Tang
School of Microelectronics, Tianjin University, Tianjin, China
Email: {qiangliu, guolongtao, hhtang}@tju.edu.cn

***Abstract*—Electromagnetic fault injection (EMFI) attack has posed serious threats to the security of integrated circuits. Memory storing sensitive codes and data has become the first choice of attacking targets. This work performs a thorough characterization of the induced faults and the associated fault model of EMFI attacks on DRAM. Specifically, we firstly carry out a set of experiments to analyse the sensitivity of various types of memory to EMFI. The analysis shows that DRAM is more sensitive to EMFI than EEPROM, Flash, and SRAM in this experiment. Then, we classify the induced faults in DRAM and formulate the fault models. Finally, we find the underlying reasons that explain the observed fault models by circuit-level simulation of DRAM under EMFI. The in-depth understanding of the fault models will guide design of DRAM against EMFI attacks.**

***Index Terms*—Electromagnetic fault injection, hardware security, DRAM**

I. Introduction

Integrated circuits (ICs) have been the backbone of various applications, such as telecommunications, industrial control, consumer electronics and military infrastructure. The information security of these applications significantly relies on ICs' security, including availability, confidentiality and integrity of information processed on ICs. Unfortunately, the emerging physical attacks pose serious threats to ICs' security.

The physical attacks on ICs can be classified into side channel attack (SCA) and fault injection attack (FIA). In SCA, the physical information, such as timing information, power consumption and electromagnetic, is collected from ICs and used to compromise security and privacy [1]. Unlike SCA, FIA is an active attack means, which induces faults into ICs by tempering with the operating conditions. Existing FIA techniques include clock glitch [2], voltage glitch [3], laser injection [4] and electromagnetic fault injection (EMFI) [5]. Among them, EMFI is non-invasive, and has high spatial resolution and low cost. Therefore, EMFI has drawn more attentions recently from both academia and industry.

Based on the information flow on an electronic system, EMFI has been applied to processors [6], interconnect/interface [7], and memory subsystems [8]. As modern processors' clock frequency becomes over GHz, the required temporal and spatial resolution for EMFI is hard to achieve [9]. Therefore, components around the processor in a system become the targets. By attacking these components, the security mechanism could be indirectly defeated, which is particularly called the second-order attack [9].

Especially, memory as a necessary component and storing sensitive codes and data, becomes the first choice of attacking targets. Laser fault injection attack was applied to static random-access memory (SRAM) and the induced faults were used to break the AES encryption algorithm implemented on ATmega163L microcontroller [8]. The flash memory of a 32-bit microcontroller was demonstrated to be sensitive to laser fault injection [10]. EMFI was applied to synchronous dynamic random-access memory (SDRAM) to skip the secure boot mechanism of a router [9] and to crack AES running in an ARM processor [11]. The existing physical attacks on memory mainly use laser for its high control resolution. However, the laser fault injection requires expensive and professional equipment. We believe that the low-cost EMFI will pose higher threat to the memory subsystems in practical applications. Nevertheless, comprehensive investigation of EMFI attacks on memory has not been carried out, and the characteristics and models of the induced faults are unclear.

Therefore, this work performs a thorough characterisation of the induced faults and the associated fault model of EMFI attacks on DRAM. Specifically, we firstly carry out the sensitivity analysis of various types of memory to EMFI by designing a set of experiments. The analysis shows that DRAM is more sensitive to EMFI. Then, we classify the induced faults in DRAM and formulate the fault models. Finally, we find the underlying reasons that explain the observed fault models by circuit-level simulation of DRAM under EMFI.

The main contributions of this work are:

- We experimentally analyze the sensitivity of various memories to real EMFI attacks and find that the tested DRAM is the most sensitive one.
- We classify the induced faults in DRAM into four categories, in terms of address patterns, and describe the fault models.
- We carry out circuit-level simulation of DRAM storage cell and find the underlying fault mechanism which explains the observed fault models.

The rest of paper is organized as follows. Section II briefly introduces the principle of EMFI and recently reported EMFI attacks. Section III presents practical EMFI attacks on four types of memories and analyzes the sensitivity. Section IV focuses on investigation of the fault models of DRAM. Section V carries out DRAM circuit simulation to reason the fault models. Section VI proposes several design countermeasures based on the fault models. Finally, Section VII concludes the paper with future works.

978-3-9819263-7-8/DATE23/© 2023 EDAA

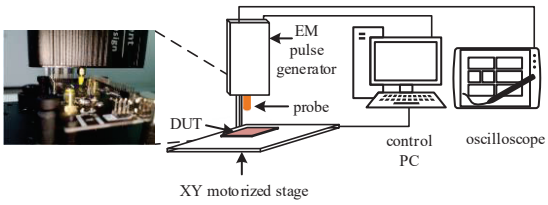


Fig. 1: Diagram of EMFI platform.

II. Background

EMFI is based on the faraday's law of electromagnetic (EM) induction, in which an electromotive force is induced in a wire loop when the magnetic field crossing the surface of loop changes. In ICs, the power and ground grids contain many vertical and horizontal loops, which could be affected by the EM disturbances [12].

In recent years, successful EMFI attacks on microprocessors have been seen. The attacks have led to instruction skip [7], encryption algorithm cracking [5], and sensitive information leakage such as address and instruction [6]. It is also shown that instruction buffer of Cortex-M4 was easily to be disrupted by EMFI [13]. In addition, EMFI was also performed on a smartphone-grade 64-bit SoC [14].

As working frequency increases, the temporal and spatial resolution required for successful EMFI to processors is difficult to achieve. Therefore, the second-order EMFI attack [9] was proposed. The main idea is that EMFI is applied to the components interacted with the processors, and the induced faults in the components are propagated to indirectly generate faults in the processors. In [9], a two-stage attack was performed. In the first stage, EMs were injected into the DRAM chip to corrupt data, causing an illegal execution of the debug interface. In the second stage, the attacker used the debug interface to load a binary that bypasses the secure boot within the trusted execution environment. In [11], the AES encryption algorithm running in an ARM processor was cracked. The persistent faults were injected into an SDRAM chip where the instructions and data (including S-box) were stored. By analyzing the multiple persistent faults induced in S-box, the key of the AES algorithm was cracked. Although successfully performing second-order EMFI attack on memories, the existing works do not investigate the associated fault models. We believe that understanding the fault models will facilitate the attack and more importantly provide guidance in countermeasure design. Therefore, this work makes deep investigation into the fault model and the underlying fault mechanism.

III. Sensitivity analysis

A. EMFI platform

The EMFI platform built in this work includes an electromagnetic pulse (EMP) generator with a probe, a low-cost computerized numerical control (CNC) machine, an oscilloscope and a PC. The module number of the EMP generator is OSR-EM-U-100 while the CNC machine is FSL40E150-05C7. The EMP generator can generate pulses with amplitude from -400 V to +400 V and with width from 10 ns to 200 ns. The probe consists of a permanent magnet and 13 turns of wound wire, and the diameter is 1.4 mm. The CNC machine is used to hold the device under test (DUT) and enable accurate XY placement with a resolution of 50 um and a stroke of 150mm. The EMP generator, the CNC machine and the target board are controlled by the PC. The diagram of the platform is shown in Fig. 1.

In the experiment, the distance between DUT and probe tip is about 1 mm. The DUT is scanned with a step of 1 mm which is less than the diameter of probe. At each position, up to 100 EMPs with the width increasing from 20 ns to 200 ns in a step of 20 ns and the trigger delay increasing from 200 ns to 4 us in a step of 200 ns, are injected into the DUT. If an error is induced or 100 EMPs are injected at a position, the probe moves to the next position.

TABLE I: EMFI on various memories.

Type	Chip information	Error rate
EEPROM	ATMEL AT24C04, 4Kb	0
Flash	Winbond W25Q128JV, 128Mb	0.08%
SRAM	ISSI IS62WV12816ALL, 2Mb	0.05%
DDR2 SDRAM	MIRA P2R10E4KGF, 128MB	0.28%
DDR3 SDRAM	Micro MT41K128M16JT, 256 MB	0.48%

B. Memory targets and experiment design

In current electronic systems, four types of memory are widely used, including DRAM, SRAM, flash memory and EEPROM. These memories have different working principle and circuit structures. We obtain off-the-shelf chips of the four types of memory as the DUT in the experiment. The information of the used memory chips are shown in Table I.

The attack experiment is designed as follows. Each memory chip is accessed by a microprocessor, which is connected to a PC. The microprocessor and memory are two chips and only the memory chip is under attack. A program runs in the microprocessor. Firstly, the program randomly initializes a set of data and writes the data into the memory chip before attack. Then, the EMFI attack is performed to the memory chip, and at the same time the program reads the data from the memory chip and compares the read data with the initialized data. When a mismatch (i.e., an error) is found, the program sends the erroneous data, its address and the correct data to the PC for analysis. After the whole memory chip is scanned, the PC analyses results.

C. Sensitivity of various memories to EMFI

After collecting all attack results, we analyze the sensitivity of the memory chips to the EMFI in terms of error rate (ER). ER is defined as the number of errors over the number of fault injections. ER evaluates the easiness of inducing errors in a memory chip by EMFI.

From Table I we can observe the following points. First, EMFI successfully induces errors in the flash, SRAM and DRAM chips. This result demonstrates that EMFI actually poses threats to the memories. Second, among the memory chips with errors, the two DRAM chips are the most sensitive to

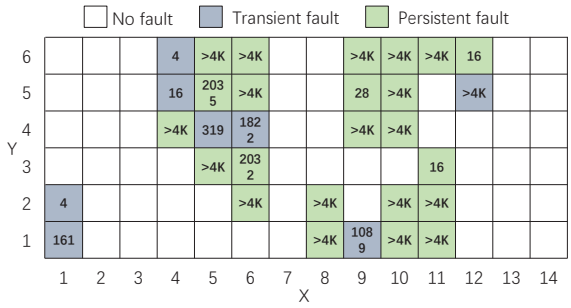


Fig. 2: EMFI attack results on DDR3 SDRAM.

EMFI. ERs of the two DRAM chips are one order of magnitude higher than that of the other two chips.

Therefore, we decide to dive into the DRAM to find out the fault model and mechanism. The fault models of the other three types of memory are left for future study.

IV. DRAM FAULT MODEL ANALYSIS

The sensitivity analysis in the above section shows the result that DRAM is more sensitive to EMFI attacks. To investigate fault models, we first characterize the induced faults. Then, we perform circuit-level simulation of DRAM to find out the reasons that could explain the observed fault models.

A. Fault model analysis

We first carry out a thorough characterization of EMFI attack on DRAM. Here, we use the DDR3 SDRAM in Table I as an example. Fig. 2 shows the EMFI attack results on the DDR3 SDRAM chip. The chip is divided into a 14×6 grid. The digit in each position of the grid indicates the number of faulty bytes per injection at the position. For example, the EMFI at position (1,1) leads to 161 erroneous bytes. "$>$4K" in a position means that the number of erroneous bytes is over 4096. Note that 4K is double the page size of the SDRAM chip. The exact number is not recorded because transfer of those bytes from the board to PC requires a large amount of time.

The attack results can be classified into three cases, including no fault, transient fault and persistent fault. No fault means that the injections in the locations do not induce errors in the SDRAM. Transient fault means that the induced errors only last one or few clock cycles. Persistent fault means that the induced errors exist until the corresponding address is written again. From Fig. 2 we can see that (1) all memory faults induced by EMFI are multiple bytes, and (2) the number of persistent faults is significantly larger than the number of transient faults.

By analyzing the memory addresses where the errors are induced, we actually find out the regularities of some faults and establish four fault models, including address offset persistent fault, column persistent fault, region persistent fault and region transient fault.

a) Address offset persistent fault: We find that the faults in positions (6,2), (10,2) and (10,6) in Fig. 2 have a common pattern that reading the value of address A returns the value of address $A + \delta_i$ persistently. Therefore, we call this type of fault as address offset persistent fault. The offset δ_i depends on the injection positions. Also, such faults appear in consecutive addresses as shown in Fig. 3.

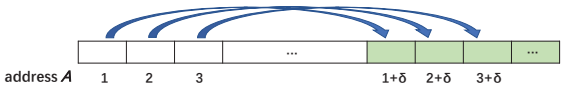


Fig. 3: Address offset persistent fault occurring in consecutive addresses.

b) Column persistent fault: As indicated in the name, the faults injected in positions (6,6) and (10,5) lead to errors in certain columns of the SDRAM and the errors in a column are the same.

c) Region transient/persistent fault: Because the difference between the region transient and persistent faults is only the last time of errors, we describe both models together. A region fault means that EMFI at a position leads to errors in a small region of the SDRAM chip. Fault injection in positions (1,1), (1,2), (4,5) and (4,6) leads to transient errors in four ranges of memory addresses. For example, injection in (1,1) induces errors in addresses from 965676 to 965679. Note that the addresses are not always consecutive. Similarly, EMFI in (9,5) (11,3) and (12,6) leads to persistent errors in certain regions of the memory.

The above fault models extract the features of some DRAM faults. With the fault models, one can understand the EMFI attack on DRAM well and demonstrate practical applications in compromising the security of a computing system as [11]. Investigation on the fault model eventually aims at enhancing the security of the hardware system. To develop countermeasure for DRAM against EMFI, we need to understand the reasons which result in the fault models. Therefore, we carry out circuit-level simulation of DRAM and try to find in which possible parts of the underlying circuit EMFI could affect the correct operations.

V. DRAM FAULT MODEL REASONING

A. Simulation setup

We use HSPICE2009.03-SP for DRAM circuit simulation, and Avanwave2009.03 to display the simulation results. The input files of simulation include the circuit netlist and TSMC 130nm process library. The simulated DRAM storage cell is a basic 1 transistor-1 capacitor (1T-1C) structure, shown in Fig. 4 (a). In the structure, 1-bit data is stored in capacitor C_s and transistor N_0 controls the charge and discharge of C_s. Reading/writing data is controlled by the word line WL_0 and bit line BL. In the simulation, the transistor size is 130nm width and 160nm length, which is the minimum size under this process. The VDD voltage is 3V. C_s and $!C_s$ both have 60fF capacitance while C_{BL} and $!C_{BL}$ have 600fF capacitance, which is ten times of C_s.

The timing sequence of reading '1' stored in the cell is shown in Fig. 4 (b). The sequence is comprised of four stages: pre-charging, activation, sensing&amplification and refreshing. In the pre-charging stage, PEQ is pulled to VDD, making BL and $!BL$ to V_{ref1} which is usually VDD/2. In the activation

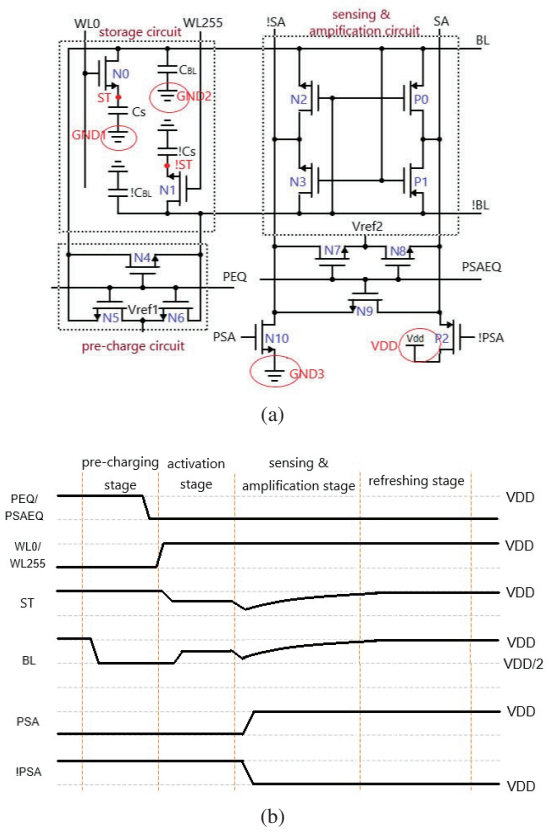


Fig. 4: (a) Basic circuit structure of 1T-1C DRAM cell and (b) timing sequence of reading '1'.

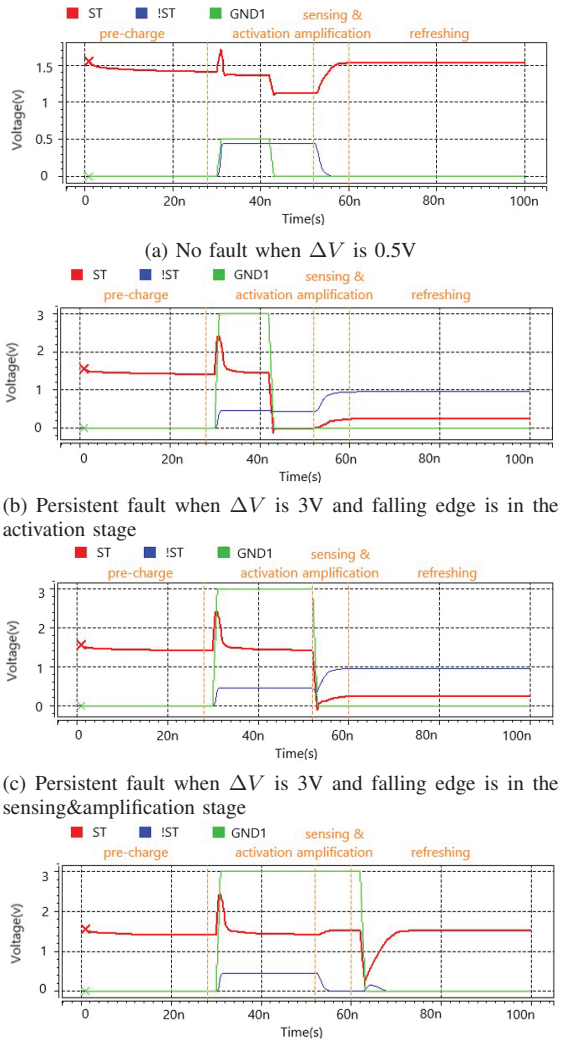
stage, WL_0 and WL_{255} are asserted, and C_s and $!C_s$ are discharged and charged respectively, leading to the increase and decrease of the voltages of BL and $!BL$. Because C_{BL} is ten times of C_s, the voltage variations of BL and $!BL$ are small. In the sensing&amplification stage, the voltage variations of BL and $!BL$ are amplified to VDD and GND. In the refreshing stage, the amplified voltage is written into C_s to ensure stable '1'. The writing operation has the similar timing sequence.

We simulate EMFI during the reading operation. As mentioned earlier, EMFI mainly disturbs the power and ground grids. In the 1T-1C circuit structure we identify four potential nodes GND_1, GND_2, GND_3 and VDD as shown Fig. 4 (a). At each node, a disturbance pulse with different amplitudes ΔV, timings and polarities is added to mimic the effect of EMFI during the simulation, respectively. Because C_s/$!C_s$ is the basic storage cell and during reading N_0/N_1 is always on, we mainly analyze the variations of ST/$!ST$, *i.e.,* the voltage level of C_s/$!C_s$.

In the next, we analyze the simulation waveform of each case and search for the reasons behind the fault models proposed in the previous section.

B. Simulation result analysis and fault reasoning

a) Injection to GND_1: We first add a pulse with various amplitudes ΔV, timings and polarities to GND_1 and the simulation results are shown in Fig. 5 and Fig. 6, respectively. The red and blue lines indicate the voltage levels of ST and $!ST$. The green line is GND_1 with the disturbance pulse.



(a) No fault when ΔV is 0.5V

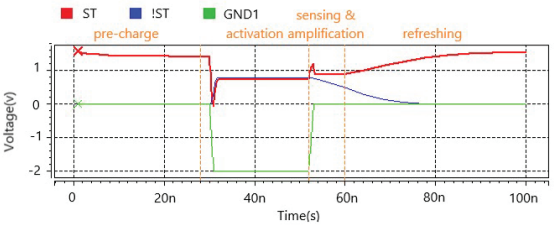
(b) Persistent fault when ΔV is 3V and falling edge is in the activation stage

(c) Persistent fault when ΔV is 3V and falling edge is in the sensing&amplification stage

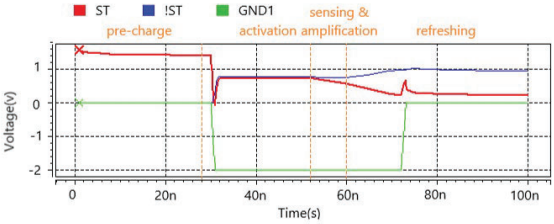
(d) Transicent fault when ΔV is 3V and falling edge is in the refreshing stage

Fig. 5: Simulation of injections with positive polarity to GND_1 when reading '1'.

In Fig. 5 (a), when $\Delta V = 0.5$V, the circuit works correctly, although the waveform of ST slightly affected by the rising and falling edges of the disturbance pulse. In (b), when ΔV increases to 3V, ST drops quickly along the falling edge of the pulse and remains 200mV. Compared to the timing in Fig. 4 (b), this shows a fault and results in output '0'. The reason of flipping from '1' to '0' is the following. After ST drops, both BL and $!BL$ are low voltage level, leading to P_0 and P_1 switching on. Then, when entering the sensing&amplification stage, P_2 and N_{10} are switched on, and due to connecting VDD, BL and $!BL$ start to increase. Because the voltage level of $!BL$ is slight higher than BL, N_2 is switched on before N_3, making BL connect to GND_3. This finally keeps ST low voltage level. Because ST remains low, it is a persistent fault. After a set of simulation we find that the threshold of ΔV is about 2V to generate a fault. Note that the threshold is obtained based on our experiment setup. When the circuit parameters and process library change, it could be different. In (c), when the falling edge is in the sensing&amplification stage, the similar



(a) No fault when ΔV is -2V and rising edge is in the sensing&amplification stage



(b) Persistent fault when ΔV is -2V and rising edge is in the refreshing stage

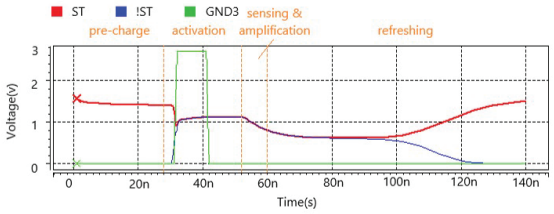
Fig. 6: Simulation of injections with negative polarity to GND_1 when reading '1'.

fault also occurs. In (d), the falling edge occurs in the refreshing stage, and we can see the sharp drop of ST along the falling edge, but it rises shortly and reaches to the normal voltage. Therefore, this case shows a transient fault. The recovery of ST is caused by the following fact. Before the falling edge in the refreshing stage, $!BL$ connects to GND_3 and BL connects to VDD. The falling edge leads to a drop of ST, but BL could pull ST back shortly.

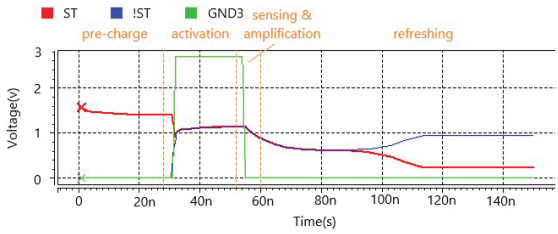
In Fig. 6 (a), a pulse with negative polarity ($\Delta V = -2V$) is added and the rising edge is in the sensing&amplification stage. Although having small variation, ST reaches to the correct level and there is not a fault. In (b), when the rising edge is in the refreshing stage, a persistent fault occurs.

In summary, with sufficient strength and right timing, the disturbance pulse with positive and negative polarities injected to GND_1 can induce persistent and transient faults. Because N_0 is on, disturbance in GND_2 will also affect ST through C_{BL}. We can observe the similar results as GND_1. Therefore, the simulation results are not shown here due to the page limit. Because each ground rail usually connects to several circuit cells placed in a region, *injection to GND_1/GND_2 could lead to the region transient/persistent faults*.

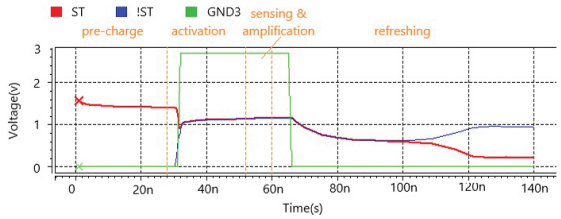
b) Injection to GND_3: We follow the similar procedure to simulate the effect of disturbance injected to GND_3. Fig. 7 shows the results. We keep the strength of the disturbance and vary time when the pulse starts to fall. In (a), the pulse falls in the activation stage and there is not fault induced. In (b), the pulse falls in the sensing&amplification stage and a persistent fault is induced. In (c), the falling edge is in the refreshing stage and also a persistent fault is induced. The reason of fault induction could be explained as follows. The rising edge of the pulse in the activation stage causes increase of the level of $!SA$ and $!BL$ which connects to the former through N_3. Then $!BL$ with high level switches N_2 on and makes BL connect to $!SA$. In the sensing&amplification and refreshing stages, N_{10} and P_2 are switched on, and the falling edge of the pulse results in the



(a) No fault when falling edge is in the activation stage



(b) Persistent fault when falling edge is in the sensing&amplification stage



(c) Persistent fault when falling edge is in the refreshing stage

Fig. 7: Simulation of injections to GND_3 when reading '1'.

decrease of $!SA$, BL and $!BL$. When BL decreases faster than $!BL$, P_1 is switched on and N_3 is switched off so that $!BL$ connects to SA which is VDD at the moment. This in turn keeps BL connect to $!SA$ which connects GND_3. Therefore, ST gradually decreases and finally remains low level. Note that, $!SA$ and SA connect to all DRAM cells in a column. As a result, *injection to GND_3 could lead to column persistent faults*.

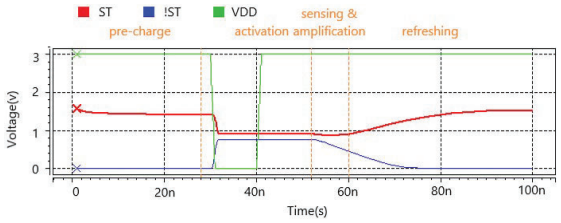
c) Injection to VDD: Fig. 8 shows the simulation results of various disturbance pulses added to VDD. From the figure we do not observe any faults.

Through the above circuit-level simulation, we can conclude that injected disturbances in the ground grid could induce column persistent fault, region persistent fault and region transient fault. Address offset persistent fault does not occur in the simulation experiment. We think this is because we only simulate the memory cell circuit and the fault is highly possible to be induced in the address decoder of DRAM. The above analysis not only shows the possibility of fault induction, but also the reasons of fault generation. By understanding the fault reason/mechanism, we propose several potential design countermeasures for DRAM against EMFI attacks.

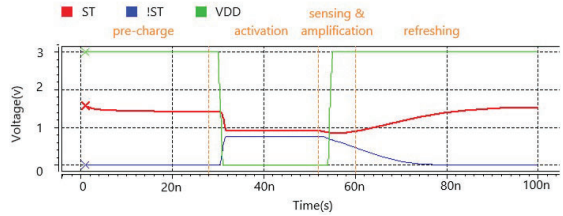
VI. Potential Design Countermeasures

Based on the fault reasoning above, we propose several possible solutions to improve the immunity of DRAM storage cell to EMFI.

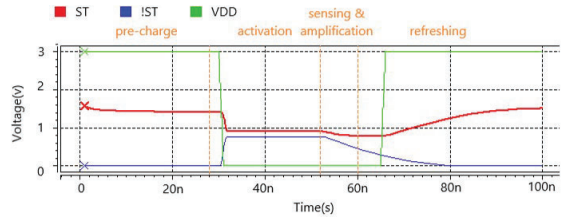
The first solution is to increase C_{BL}, which could stabilize the voltage level of ST. We test this by increasing C_{BL} from



(a) No fault when rising edge is in the activation stage



(b) No fault when rising edge is in the sensing&amplification stage



(c) No fault when rising edge is in the refreshing stage

Fig. 8: Simulation of injections to VDD when reading '1'.

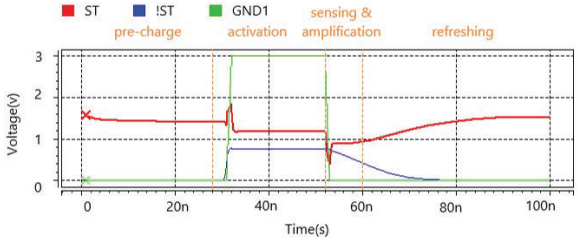


Fig. 9: No fault when increasing C_{BL}.

600fF to 700fF and simulating the circuit. A pulse, who induces a fault in Fig. 5 (c), is added in GND_1. The result in Fig. 9 shows that no fault occurs in this case. The result simply demonstrates the possibility of the solution. However, a design exploration is needed to find the right value of C_{BL} to increase security while not degrading the memory performance.

The second possible solution is to use a lowpass filter beside the GND paths to filter out the disturbances.

The third possible solution is to reduce the period of the activation and sensing&amplification stages. This is because the edges of disturbance pulse occurring in these two stages are easy to induce faults. By reducing the period of both stages, the fault probability could be reduced.

VII. Conclusion

This work investigates fault model and mechanism of DRAM under EMFI attacks. Four categories of fault models are observed in real EMFI attacks on DRAM chips. Circuit-level simulation of 1T-1C structure of DRAM bit reveals the reasons of fault induction. The in-depth investigation help us to understand the fault models of DRAM under EMFI attacks and indicates the directions of countermeasure designs against EMFI. In future we would like to further investigate the effects of different design parameters of circuit structure and manufacture process on immunity of DRAM against EMFI.

Acknowledgment

This work was supported by the National Nature Science Foundation of China under Grant 61974102.

References

[1] R. Wang, H. Wang, E. Dubrova, and M. Brisfors, "Advanced Far Field EM Side-Channel Attack on AES," in *Proceedings of the 7th ACM on Cyber-Physical System Security Workshop*, 2021, pp. 29–39.

[2] D. Zhou, P. Yang, and Q. Ou, "Analysis of Fault Characteristics Based on Clock Glitch Injection," in *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, vol. 5, 2021, pp. 785–790.

[3] C. Bozzato, R. Focardi, and F. Palmarini, "Shaping the Glitch: Optimizing Voltage Fault Injection Attacks," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 199–224, 2019.

[4] J. Breier, D. Jap, and C.-N. Chen, "Laser-based Fault Injection on Microcontrollers," in *Fault Tolerant Architectures for Cryptography and Hardware Security*, 2018, pp. 81–110.

[5] Liao, Haohao, "Electromagnetic Fault Injection on Two Microcontrollers: Methodology, Fault Model, Attack and Countermeasures," 2020.

[6] M. A. Elmohr, H. Liao, and C. H. Gebotys, "EM Fault Injection on ARM and RISC-V," in *2020 21st International Symposium on Quality Electronic Design (ISQED)*. IEEE, 2020, pp. 206–212.

[7] A. Menu, S. Bhasin, J.-M. Dutertre, J.-B. Rigaud, and J.-L. Danger, "Precise spatio-temporal electromagnetic fault injections on data transfers," in *2019 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*, 2019, pp. 1–8.

[8] F. Zhang, Y. Zhang, H. Jiang, X. Zhu, S. Bhasin, X. Zhao, Z. Liu, D. Gu, and K. Ren, "Persistent fault attack in practice," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2020, Issue 2, pp. 172–195, 2020. [Online]. Available: https://tches.iacr.org/index.php/TCHES/article/view/8548

[9] A. Cui and R. Housley, "BADFET: Defeating modern secure boot using Second-Order pulsed electromagnetic fault injection," in *11th USENIX Workshop on Offensive Technologies (WOOT 17)*. Vancouver, BC: USENIX Association, Aug. 2017. [Online]. Available: https://www.usenix.org/conference/woot17/workshop-program/presentation/cui

[10] B. Colombier, A. Menu, J.-M. Dutertre, P.-A. Moëllic, J.-B. Rigaud, and J.-L. Danger, "Laser-induced single-bit faults in flash memory: Instructions corruption on a 32-bit microcontroller," in *2019 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, 2019, pp. 1–10.

[11] H. Tang and Q. Liu, "MPFA: An efficient multiple faults-based persistent fault analysis method for low-cost FIA," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 9, pp. 2821–2834, 2022.

[12] M. Dumont, M. Lisart, and P. Maurine, "Modeling and simulating electromagnetic fault injection," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, no. 4, pp. 680–693, 2021.

[13] O. Trabelsi Ltci, L. Sauvage Ltci, and J.-L. Danger Ltci, "Characterization of electromagnetic fault injection on a 32-bit microcontroller instruction buffer," in *2020 Asian Hardware Oriented Security and Trust Symposium (AsianHOST)*, 2020, pp. 1–6.

[14] C. Gaine, D. Aboulkassimi, S. Pontié, J.-P. Nikolovski, and J.-M. Dutertre, "Electromagnetic fault injection as a new forensic approach for socs," in *2020 IEEE International Workshop on Information Forensics and Security (WIFS)*, 2020, pp. 1–6.