Hardware Trojans in Emerging Non-Volatile Memories

Mohammad Nasim Imtiaz Khan
School of EECS
The Pennsylvania State University
University Park, PA, USA
muk392@psu.edu

Karthikeyan Nagarajan
School of EECS
The Pennsylvania State University
University Park, PA, USA
kxn287@psu.edu

Swaroop Ghosh
School of EECS
The Pennsylvania State University
University Park, PA, USA
szg212@psu.edu

***Abstract*—Emerging Non-Volatile Memories (NVMs) possess unique characteristics that make them a top target for deploying Hardware Trojan. In this paper, we investigate such knobs that can be targeted by the Trojans to cause read/write failure. For example, NVM read operation depends on clamp voltage which the adversary can manipulate. Adversary can also use ground bounce generated in NVM write operation to hamper another parallel read/write operation. We have designed a Trojan that can be activated and deactivated by writing a specific data pattern to a particular address. Once activated, the Trojan can couple two predetermined addresses and data written to one address (victim's address space) will get copied to another address (adversary's address space). This will leak sensitive information e.g., encryption keys. Adversary can also create read/write failure to predetermined locations (fault injection). Simulation results indicate that the Trojan can be activated by writing a specific data pattern to a specific address for 1956 times. Once activated, the attack duration can be as low as 52.4μs and as high as 1.1ms (with reset-enable trigger). We also show that the proposed Trojan can scale down the clamp voltage by 400mV from optimum value which is sufficient to inject specific data-polarity read error. We also propose techniques to inject noise in the ground/power rail to cause read/write failure.**

***Index Terms*—Hardware Trojan, Memory Trojan, Trigger, Payloads, Information Leakage, Fault Injection, DoS.**

I. Introduction

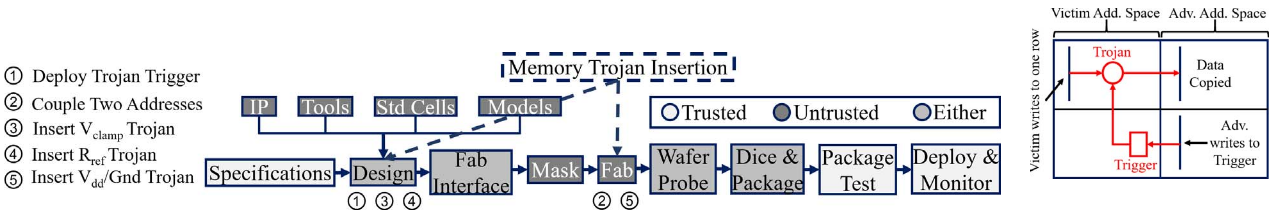
Hardware Trojan [1] is a malicious and dormant modification of a circuit which can cause a chip to perform undesirable operations. Adversaries can modify the circuit in different phases of chip manufacturing (Fig. 1a) due to outsourcing to different untrusted third parties for cost benefit [2], [3]. The Trojan is carefully designed so that it only gets activated under certain conditions. Therefore, an intelligently designed Trojan remains undetected during testing.

Many prior works have investigated possible hardware Trojans. In [4], the authors have proposed a Trojan in embedded SRAM which evades industry standard post-manufacturing memory tests (for example, March test). They demonstrate various forms of Trojan transistors in SRAM that gets turned ON by a unique pattern stored in other memory addresses. Those transistors, when turned ON, shorts the data node of a victim cell with ground (causes data corruption). Although, the work employs data pattern which is not tested by conventional tests, there still lies a probabilistic possibility of activating the Trojan during testing phase. Furthermore, the Trojan proposed in [4] is applicable to charge-based memories e.g., SRAM and DRAM since the charge stored at a data node can be used to turn ON the malicious transistors. However, these Trojans will not work on Non-Volatile Memories (NVMs) which use cell resistance/material magnetization to store data. In [5], an analog Trojan trigger is presented which is controllable, stealthy and small. The trigger proposed in this work employs a technique similar to the one presented in [5] (i.e. charging a capacitor). However, the novelty of the proposed trigger lies in the activation mechanism i.e., writing a specific memory address with a specific data pattern for a certain number of times (inputs of the Trojan). The proposed trigger can also be reset to control the attack duration. The reset signal can be generated by a reset-trigger circuit (similar to Trojan trigger) which can be activated by writing a different address with a specific data pattern for a certain number of times.

Attack Model: Hardware Trojan is composed of two parts: Trigger and Payload [1], [2]. Therefore, in this work, first we propose a Trojan trigger circuit by leveraging high NVM write current. It has been pointed out in [6], [7] that NVM write current can cause data polarity-dependent supply noise. An adversary can write a specific address with a specific data pattern to generate a specific ground bounce [6], [7] which can be used to charge a capacitor incrementally. Trojan trigger will be activated if the capacitor is charged to a certain threshold by repeated writing (say, N times). In a more complex attack, the adversary can also use the actual data pattern written to the specific address by tapping the data bus to charge up the capacitor. If N is a high number, it is unlikely that the Trojan will be triggered/detected during testing phase.

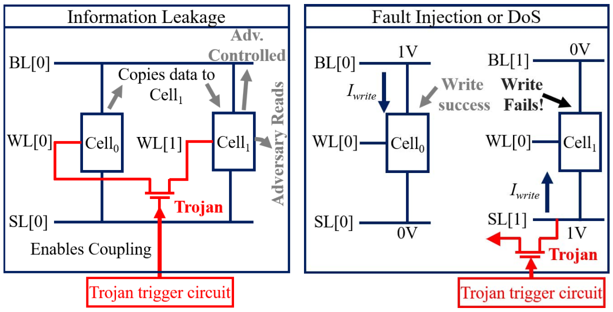
We propose three Trojan payloads once it is triggered: (a) Payload 1 couples two predetermined memory addresses. Therefore, data will be copied to the adversary-controlled address when the data is written to a victim's address (Fig. 1b); (b) Payload 2 manipulates clamp voltage to launch read failure since NVM read Sense Margin (SM) depends on this parameter; (c) Payload 3 injects noise to cause read/write failure by generating ground bounce from a parallel operation and/or shorting the ground (V_{dd}) rail with a voltage source (ground) via a Trojan transistor (since slight disturbance in the ground or power rail can increase the write latency or decrease the read SM).

978-3-9819263-2-3/DATE19/©2019 EDAA



(a) (b)

Fig. 1: (a) Semiconductor supply chain showing trusted, untrusted and both trusted/untrusted steps [3]. Design/Fab can be leveraged to insert memory Trojan; (b) cartoon showing the concept for memory Trojan which can leak information.



(a) (b)

Fig. 2: Malicious memory Trojan causing, (a) information leakage attack; and, (b) fault injection or DoS attack. Writing predefined address with predefined data works as the trigger causing data copy or read/write failure as the payload.

In this work, we have mainly assumed that the attacker is the actual designer. The attacker can also be present in fabrication house if the modification is minor. For example, Trojans 1, 3 and 4 proposed in this work (Fig. 1a) can be introduced by the malicious designer while Trojans 2 and 5 can be introduced by both the designer and the fabrication house. After the deployment of the chip in the market, the adversary can launch a malicious program to trigger the Trojan for the desired payload. We mainly focus on one flavor of NVM namely, Spin-Transfer Torque RAM (STTRAM) for the sake of brevity. The objectives of memory Trojan can be:

(a) Information Leakage: The overview of this case is shown in Fig. 1b and bit-level example is given in Fig. 2a. Fig. 2a assumes that victim and adversary have access to only $Cell_0$ and $Cell_1$ respectively. These two addresses share the same Bitline ($BL[0]$) and Sourceline ($SL[0]$) and wordlines (WLs) are coupled through a Trojan transistor. If the Trojan is activated, the data will be copied to $Cell_1$ whenever victim writes to $Cell_0$. The adversary can read $Cell_1$ and get victim's write data.

(b) Fault Injection: The Trojan can target memory addresses to prevent writing one particular data polarity (either 0→ 1 or 1→ 0) while the other one passes. In Fig. 2b, we see that 0→ 1 writing to $Cell_1$ fails since the voltage headroom between BL and SL is not sufficient. However, writing 1→ 0 to $Cell_0$ is successful. Such kind of fault injection can leak system assets such as, keys. One example is setting plaintext to all 0 or 1 by injecting fault which makes ciphertext (that is sent out) same as keys (since in simple XOR encryption, ciphertext = plaintext ⊗ key), and can be recovered by the adversary. Note that, reduced voltage headroom can also cause read SM degradation and lead to data polarity-specific read error.

(c) Denial of Service (DoS): If Trojan targets both writes (1→ 0 and 0→ 1) and causes failure, it results in a complete write failure i.e. Denial of Service (DoS).

Evading Trojan Detection: Trojan detection techniques have been proposed using sophisticated failure analysis like Light-Induced Voltage Alternation (LIVA), Charge Induced Voltage Alternation (CIVA) and other imaging techniques. However, these methods require significant time/effort (requires chip delayering) and are not highly effective for nanometer technologies [8]. Two other techniques are proposed in [8] namely, Automatic Test Pattern Generation (ATPG) and Side Channel Analysis (SCA). ATPG does not work for logic Trojan where the malicious inserted logic is unknown [8]. Therefore, it cannot detect the proposed trigger. It might be possible to trigger the proposed Trojan by writing each address with all possible combination for many times (1956 for this work). However, this increases the test time and time to market the chip significantly. Typically, each chip is tested for 2-3 secs [9] which is not enough to catch such Trojans. Furthermore, the Trojan becomes easier to deploy effectively if the designer itself is the adversary. SCA is also ineffective against the proposed memory Trojan since the trigger only consumes dynamic power when it is activated/deactivated. Although, the payload proposed in this work, shows significant power consumption (e.g., write current is doubled due to copy operation), the probability of triggering the Trojan during testing is considerably low.

In particular, we have made the following contributions: We, (a) propose Trojan trigger circuit which evades testing phase but gets activated by leveraging ground bounce caused by NVM high write current; (b) propose circuitry to reset the trigger and control the attack duration; (c) show that two memory addresses can be coupled together and sensitive data can be copied from one address to another; (d) investigate read and write operation of emerging NVMs for possible vulnerabilities which can be leveraged (e.g., clamp voltage and supply noise) to deploy memory Trojan; (e) propose techniques for Trojan payloads such as read/write failure, fault injection and DoS.

The rest of the paper is organized as follows: Section II describes emerging NVM basics; Section III proposes the Trojan and information leakage payload; Section IV describes the read/write failure payloads; Section V presents conclusion.

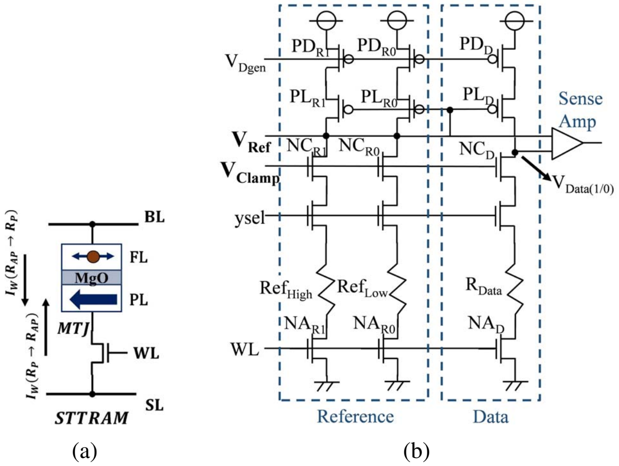


Fig. 3: STTRAM (a) bitcell and (b) read circuitry [10].

II. Basics of Emerging NVMs

A. STTRAM

STTRAM cell (Fig. 3a) contains one Magnetic Tunnel Junction (MTJ) as the storage element which contains a free (FL) and a pinned (PL) magnetic layer. The resistance of the MTJ stack is high (low) if FL magnetic orientation is anti-parallel (parallel) compared to the PL. MTJ can be toggled from parallel (P) (data '0') to anti-parallel (AP) (data '1') (or vice versa) using current induced Spin-Transfer Torque by passing the appropriate write current (> critical current) from source-line to bit-line (or vice versa).

B. Polarity Dependent Supply Noise in NVMs

Extremely high current (∼50-100mA, assuming ∼100μA/bit) is drawn for writing a full NVM memory word (512-1024bit). When the huge charge (due to write current) is dumped to the local ground (implemented in lower metal layer e.g., $Metal_1$) of the memory, the voltage of that local ground bounces [6]. This bounce is a function of write data pattern and can be as high as $350mV$ [6].

C. Read Circuitry

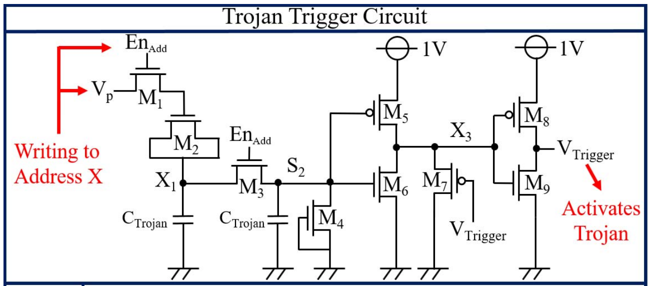
STTRAM suffers from poor Tunnel Magneto-Resistance (TMR). Low TMR causes low SM which may lead to read error under process variation. Therefore, STTRAM requires special read circuity. The read circuit [10] used in this work is shown in Fig. 3b. The voltage source V_{Dgen} and V_{Clamp} are DC sources. Signal $ysel$ connects the bitline with the bitcell and WL enables the bitcell for read. The circuit has two reference legs containing reference cells with resistance equivalent to Ref_{High} and Ref_{Low}. V_{Clamp} is selected from a resistance ladder based on the test results after the chip is manufactured. Therefore, V_{Clamp} modification could be a vulnerability.

III. Proposed Trojan Trigger and Information Leakage Payload

A. Trigger Circuit

Trigger Design: The trigger circuit (Fig. 4) is designed to get activated if a particular memory address (chosen during design phase, let's call it $Address_X$) is written with a specific data pattern (let's say, P_{X}) for at least N_X times. The trigger has two inputs namely, En_{Add} and V_{p}. En_{Add} is the wordline enable signal of $Address_X$. Whenever $Address_X$ is written, En_{Add} will be asserted and MOSFETs M_1 and M_3 will be activated. The source and drain of M_2 is shorted and it has a thinner gate oxide compared to other MOSFETs. Therefore, M_2 works as capacitor and charges C_{Trojan} through Fowler Nordheim (FN) tunneling [11] from the V_{p} source if En_{Add} is asserted. M_4 is an OFF transistor which offsets gate leakage of M_5 and prevents unwanted-charging up of node S_2. M_7 keeps node X_3 as low as possible until node S_2 charges up sufficiently. V_{p} is the other input of the trigger circuit. Adversary can insert a weak NMOS ($W/L < 1$) (controlled by a RESET signal) between the source of M_4 and ground to control the duration of the attack. If RESET is asserted, trigger resets. Else, the attack will continue for 1.1ms (i.e. until node S_2 leaks through M_4 and RESET NMOS). Note that similar circuit to Fig. 4 can be implemented to generate RESET signal by writing to a different address (let's say $Address_Y$) with a specific data pattern (let's say P_{Y}) for N_{Y} times.

Activating the Trigger Circuit: En_{Add} signal assertion and a voltage source V_{p} both are required to charge the capacitor. Ground bounce generated during writing to $Address_X$ can be one source for V_{p}. In that case, the local ground of that address needs to be connected to the drain of MOSFET M_1. However, in a more complex attack, a simple logic circuit can be implemented which outputs logic 1 (1V) if a specific data pattern is sent to data bus. For example, let's consider that the data bus width is 8-bit. Assume that we take four specific data bits to design the trigger logic e.g., data[0], data[3], data[4] and data[6]. The logic circuit will output '1' if these bits are asserted except data[4] which should be de-



Specs: $|V_{tPMOS}| = 0.365V$, $VtNMOS = 0.423V$, $C_1 = 1\text{pF}$, , $T_{ox} = 1.85\text{nm}$
ΔV_{th}: $M_5 = +0.2V$, $M_6 = +0.1V$, $M_8 = +0.2V$, $M_9 = -0.2V$. ΔT_{ox} ; $M_2 = -0.2\text{nm}$
(+/- means V_{th}/T_{ox} magnitude increase/decrease)
(W/L): $M_1 = 4$, $M_2 = 9$, $M_3 = 3$, $M_4 = 1$, $M_5 = 0.5$, $M_6 = 12$, $M_7 = 0.06$, $M_8 = 8$, $M_9 = 2$

Fig. 4: Proposed Trojan trigger circuit (specification given at the bottom). V_{p} and En_{Add} are the inputs of the trigger.

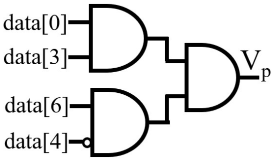
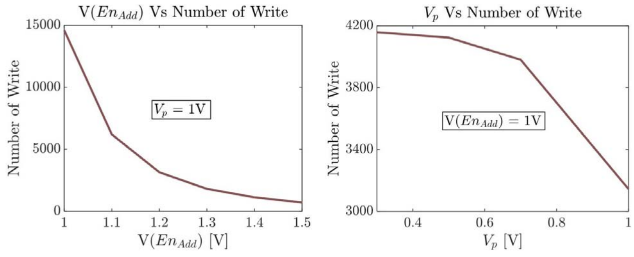
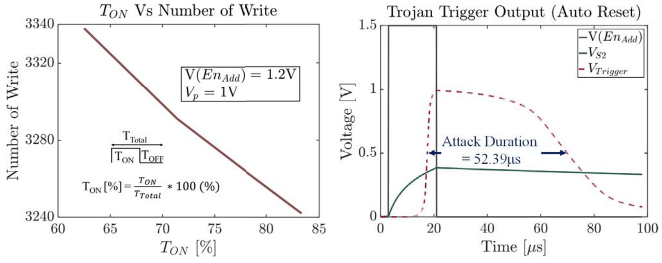


Fig. 5: An example of logic circuit to generate V_{p} from a specific data pattern.



(a) (b)

Fig. 6: (a) Required number of writes (N_X) to $Address_X$ Vs V(En_{Add}) to trigger the Trojan; (b) N_X decreases as V_{p} increases for a particular V(En_{Add}).



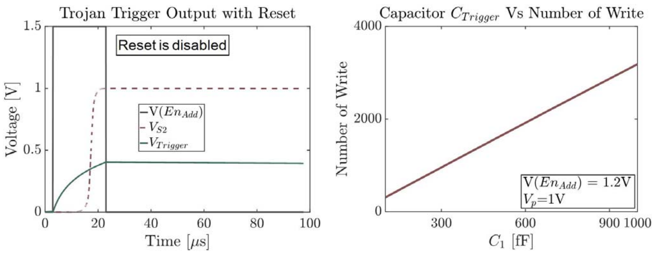
(a) (b)

Fig. 7: (a) Required number of writes (N_X) increases if adversary cannot write continuously; (b) attack resets automatically after 52.39μs once writing to $Address_X$ stops.

asserted (Fig. 5). In practice, data bits with low activation probabilities should be used to design the trigger logic to lower the overall probability of assertion unless intended. Note that, even if the trigger is asserted once or twice (probabilistically) during normal/test conditions, the Trojan will not be activated since N_X time-writing is necessary to charge C_{Trojan} whereas the Trojan proposed in [4] can be activated if the malicious address/cells is/are written just once with the desired data.

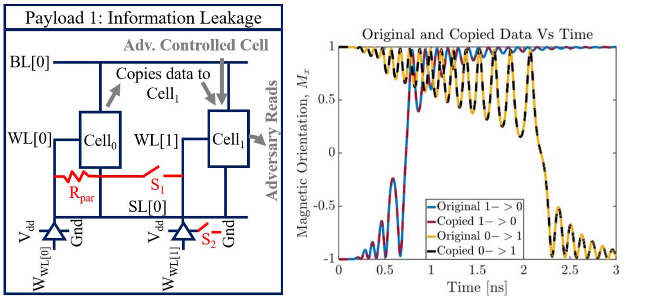
Simulation: Node S_2 charges up to 180mV (steady state) from all the leakage considering V_{p} = 1V all the time (worst-case charging up from leakage). This value is not enough to trigger the circuit. For the rest of the simulation, we considered that both V_{p} and En_{Add} are pulse sources with ON/OFF time of 10ns/10ns to consider that adversary can only write two consecutive times with one break in between. We consider the circuit to be triggered when V_{Trigger} reaches up to 0.5V. We started our analysis with C_{Trojan} = 1pF.

Fig. 6a shows the required number of writes (N_X) to $Address_X$ with respect to the $V(En_{Add})$ to trigger the Trojan for V_{p} = 1V. We find N_X = 14500 for $V(En_{\text{Add}})$ = 1V. Fig. 6b shows that N_X decreases for a particular $V(En_{Add})$ as V_{p} increases from 300mV (from ground bounce) to 1V (from logic circuit). This means that a higher voltage of V_{p} can activate the trigger quickly. Next, we considered that adversary writes for T_{ON} = 10μs and then stays idle for T_{OFF} = 2/4/6μs and repeats this cycle. We found that the C_{Trojan} does not leak in the OFF cycle significantly and the circuit can still be triggered but with a higher N_X (Fig. 7a). If adversary stops writing to $Address_X$ after the trigger activates, the node S_2 discharges and eventually, V_{Trigger} becomes zero (attack auto resets). We found that the attack lasts for 52.39μs if $Address_X$ is written for 16μs with $V(En_{\text{Add}})$ = 1.2V, V_{p} = 1V (Fig. 7b). However, by adding the RESET MOSFET, the attack lasts up to 1.1ms (Fig. 8a) and can be reset before 1.1ms by generating the RESET signal.



(a) (b)

Fig. 8: (a) Controlling trigger reset by inserting a RESET NMOS between source of M_4 and ground; (c) a scaled down $C_{Trigger}$ leads to less number of writes to activate the trigger.



(a) (b)

Fig. 9: (a) Payload 1: Information leakage. Switch S_1/S_2 is implemented using transmission/NMOS gate. WL[1] driver is gated with switch S_2 to prevent contention. R_{par} denotes the metal resistance to connect WL[0] and WL[1]; (b) data gets copied to $Cell_1$.

We optimized the value of C_{Trojan} since 1pF incurs significant overhead. The downside of a reduced C_{Trojan} is that the trigger will require less number of writes to get activated. For example, the required number of writes, N_X = 26 for C_{Trojan} = 10fF (Fig. 8b). Although the possibility of trigger activating in testing phase increases for a lower C_{Trojan}, it consumes significantly less area.

B. Payload 1: Information Leakage

Fig. 9a shows the payload circuit for information leakage. Two WLs sharing the same BL/SL can be coupled through a transmission gate (Sw_1). R_{par} denotes the resistance of metal connecting WL and the transmission gate. Therefore, WL[1] gets enabled when WL[0] is enabled since the transmission gate transfers full swing. WL[1] driver needs to be gated to prevent the contention (since WL[0] drives a '1' and WL[1] drives a '0'). Switch Sw_1/Sw_2 are asserted by the Trojan trigger. Fig. 9b shows write waveforms for 1→ 0 and 0→ 1 for original and copied cell. During write operation, the WL voltage is low (1.2V). Therefore, the current through R_{par} is very low and thereby, the voltage drop across R_{par} and Sw_1 is negligible. We notice that $Cell_0$/$Cell_1$ both are written at the same time (Fig. 9b). Once copied, the adversary can read $Cell_1$ and get victim's write data.

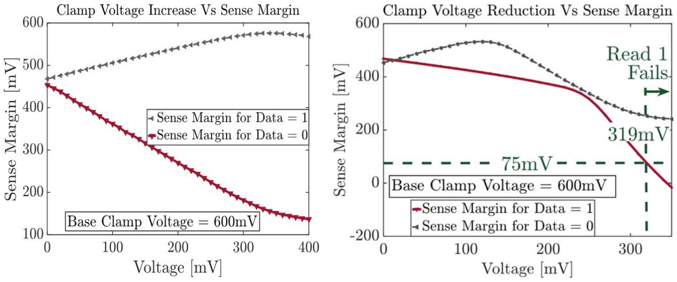
IV. Vulnerabilities of Read and Write Operation & Trojan Payloads

In this section, we investigate NVM read/write operations and explore their vulnerabilities to deploy memory Trojans.

A. Read Operation

(i) Manipulating V_{Clamp}: An adversary can manipulate the voltage output of V_{Clamp} source (details given in Section IV.C) and thereby, affect the read. In this work, the base V_{Clamp} is chosen to be 600mV for the best SM for both data '0' and '1'. Fig. 10a and Fig. 10b shows the impact of V_{Clamp} scaling up and down respectively. As V_{Clamp} scales up, the equivalent resistance of both reference legs reduces. Therefore, the parallel resistance of these two legs reduces significantly which in turn lowers the V_{Ref}. Similarly, V_{Data} reduces for data '0' but at a much higher rate compared to V_{Ref}. On the other hand, V_{Data} for data '1' does not change significantly. Therefore, the SM for data '1' increases while SM for data '0' decreases as V_{Clamp} is scaled up (Fig. 10a). However, SM for data '0' is sufficient (135mV which is higher than our threshold of 75mV, needed for correct sensing) when V_{Clamp} is scaled up by 400mV (Fig. 10a). Therefore, much higher V_{Clamp} is required to cause a read error. Fig. 10b shows that SM for both data '0' and '1' reduces as the V_{Clamp} is scaled down from the base value. However, the SM for data '1' goes below 75mV to trigger a read error with 319mV of V_{Clamp} reduction. Therefore, we conclude that it is easier from adversary's perspective to cause read error by reducing the V_{Clamp} voltage. In general, the adversary can manipulate the robustness of read operation by tampering with the V_{Clamp}.

(ii) Injecting Noise in the Ground Rail: Fig. 11 shows the SM for both data '0'/'1' with respect to ground bounce. SM for data '1' reduces whereas SM for data '0' stays relatively constant as the bounce experienced by a bitcell (during a read) increases. This is because the reference voltage also increases with bounce (Fig. 11). We conclude that if the bitcell incurs bounce > 400mV during read, the operation reads '0' incorrectly (since our threshold for correct sensing is 75mV). However, read '1' requires even higher bounce. Therefore, adversary can inject noise in the ground rail of victim's memory space and affect victim's read.



(a) (b)

Fig. 10: Impact of clamp voltage (a) scaled up and (b) scaled down in Fig. 3b from ideal value of 600mV.

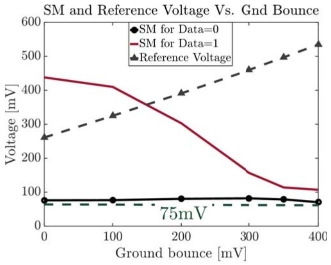


Fig. 11: STTRAM SM degradation as the bitcell being read, experiences higher ground bounce.

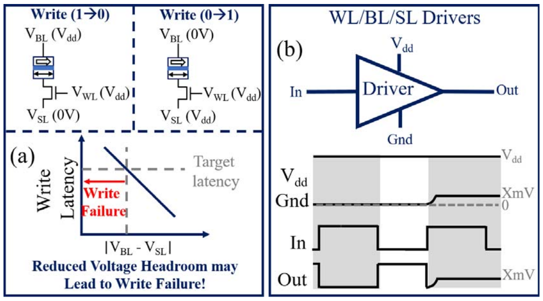
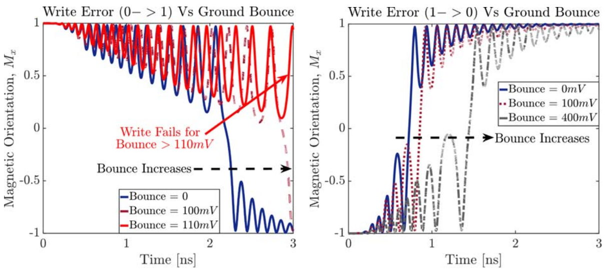


Fig. 12: (a) Reduced voltage headroom increases write latency (may cause failure); (b) $WL/BL/SL$ drivers can output non-zero voltage instead of 0V if the ground rail bounces.

B. Write Operation

Write operation of a typical three terminal NVM bitcell depends on three voltages namely, WL, BL and SL voltages (Fig. 12a). WL voltage turns ON the memory cell and determines the equivalent resistance of the access transistor. The BL/SL voltages determine the current magnitude and direction through the cell. The write latency depends on these voltages. Therefore, if any of these voltages are altered to lower the write voltage across the bitcell, the write latency will increase which can lead to failure (Fig. 12a). In this work, we have focused on increasing the voltage of BL or SL (Fig. 12b), whichever has 0V based on the write data polarity by injecting noise in the ground rail of the corresponding drivers. As their voltage increases, the voltage across the cell decreases leading to failure. Fig. 13a and 13b represents the impact of ground bounce on 0→ 1 and 1→ 0 writing respectively [12]. 0→ 1 writing fails if the bitcell experiences $> 110mV$ of ground bounce as the magnetic orientation (M_{x}) does not reach -1 (AP state). However, 1→ 0 write failure requires



(a) (b)

Fig. 13: STTRAM write latency for (a) 0→ 1 and (b) 1→ 0 increases as the cell incurs higher ground bounce [12].

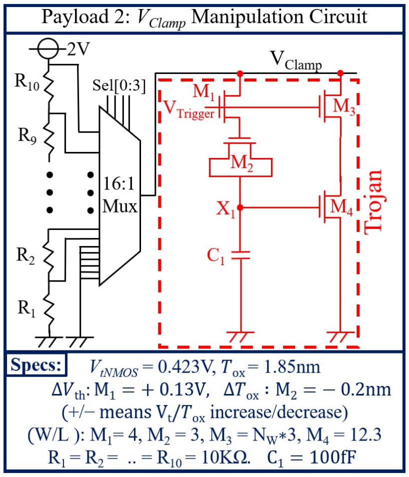
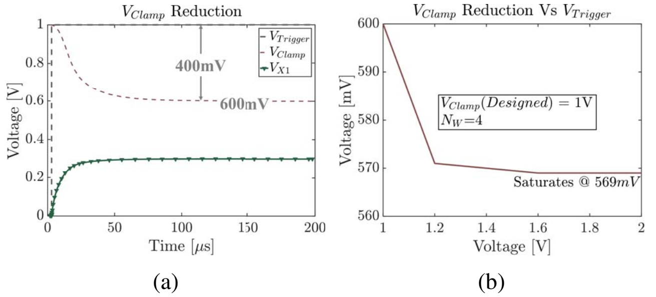

Payload 2: V_{Clamp} Manipulation Circuit
2V
R_{10}
R_9
R_2
R_1
Sel[0:3]
16:1
Mux
V_{Clamp}
M_1
$V_{Trigger}$
M_2
M_3
X_1
M_4
C_1
Trojan
Specs:
V_{tNMOS} = 0.423V, T_{ox} = 1.85nm
ΔV_{th}: M_1 = + 0.13V, ΔT_{ox} : M_2 = − 0.2nm
(+/− means V_t/T_{ox} increase/decrease)
(W/L): M_1= 4, M_2 = 3, M_3 = N_W*3, M_4 = 12.3
R_1 = R_2 = .. = R_{10} = 10KΩ. C_1 = 100fF

Fig. 14: Payload 2: Trojan in V_{Clamp} circuit.



(a)

(b)

Fig. 15: (a) V_{Clamp} reduction using Trojan proposed in Fig. 14. 400mV reduction is possible for V_{Clamp} = 1V, $V_{Trigger}$ = 1V and N_W = 4; (b) V_{Clamp} reduction is a function of $V_{Trigger}$.

very high ground bounce since write does not fail even with 400mV.

C. Payloads (Read/Write Failure)

Payload 2 - $V_{\textbf{Clamp}}$ Modification (Read Failure): Fig. 14 shows conventional V_{Clamp} generator circuit along with the proposed Trojan. A resistance ladder is used with ten resistors of equal values and the stack is connected to a 2V source. Therefore, the voltage across each of the resistances is 0.2V. These voltages can be used as V_{Clamp} voltages. A 16:1 multiplexer is used to select a V_{Clamp} after testing the manufactured chip for the maximum SM across all the memory bitcells. An adversary can modify the circuit to disrupt the SM. M_1 and M_3 turns ON once the trigger is activated (i.e. V_{Trigger} is asserted). Capacitor C_1 charges up and eventually turns ON M_4. So, the stack M_3 and M_4 pulls down the original V_{Clamp} voltage. Fig. 15a shows that roughly 75μs is required to achieve 400mV of voltage reduction if V_{Clamp} = 1V, V_{Trigger} = 1V and N_W = 4 where, N_W is the width multiplicative factor of M_3. Fig. 15b shows that V_{Clamp} reduction is a function of V_{Trigger}.

Payload 3 - Voltage Headroom Reduction (Read/Write Failure): Fig. 16 shows techniques to reduce voltage headroom by injecting noise or by shorting BL/SL with ground by a NMOS switch. The source of noise, $V_{disturb}$ can be any voltage source or it can be a parallel write operation which creates a ground bounce. Voltage headroom reduction worsens SM/write latency during read/write operation respectively. Therefore, this can lead to read/write failure.

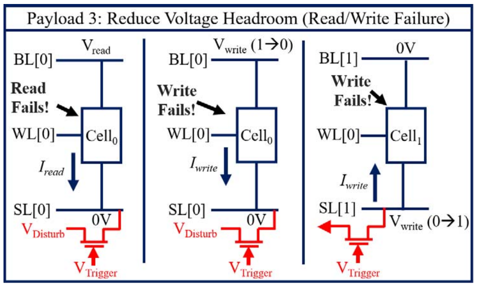


Fig. 16: Payload 3: Voltage headroom reduction techniques.

V. Conclusion

In this paper, we propose a Trojan trigger for emerging NVMs capable of evading the post-manufacturing testing. The Trojan is triggered by writing a preselected address with a preselected data pattern for a specific number of times. We propose coupling two addresses which can copy victim's write data to a adversary-controlled address. We also propose Trojan payloads which exploit NVM-specific read/write operation (e.g., clamp voltage and supply noise) which can lead to read/write failure.

Acknowledgment

This work is supported by SRC (2727.001), NSF (CNS-1722557, CCF-1718474, DGE-1723687 and DGE-1821766) and DARPA Young Faculty Award (D15AP00089).

References

[1] M. H. Tehranipoor et al, *"Introduction to Hardware Security and Trust"*. Springer, 2012.

[2] R. S. Chakraborty et al, "Hardware Trojan: Threats and Emerging Solutions," in *2009 IEEE International High Level Design Validation and Test Workshop*, pp. 166–171, Nov 2009.

[3] "DARPA TRUST in ICs Effort. [Online]. Available: http://www.dtic.mil/dtic/tr/fulltext/u2/a503809.pdf. Accessed: Oct 28, 2018," 2018.

[4] T. Hoque et al, "Hardware Trojan Attacks in Embedded Memory," in *2018 IEEE 36th VLSI Test Symposium (VTS)*, pp. 1–6, April 2018.

[5] K. Yang et al, "A2: Analog Malicious Hardware," in *2016 IEEE Symposium on Security and Privacy (SP)*, pp. 18–37, May 2016.

[6] M. N. I. Khan et al, "Information Leakage Attacks on Emerging Non-Volatile Memory and Countermeasures," in *Proceedings of the International Symposium on Low Power Electronics and Design*, ISLPED '18, (New York, NY, USA), pp. 25:1–25:6, ACM, 2018.

[7] M. N. I. Khan et al, "Fault Injection Attacks on Emerging Non-Volatile Memory and Countermeasures," in *Proceedings of the 7th International Workshop on Hardware and Architectural Support for Security and Privacy*, HASP '18, (New York, NY, USA), pp. 10:1–10:8, ACM, 2018.

[8] X. Wang et al, "Detecting Malicious Inclusions in Secure Hardware: Challenges and Solutions," in *2008 IEEE International Workshop on Hardware-Oriented Security and Trust*, pp. 15–19, June 2008.

[9] "Addressing Test Time Challenges. [Online]. Available: https://semiengineering.com/addressing-test-time-challenges/. Accessed: Oct 28, 2018," 2017.

[10] J.-H. Song et al, "Sensing Margin Trend with Technology Scaling in MRAM," *International Journal of Circuit Theory and Applications*, vol. 39, no. 3, pp. 313–325.

[11] N. M. Ravindra et al, "Fowler-Nordheim Tunneling in Thin SiO_2 Films," *Smart Materials and Structures*, vol. 1, no. 3, p. 197, 1992.

[12] M. N. I. Khan et al, "Analysis of Row Hammer Attack on STTRAM," in *2018 IEEE International Conference on Computer Design (ICCD)*, Oct 2018.