Securing Hardware via Dynamic Obfuscation Utilizing Reconfigurable Interconnect and Logic Blocks

Gaurav Kolhe, Soheil Salehi,
Tyler David Sheaves, Houman Homayoun
Dept. of Electrical & Computer Engineering
University of California, Davis, CA, USA
{gskolhe, ssalehi, tsheaves, hhomayoun}@ucdavis.edu

Setareh Rafatirad
Dept. of Computer Science
University of California, Davis
CA, USA
srafatirad@ucdavis.edu

Sai Manoj P D, Avesta Sasan
Dept. of Electrical & Computer Engineering
George Mason University, Fairfax
VA, USA
{spudukot, asasan}@gmu.edu

2021 58th ACM/IEEE Design Automation Conference (DAC) | 978-1-6654-3274-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/DAC18074.2021.9586242

***Abstract*—Maximizing profits while minimizing risk in a technologically advanced silicon industry has motivated the globalization of the fabrication process and electronic hardware supply chain. However, with the increasing magnitude of successful hardware attacks, the security of many hardware IPs has been compromised. Many existing security works have focused on resolving a single vulnerability while neglecting other threats. This motivated to propose a novel approach for securing hardware IPs during the fabrication process and supply chain via logic obfuscation by utilizing emerging spin-based devices. Our proposed dynamic obfuscation approach uses reconfigurable logic and interconnects blocks (RIL-Blocks), consisting of Magnetic Random Access Memory (MRAM)-based Look Up Tables and switch boxes flexibility and resiliency against state-of-the-art SAT-based attacks and power side-channel attacks while incurring a small overhead. The proposed Scan Enabled Obfuscation circuitry obfuscates the oracle circuit's responses and further fortifies the logic and routing obfuscation provided by the RIL-Blocks, resembling a defense-in-depth approach. The empirical evaluation of security provided by the proposed RIL-Blocks on the ISCAS benchmark and common evaluation platform (CEP) circuit shows that resiliency comes with reduced overhead while providing resiliency to various hardware security threats.**

***Index Terms*—Reverse Engineering, Logic Locking, SAT Attack, Emerging Devices, Dynamic Obfuscation, Power side channel attack**

I. Introduction

The ever-increasing complexity of IP designs and shrinking transistor size have proliferated the globalization of the fabrication process and hardware supply chain. Today, the design, manufacturing, and assembly of the modern ICs consisting of millions and billions of transistors with intricate fabrication processes are backed by a complex network of global suppliers. One of the U.S.-based semiconductor companies has $7,300$ suppliers spread over 46 states in the U.S., and more than $8,500$ suppliers are geographically located outside of the U.S. The globalization of the fabrication process and hardware supply chain is structured to benefit mainly from the diverse and varied skills of human resources and geographical advantages of suppliers. While the global ecosystem has continued to benefit both participant and their global economy, the security of the underlying hardware is compromised due to various emerging hardware security threats such as overproduction, trojan insertion, reverse engineering, IP theft, and counterfeiting.

With the increasing success of hardware security attacks, the tenet "trust starts in silicon" which assumes hardware as the root of security is questionable. Fortifying the security of hardware is a multifaceted problem, and the hardware security community has developed various techniques to re-imagine the security of the hardware. Hardware design-for-trust techniques such as split manufacturing, IC camouflaging, and logic locking have shown promise in alleviating the major issue of hardware security threats such as IP piracy, overproduction, and reverse-engineering, to name a few. However, amongst multiple aforementioned techniques, logic locking has the ability to thwart the majority of hardware security attacks during various phases of globalized supply chain [1]. Logic locking requires the correct keys to unlock the true functionality of the design, and as a part of the post-manufacturing process, the activation of IC (i.e., providing correct keys) is accomplished in a trusted regime. Until this stage, the IC is in the obfuscated state, and the functionality of IP is unknown.

This work was supported in part by the National Science Foundation (NSF) through Computing Research Association for CIFellows Project 2030859.

With ongoing widespread research in hardware security, the boolean attack, also known as SAT-attack, has the ability to restore the functionality of the obfuscated IP [2]. The SAT attack requires an **oracle circuit** (activated IC) and the netlist retrieved using invasive reverse engineering efforts. The SAT attack tries to find a distinguishing input pattern (DIP) that can differentiate between two keys. Applying this DIP to the oracle helps in eliminating the incorrect keys. SAT-attack iteratively repeats this procedure until it can no longer find the DIP.

To thwart the SAT-attack, many works seek to increase the number of iterations required to retrieve the correct key for unlocking the circuit. Stripped Functionality Logic Locking (SFLL) is the state-of-the-art method requiring exponential SAT iterations to find the correct key. It thwarts not only the SAT-attack but also bypass and removal attacks [3]. However, work in [4], and [5] exploits the vulnerability in the SFLL method implementation and shows that the obfuscation key can be found within several minutes. Moreover, SFLL, SAR-Lock, and Anti-SAT obfuscation primitives fall into categories of one-point function, which evaluates to correct output upon applying a specific input pattern. Though it requires SAT-attack to apply many inputs to find the correct key, the output corruptibility offered by such techniques is very low, which means that the circuit works almost identical to the oracle circuit even when the wrong key is used [6].

In the quest to resist the SAT-attack, the community also formulated obfuscation techniques that resist the obfuscated netlist's transformation to SAT problems. Delay-based locking or cyclic obfuscation are among a few of the examples that resist the SAT-attacks. However, shortly after introducing these obfuscation techniques, the attacks such as cycSAT and satisfiability module theories (SMT) attack were able to model the cyclic or behavioral locking into an SAT or SAT+theory solvable logic problems [7].

On the other hand, works in the reconfigurable domain thwart SAT-attacks by increasing the key search space. The M-input Look Up Table (LUT) based obfuscation [8] offers a key search space of 2^{2^m} to the attacker. However, the larger overhead imposed by the LUT refrains their practical implementation for security purposes. Another work from the reconfigurable domain uses a magneto-electric spin-orbit (MESO) device, which offers both reconfigurability and dynamic morphing to thwart most of the attacks while imposing lower overheads. While this is a step in the right direction, the dynamic morphing/camouflaging based obfuscation can only be leveraged in applications that can tolerate a certain degree of error [9]. Finally, the work in [10] tries to resist the SAT-attack by building an SAT-hard instance using the key-configurable logarithmic-based network for obfuscation of routes. While this seems to thwart SAT-attack, a different formulation of the SAT problem can help the attacker in retrieving the keys quickly [11].

With this cat and mouse race of the obfuscation techniques to beat SAT-based attacks, new emerging attacks have been able to find vulnerabilities time after time. Additionally, while other works [8], [11], [12] mainly focus on resisting one type of attack, in this work, we present a novel approach to resist and thwart various types of attacks. The contribution of the work is as follows:

- The proposed RIL-Blocks use MRAM-based LUTs and routing-based obfuscation to present an SAT-hard instance to the SAT-attack while offers resiliency to removal attacks.
- LUT and routing blocks can dynamically morph during runtime,

Authorized licensed use limited to: Univ of Calif Davis. Downloaded on October 05,2022 at 21:44:15 UTC from IEEE Xplore. Restrictions apply.

thus offering another opportunity to thwart the SAT-based attack.
- As the SAT-attack requires oracle responses to find the correct key successfully, the Scan Enabled obfuscation circuitry obfuscates the oracle's responses, thus eliminating the SAT-attack and offers defense-in-depth.
- The RIL-Blocks are built utilizing commercially available Spin Transfer Torque (STT) Magnetic Tunnel Junction (MTJ), which resists the Power Side-Channel Attack by attaining symmetrical power footprint with a near-zero power variation in the output.
- Finally, the empirical evaluation of the RIL-Blocks shows that security comes with minimal overhead.

II. Background and Motivation

A. Polymorphic Logic using Emerging Devices

Among the most commonly used reconfigurable fabrics for logic obfuscation, LUTs have been the primary focus due to their flexibility, allowing the realization of logic elements at medium and fine granularities while incurring low non-recurring engineering costs. Additionally, LUTs have been researched as a promising platform that can be utilized effectively to increase reliability in process-voltage-temperature variation [8], [13]. The main challenge of static random access memory (SRAM)-based LUTs is their increased area and power consumption to achieve flexible design [14]. However, SRAM-based LUTs incur limitations such as high static power, volatility, and low logic density. Moreover, traditional SRAM-based LUTs are vulnerable to Power Side-Channel Attacks (P-SCA) from the security perspective.

Innovations using emerging devices for LUT design have been sought to bridge the gaps needed to overcome the limitations of SRAM-based LUTs [13], [15]. High-endurance non-volatile spin-based LUTs have been studied in the literature as promising alternatives to SRAM-based LUTs, Flash-based LUTs, and other state-of-the-art emerging LUTs such as resistive random access memory (RRAM)-based LUTs and phase-change memory (PCM)-based LUTs [13], [15]–[17]. Additionally, spin-based devices offer non-volatility, near-zero static power, high endurance, high integration density, and compatibility with the CMOS fabrication process [13]. Majority of the spin-based LUTs proposed in the literature fail to maintain a wide sense margin and suffer from high error rate specially in the presence Process Variation (PV) [13]. In this paper, to address the challenges mentioned earlier, we develop a novel secure and P-SCA resilient non-volatile Magnetic RAM (MRAM)-based LUT with a wide read margin using commercially available Spin Transfer Torque (STT) Magnetic Tunnel Junction (MTJ) and provide a comparison with traditional SRAM-based LUT. Additionally, we provide an analysis of the reliability in the presence of Process Variation (PV) and the security and attack resiliency of our proposed LUT. Researchers have demonstrated polymorphic logic implementation using Magneto-Electric Spin-Orbit (MESO) devices can achieve significant power dissipation reduction [9]. However, in this paper, we use STT-MTJ devices due to the fact that they are commercially-available as opposed to MESO devices.

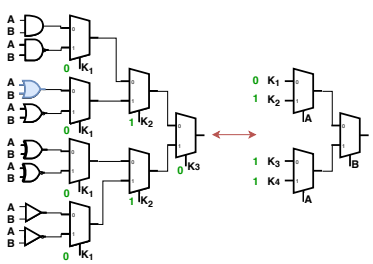


Fig. 1. Example of replacing circuit on the left with LUT representation in the bench file for SAT-simulation. Key Value of **010** represent OR circuit, while Key value of **0111** represent OR circuit with LUT.

B. Motivation

The work in [9] shows that without dynamic camouflaging, the MESO device can still render a SAT-hard instance, which results in timeout states. Moreover, their SAT-formulation of MESO devices in bench file resembles a MUX with additional 8 gates and 7 MUXes. However, the same MESO device can be replaced with LUT of size 2, using just 3 MUXes, as shown in Figure 1. LUT of size 2 can emulate all 8 functions that a MESO device can offer. Using this transformation, the SAT-attack runtime of MESO-device-based obfuscation is decreased significantly. While it is true that the dynamic camouflaging of MESO devices can still thwart SAT-attack, nonetheless, their **application is limited to the obfuscation for approximate applications like image processing, which can tolerate a certain degree of errors**.

On the other hand, the LUT-based obfuscation provides a large key search space for the SAT-solver to find the key. However, LUTs impose a substantial overhead. Recent work in [12] endeavor to reduce the incurred overhead without sacrificing the SAT-resiliency. However, the incurred overhead is still significant. On the contrary, work in [10] tries to increase SAT-hardness by using the key-configurable logarithmic-based network, which tries to push clause to the variable ratio between 3 to 6 for creating SAT-hard problem. Despite their efforts to thwart SAT-attack, work in [11] shows that the key can be retrieved within polynomial time. Furthermore, they propose an incremental version of the routing-based obfuscation, but creating an SAT-hard problem is still subjective in terms of security. An SAT-hard problem of today can be solved faster with emerging SAT-solvers and hardware. While most strategies focus on curbing the SAT-attack, one cannot rule out the power side-channel attack (P-SCA) possibility.

To fortify the security of the hardware, inspired by the ideas discussed above, we propose the RIL-Blocks, which thwarts SAT-attack [2], removal attack, probabilistic attacks, and mitigating the non-invasive P-SCAs by utilizing MRAM-based LUTs. We discuss the construction of the RIL-Blocks in the following section.

III. Proposed RIL-Blocks

A. Building SAT-hard instances

SAT-solvers are widely used because of their ability to solve the problems effectively in practice using the Conflict-Driven Clause Learning (CDCL) algorithm. The CDCL uses the Davis–Putnam–Logemann–Loveland (DPLL) algorithm, with the additional ability to learn new clauses and backtrack non-chronologically [18]. The CDCL algorithm was the fastest SAT-solver at the 2019 CaDiCaL SAT-solving competition [18]. As previously discussed, many obfuscation techniques use the one-point function to increase the number of DIP for finding the correct key and in the SAT-solver domain, this phenomenon corresponds to increase the number of DPLL iterations. While this is a promising solution, one must trade the less output corruptibility offered by the obfuscation with the higher SAT-resiliency. To avoid this, in the LUT-based obfuscation, LUTs can be placed in different output logic cones, for increasing the output corruptibility while leveraging large sizes of LUT for building higher SAT-resiliency. The improved version of the LUT proposed in [12] uses the blend of small and large LUT, which results in SAT-hard instances. The symmetric problem formed due to the MUX-tree structure of the LUT forces a relationship between the number of clauses and the number of variables to maximize the penalty associated with incorrect variable assignment symmetrically across the search tree. Furthermore, the same symmetric problem can be created using the logarithmic routing network, as discussed in [10].

Inspired by this idea and the FPGA architecture, we propose a *blend of LUT and a logarithmic routing-based obfuscation that increases the search key space for the SAT-solver and also helps in creating an SAT-hard problem.* Furthermore, to reduce the overhead incurred by the LUT, we impose the size constrain on the LUT, and proceed with LUT of size 2, which can be even replaced by MESO or other kinds of emerging devices. While this method resists the SAT-attack, increasing

Authorized licensed use limited to: Univ of Calif Davis. Downloaded on October 05,2022 at 21:44:15 UTC from IEEE Xplore. Restrictions apply.

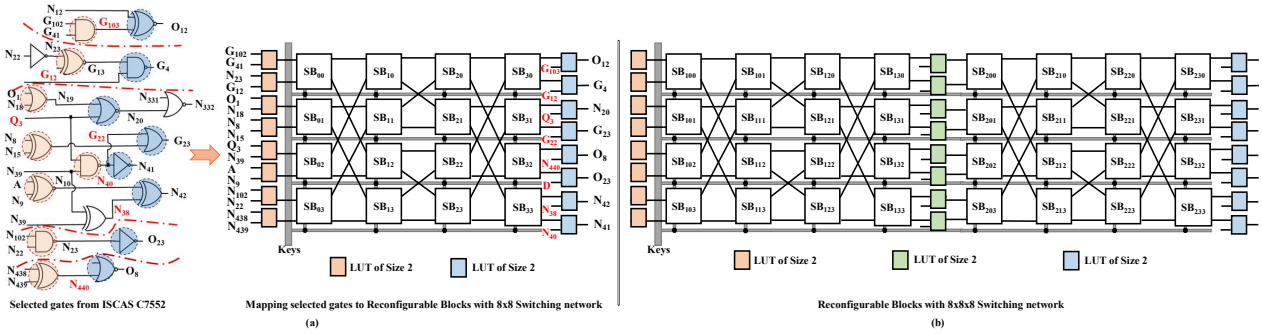


Fig. 2. (a) A simple example showing the mapping of gates in an ISCAS C7552 benchmark to 8×8 RIL-Block (b) Implementation details of 8×8×8 RIL-Block

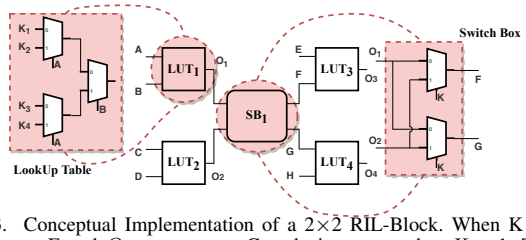


Fig. 3. Conceptual Implementation of a 2×2 RIL-Block. When K = 0, O_1 connect to F and O_2 connects to G and vice versa when K = 1. The LUT represented here can be replaced with a emerging device that can provide 16-functions.

the size of LUT can further fortify the security. Figure 3 shows the RIL-blocks using LUT of size 2 and a single Logarithmic banyan network. The banyan network used here is almost non-blocking in nature and has (N/2)$\log_2$N switching blocks.

While constructing the RIL-block, we tested various configurations of the RIL Block. Figure 3 shows 2×2 RIL-block that uses a single switch box. This switch box's inputs are connected to the fanout of the LUTs (O_1, O_2), which replaces a 2-input gate in the circuit, while the switch box's output acts as the input to one of the 2-input LUTs (F and G). As the 2×2 RIL-block can be instantly de-obfuscated using the SAT-attack, we increased the size of the switching network, which resembles the banyan network. We tested 8×8 RIL-block with 16 switching elements and 8×8×8 RIL-block with 32 switching elements for obfuscation purposes. The implementation of 8×8 and 8×8×8 RIL-block are shown in Figure 2 while the representation of each switching block for the SAT-solver simulation is shown in Figure 3. Additionally, we show an elementary example of mapping the gates of the *ISCAS C7552* benchmark to the 8×8 RIL-block for a better understanding of the implementation.

For the SAT-simulation, each LUT and routing block are converted into the MUXes and contribute to the increasing recursive DPLL tree search. The addition of the MUXes helps in increasing the clause to variable ratio, and the resulting symmetric switching networks in the circuit with back-to-back interconnection with LUTs increases the SAT-hardness even for state-of-the-art CaDiCaL SAT-solver. This increase in the SAT-hardness is due to the penalty associated with incorrect variable assignment in the DPLL algorithm. Upon a wrong, incorrect variable assignment, DPLL has to backtrack from a deep end of the tree and start again from the branch that had the wrong variable assignment. **The powerful aspect of this form of building SAT-hard solutions using RIL-block is that (1) the problems posed at each iteration of SAT attack is an SAT-hard problem, (2) the output corruption of this methods is significantly higher than obfuscating solution relying on one-point function, (3) it is not susceptible to bypass, removal, or approximate attack and (4) it also attempts to eliminate SAT-attack and mitigates P-SCA.** To test the resiliency and SAT-hardness of the proposed RIL-block, we obfuscated 'C7552' benchmark from ISCAS-85 with different RIL-blocks size and swept the number of replaced gates. It can be inferred from the results in Table I, increasing the number of RIL-blocks increases the SAT-hardness, but increasing the size of RIL-blocks shows stronger SAT-resiliency. Instead of obfuscating 75 gates with 2×2 RIL-blocks, utilizing 3-8×8×8 blocks results in SAT-timeout and the overhead incurred by leveraging the 8×8×8 blocks is ∼3× lower when compared to 75-2×2 RIL-blocks.

This is indicative of the fact that inserting the increasing size of SAT-hard RIL-block instances in the circuit, the SAT encounters with a very deep and especially large DPLL recursion tree in each iteration, which considerably and exponentially increases the execution time of DPLL recursive calls. In comparison with the work in [10], the proposed switch box in our work contains just 2 MUXes instead of 4. An additional inverter in the switch box of FullLock adds to extra overhead and increases the number of correct keys in the circuit. For example, one wrong inversion in the first switchbox element can be converted back to the right value using another inversion element in the next switch box. In the following sections, we expand upon the construction of the LUT used in our proposed RIL-Blocks.

TABLE I
TIME TAKEN BY SAT-SOLVER IN SECONDS TO FIND KEY WITH DIFFERENT SIZES AND QUANTITY OF RIL-BLOCKS FOR C7552 BENCHMARK.

	Size of RIL Blocks				Size of RIL Blocks		
RIL Blocks	2x2	8x8	8x8x8	RIL Blocks	2x2	8x8	8x8x8
1	0.31	0.63	23.53	10	1.16	∞	∞
2	0.35	6.33	198.556	25	34.5	∞	∞
3	0.405	20.422	∞	50	102.319	∞	∞
4	0.55	180.938	∞	75	∞	∞	∞
5	0.67	316.231	∞	100	∞	∞	∞

B. MRAM-based LUT for RIL-Blocks

The primary goal of using LUTs is implementing combinational logic. Generally, M-input LUTs have 2^M memory cells to implement M-input Boolean functions. A select tree MUX circuit is utilized to select the memory cell that holds the correct value of the function the LUT is implementing. The select tree MUX is constructed with Pass Transistors and Transmission Gates (TGs). Figure 4 depicts a 2-input example of our proposed MRAM-based LUT design.

As shown in Figure 4, **WE** and $\overline{\textbf{WE}}$ signals control the write operation via connecting of each memory cell to the Bit Line, **BL**, and Source Line, **SL**. During the write operation, by using the inputs **A** and **B** we can access each memory cell separately, and by setting **BL** and **SL**, we can change the content of the memory. Additionally, in each writes operation, we change the content of MTJs in each memory cell in a complementary fashion so that $\textbf{MTJ}_i$ and $\overline{\textbf{MTJ}_i}$ always hold opposite values. In particular, assuming the data stored in the $\textbf{MTJ}_1$ is in the P or low-resistance state, then $\overline{\textbf{MTJ}_1}$ is going to be in the AP or high-resistance state and vice versa. Thus, instead of using a sense amplifier with a reference cell, we can use each cell's complementary value to read the data stored in the main memory cell reliably. This is mainly due to the wide read margin during the read operation enabled by this sensing method.

Authorized licensed use limited to: Univ of Calif Davis. Downloaded on October 05,2022 at 21:44:15 UTC from IEEE Xplore. Restrictions apply.

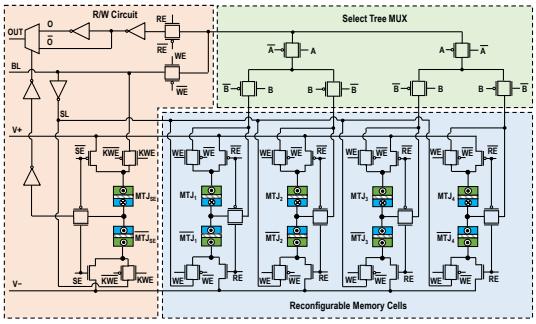


Fig. 4. The circuit-level diagram of the proposed 2-input MRAM-based LUT using STT-MTJ devices.

Once the write operation is terminated, by asserting the **RE** and $\overline{\textbf{RE}}$ signals, we can read the data stored in the MTJs. Read enable signals **RE** and $\overline{\textbf{RE}}$ enable the read path from **V+** to **V-**. This will result in a voltage divider circuit, which will be used the observe the resistance difference between the $\textbf{MTJ}_i$ and $\overline{\textbf{MTJ}}_i$. The terminal between the two MTJs is connected to the select tree MUX to direct the proper output according to the input signals **A** and **B**. This value of the LUT function is observed at the output nodes **O** and $\overline{\textbf{O}}$.

The proposed LUT is designed to increase the security of the design via dynamic obfuscation and P-SCA mitigation. This is due to the fact the output of a LUT is a function of all of the inputs, and the proposed MRAM-based LUT maintains a symmetrical power consumption footprint and delays to provide an output of '0' or '1,' and as a result, it maintains a near-zero power variation in the output. Furthermore, the MTJ devices' contents can be reconfigured in every LUT to change the functionality that each LUT is implementing. To define the functionality of each 2-input LUT, a set of keys are shifted in via the Bit line **BL** signal and by controlling **A** and **B** inputs, we can control which memory cell we wish to change or update the contents of. For example, for the AND function, **A** and **B** inputs are used to select each memory cell in the order of AB such as 11, 10, 01, and 00 while the keys to configure the functionality of the LUT are shifted through the **BL** as 1, 0, 0, and 0, respectively. Next, we will discuss the Scan Enable obfuscation of the LUT output.

TABLE II
CONFIGURATION KEY BITS FOR IMPLEMENTING 16 DIFFERENT BOOLEAN FUNCTIONS UTILIZING THE PROPOSED MRAM-BASED 2-INPUT LUT

Function	K1	K2	K3	K4	Function	K1	K2	K3	K4
0	0	0	0	0	1	1	1	1	1
A NOR **B**	0	0	0	1	**A** OR **B**	1	1	1	0
$\overline{\textbf{A}}$ AND **B**	0	0	1	0	$\overline{\textbf{A}}$ NAND **B**	1	1	0	1
$\overline{\textbf{A}}$	0	0	1	1	**A**	1	1	0	0
A AND $\overline{\textbf{B}}$	0	1	0	0	**A** NAND $\overline{\textbf{B}}$	1	0	1	1
$\overline{\textbf{B}}$	0	1	0	1	**B**	1	0	1	0
A XOR **B**	0	1	1	0	**A** XNOR **B**	1	0	0	1
A NAND **B**	0	1	1	1	**A** AND **B**	1	0	0	0

C. Scan Enable Obfuscation Mechanism

The RIL-block proposed uses a mix of LUTs and interconnect obfuscation for SAT-hardness, while MRAM-based LUTs help with dynamic morphing and P-SCA mitigation. To further secure the primitive against SAT-attack and zero-day threats, we propose using an additional MTJ memory cell within the LUT that holds a key value that can be dynamically set for each LUT. During the read operation, if the Scan Enable signal **SE** is asserted, the data stored in $\textbf{MTJ}_{SE}$ and $\overline{\textbf{MTJ}}_{SE}$ will determine whether **O** makes it to the **OUT** or $\overline{\textbf{O}}$. The **KWE** and $\overline{\textbf{KWE}}$ signals are utilized to control the write operation of the memory cell holding this key. The contents of this additional memory cell is randomly set for each MRAM-based LUT used in the design. When the attacker wants to perform the SAT-attack, the **SE** signal in the Oracle IC will be asserted to apply the DIP and receive the responses. At this instance, the output of the selected LUTs that have a key of '1' stored in $\textbf{MTJ}_{SE}$ will be inverted, and the obfuscated response is obtained, thus providing an additional layer of security. Since the IP designer is aware of the values stored in the $\textbf{MTJ}_{SE}$ memory cells, it will not cause any errors during the test phase. This is because the IP designer is aware of possible inversions in the **OUT** signal after the assertion of **SE**. However, we assume the attacker will not have access to the contents of the $\textbf{MTJ}_{SE}$ and thus, will not be able to find out the correct functionality the LUTs.

D. Insertion of the RIL-Blocks

The proposed RIL-blocks can be inserted in the circuit without employing a specific insertion policy. There are no restrictions on gates or wires that one must select for the obfuscation, thus reducing the extra efforts on the IP designer. However, the hardware security community commonly focuses on inserting the obfuscation in the logic cone with a large number of gates, but this reduces the number of corrupted outputs. The RIL-blocks can be inserted randomly in the circuit with the random gates selected for replacement with the LUT. Despite random selection, the RIL-blocks provides superior SAT-resiliency with high output corruptibility. Since the insertion is random, we have multiple gates and wires to pick from, which eases the insertion process.

IV. EXPERIMENTAL SETUP

For empirically demonstrating the efficacy of the RIL-blocks, we test it against state-of-the-art SAT-attack with new CaDiCaL solver [18]. The reason we updated the SAT-solver from [2] with CaDiCaL solver is because it is much faster than the older lingeling solver used in SAT-solver from [2]. Furthermore, SAT-solver with CaDiCaL solver can solve the obfuscated circuits with $\sim$1.8$\times$ faster than the [2] on average. The benchmark suite for testing the RIL-Blocks consists of ISCAS-89 benchmark along with CEP benchmarks[1]. The benchmark suite consists of an ISCAS benchmark widely used in the community with few IPs that are representative example of the size and complexity of real-world designs. All experiments were conducted on a 64-bit machine with a quad-core processor (Intel(R) Xeon(R) CPU E3-1271 v3 @ 3.60GHz) with 64GB memory. The threat model assumes that the key is stored in the tamper-proof memory, and the attacker has access to the fully reverse engineered IC and the activated Oracle IC. While most of the work in Hardware Security uses 1-hour [19] or at most 2 days of SAT-runtime [3], [10], [11] to support their case for SAT-hardness, in our case, the timeout is set to 5 days. Furthermore, we use the HSPICE circuit simulator to validate the functionality of proposed MRAM-based LUT using 45nm CMOS technology and the STT-MRAM model developed in [20].

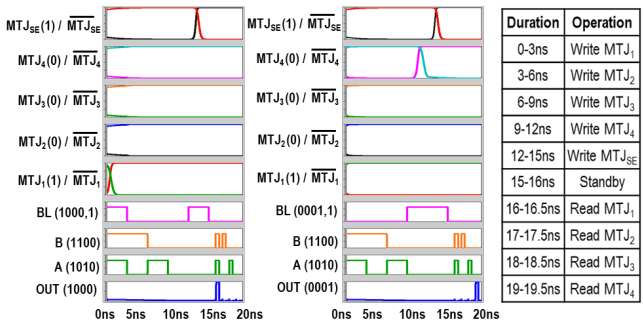
A. Circuit-Level Simulation Setup and Results

MTJs are constructed of two ferromagnetic layers, called free layer and fixed layer, and a thin oxide layer [13]. In the STT switching approach used in Spin Transfer Torque Magnetic Random Access Memory (STT-MRAM), applying a bidirectional charge current through the terminals of the MTJ using a MOS-based circuit, will result in the generation of a spin current that changes the magnetic polarity of the free layer to represent: 1) high resistance or Anti-Parallel (**AP**) state, and 2) low resistance or Parallel (**P**) state. The states of the MTJ are determined according to the angle, θ, between the magnetization orientation of the ferromagnetic layers. Herein, we have adopted the MTJ device parameters from [20].

Figure 5 shows the transient response of the proposed MRAM-based LUT. Figure 5(a) depicts an implementation of a 2-input AND gate while Figure 5(b) illustrates an implementation of NOR gate within the same LUT by reconfiguring the key bits. Additionally, as it can be observed in Figure 5, the content of the $\textbf{MTJ}_{SE}$ is updated to provide the correct output. As shown, the HSPICE simulations verify the correct functionality of our proposed MRAM-based LUT.

[1]The CEP is a system on a chip design that is representative of typical microelectronics that are used by the body of the Department of Defense (DoD) and includes instrumentation and government-specific benchmarks.

Authorized licensed use limited to: Univ of Calif Davis. Downloaded on October 05,2022 at 21:44:15 UTC from IEEE Xplore. Restrictions apply.



Duration	Operation
0-3ns	Write MTJ_1
3-6ns	Write MTJ_2
6-9ns	Write MTJ_3
9-12ns	Write MTJ_4
12-15ns	Write MTJ_{SE}
15-16ns	Standby
16-16.5ns	Read MTJ_1
17-17.5ns	Read MTJ_2
18-18.5ns	Read MTJ_3
19-19.5ns	Read MTJ_4

(c)

Fig. 5. Simulation waveforms for implementation of 2-input (a) AND gate, (b) NOR gate, and (c) operating modes.

B. Resiliency Against Attack

The SAT-attack conducted on the obfuscated circuit involved one-layer linear encoding, which replaces the original sub-CNF of the routing block with a CNF with one layer of MUX controlled by the one-hot key and further reduces the CNF using the BVA algorithm discussed in [11]. This pre-processing step increases the efficacy of the SAT-solver in solving routing obfuscation-related problems. Table III shows the time taken by the SAT-solver in seconds when RIL-blocks of $8\times8\times8$ was leveraged for the obfuscation. It can be seen that RIL-block can easily thwart the SAT-attack in the static operational mode. With 3 RIL-blocks used in the circuit, the SAT-solver fails to find the keys for all of the benchmarks. Table III also shows that AppSAT fails and terminates erroneously for all circuits with the default setting when Scan enabled circuitry is activated.

As shown in [9], [19] leveraging the dynamic morphing offered by the emerging devices thwarts the SAT-attack ultimately. However, controlling the functionality of the devices using TRNG, as demonstrated in [9], reduces the scope of obfuscation. Such obfuscation can only be used in the IPs that can tolerate a certain degree of error. We can certainly create circuitry that can control the dynamic morphing of the emerging devices in real-time, however, in this case, the attacker's focus will shift in reverse engineering of that circuitry, or even that circuitry can fall prey to removal attacks. Therefore, in this attack, we use the statically programmed emerging devices such that they do not morph dynamically during runtime. Nonetheless, they offer superior SAT-attack resiliency and thus do not limit the scope of the obfuscation. When the emerging devices presented in the work [9], [19] has to be used for the IPs that cannot tolerate the error, we program them in static version and as discussed in Section II-B they can be replaced with LUT of size 2 and the key can be recovered in few minutes. Lastly, while primitives proposed in [9], [19] provides at most 16 logical functions, the LUT used in RIL-block can be increased to increase the SAT-hardness of the resulting RIL-Block.

Furthermore, as discussed earlier, any single countermeasure against a given attack is not sufficient. Thus as a defense-in-depth approach, the Scan Enable Obfuscation Method helps in defeating the SAT-attack altogether. As the SAT-attack requires the attacker to access the oracle circuit, he/she must enable the Scan Enable signal in the oracle circuit to apply the test vector and get the corresponding responses. As discussed in Section III-C, the responses received are obfuscated by our proposed MRAM-based LUT, and thus, it helps in thwarting all oracle-based attack such as original SAT-attack, SMT-attack, CycSAT, and AppSAT. As the RIL-Blocks, replaces the gates and their interconnects, and removal of the RIL-blocks does not benefit the attacker in any way.

C. Resiliency against ScanSAT and Scan and Shift attack

As discussed in the ScanSAT, which targets the obfuscated scan chain designs, it aims to convert the circuit that inverts the scan flip-flop responses as a part of an SAT-problem. This allows the SAT-solver to find the key which is responsible for output inversions. However, in our proposed RIL-block, the circuitry that handles the inversion is not part of the scan chain. To further support our claim, let's consider a single reconfigurable block that replaces the OR gate in the circuit with the 2-input MRAM-LUT and adds the SE enabled obfuscation circuitry on top of the MRAM-LUT as proposed in Figure 4. If the key in the MTJ is '0', the obfuscation circuitry is enabled, and this gate effectively behaves as NOR gate on the application of Scan Enable signal, and there is no way to distinguish *if the LUT is configured as OR and negation is provided by Scan Enable ofuscated circuitry* **or** *LUT is configured as NOR*. For Scan and Shift attack, the key values are saved by the Secure Cell (SC), and one can have a separate scan chain in place for the SCs, which can shift-in the keys for configuring the emerging devices and the scan out of this circuitry can be blocked.

TABLE III
TIME TAKEN BY THE SAT-SOLVER IN SECONDS TO FIND THE CORRECT KEY WITH DIFFERENT SIZES AND QUANTITY OF RIL-BLOCKS FOR DIFFERENT ISCAS AND CEP BENCHMARKS

Benchmark Suite	Benchmark Circuit	Number of RIL-Block			AppSAT Success
		1	2	3	
ISCAS-89	b15	124.25	546.2	∞	✘
	s35932	105.1	1864.2	∞	✘
	s38584	345.2	∞	∞	✘
	b20	240.4	2454.26	∞	✘
CEP	AES	1060.56	∞	∞	✘
	SHA-256	846.87	∞	∞	✘
	MD5	1450.1	∞	∞	✘
	GPS	∞	∞	∞	✘

D. Reliability and Resiliency against Power Side Channel Attack

To analyze the reliability of the read and write operations of the proposed RIL-block in the presence of PV and to analyze the P-SCA mitigation and resiliency of our proposed method, Monte Carlo (MC) simulation is performed. The simulation helps in covering a wide range of PV scenarios that may occur in the fabricated device. The MC simulation is performed with 100 instances considering the effects of PV on the CMOS peripheral circuit and the MTJs. In particular, the variation of 1% for the MTJs' dimensions along with 10% variation on the threshold voltage and 1% variation on transistors dimensions are assessed [13]. As mentioned earlier, utilizing complementary STT-MRAM memory cells will result in a symmetrical power footprint for the read and write operations and provides near-zero standby power. Thus, our proposed design maintains near-zero power variation in the output, which can mitigate P-SCAs significantly.

Figure 6(a) shows the distribution of read currents, Figure 6(b) depicts the distribution of read power, Figure 6(c) illustrates the distribution of MTJ resistances in R_{AP} and R_P states for the 100 MC instances of a 2-input MRAM-based LUT implementing an AND gate. According to the MC simulation results, MRAM-based LUT provides reliable write performance resulting in less than 0.01% write errors in 100 error-free MC instances. Additionally, since the states of the MTJs are complementary, they provide a wide read margin, and as a result, there are less than 0.01% read errors caused by PV based on the 100 error-free MC simulation results. Furthermore, it can be noted that the power dissipation for reading '0' and '1' are almost identical as demonstrated in Figure 6(a) and Figure 6(b), which can mitigate P-SCA.

E. Overhead Analysis

There are three energy profiles in the LUT circuits: (1) Read energy consumption during the regular operation, (2) Standby energy for the LUTs that are not on the active datapath, which can constitute a significant portion of the circuit, and (3) write energy that is consumed during the LUTs' configuration operation which occurs rarely. Table IV lists show the energy consumption results of the proposed MRAM-based LUT. The results show significantly reduced standby energy in the order of Attojoules at the cost of increased write energy consumption compared to SRAM-LUT. However, this

Authorized licensed use limited to: Univ of Calif Davis. Downloaded on October 05,2022 at 21:44:15 UTC from IEEE Xplore. Restrictions apply.

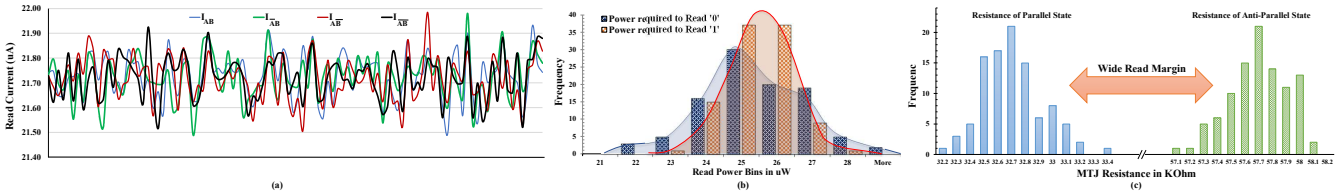


Fig. 6. Results of 100 MC instances of 2-input MRAM-based LUT implementing an AND gate: (a) read currents, (b) read power, and (c) MTJ resistances.

TABLE IV
ENERGY CONSUMPTION OF THE PROPOSED MRAM-BASED LUT.

		Energy Consumption		
		Read	Write	Standby
MRAM-based LUT	Logic "0"	12.47fJ	34.45fJ	36.90aJ
	Logic "1"	12.50fJ	34.94fJ	36.90aJ
	Average	12.48fJ	34.69fJ	36.90aJ

energy overhead can be tolerated due to the infrequent occurrence of write operations in LUTs. Assuming both MRAM-based LUT and SRAM-LUT use the same select tree MUX, the structure of a 2-input MRAM-based LUT requires 32 MOS transistors plus 4 MTJs in each memory cell, while the conventional 2-input SRAM-LUT memory cell includes 24 MOS transistors in each memory cell. Additionally, MTJs can be fabricated on top of the baseline CMOS transistors and thus, incurring low area overhead. Innovations are sought to reduce the area and energy consumption of the MRAM cells' read and write circuit. Recently, SHE-MRAM cells have attracted considerable attention as an alternative for the conventional STT-MRAMs [13]. The write circuit does not scale with the increase in the number of LUT inputs since we can use the same write circuit for all LUT memory cells. Thus, it is worth noting that increasing the LUT size helps to reduce the overhead while increasing SAT-resiliency.

F. Comparison

Table V compare the proposed RIL-blocks against state-of-the-art hardware security primitives. While SFLL can provide SAT-resistance, it has shown vulnerability to [4], [5]. Moreover, GHSE/MESO have their disadvantages in terms of applicability. InterLock offers the SAT-resiliency but requires a large 64×64 interconnect, where each switching element consists of a 2-input LUT and thus, incurs more significant overhead than RIL-block. Finally, InterLock and CAS-block, while SAT-resilient, do not offer dynamic morphing and thus cannot eliminate the SAT-attack. Among all the listed primitives, none of the techniques aims to thwart the P-SCA, while RIL-blocks addresses most of the current hardware security threats and uses a defense-in-depth approach with lower overhead and is better prepared for zero-day attacks.

TABLE V
COMPARISON OF RIL-BLOCK WITH EXISTING HARDWARE SECURITY

Attacks	SFLL [3]	GHSE/MESO [9], [19]	InterLock [11]	CAS-Lock [6]	LUT [12]	Proposed
SAT-attack	✓	✘[2]	✓	✓	✓	✓
AppSAT	✓	✓	✓	✓	✓	✓
Power side channel attack	-	-	-	-	-	✓
Removal attack	✓	-	✓	✓	✓	✓
ScanSAT	-	-	-	-	✓	✓
Shift and Scan attack	-	-	-	-	✓	✓
Features	SFLL	GHSE/MESO	InterLock	CAS-Lock	Proposed	
Dynamic morphing	✘	✓	✘	✘	✘	✓
Application	-	Limited	-	-	-	-

V. CONCLUSION

In this work, we introduced Reconfigurable Interconnect and Logic Blocks, which combines the LUT-based logic obfuscation and interconnect obfuscation and helps in creating the SAT-hard instances. The mix of the two helps increase the number of DPLL calls, which the SAT-solver has to prune for finding a key. We successfully demonstrate this effect by scaling the size of RIL-block, and a few 8×8×8 helps in achieving timeout states. Further, the security of the proposed primitive is reinforced by leveraging MRAM-based LUT, which allows the RIL-blocks to be reconfigured during runtime while maintaining small overheads compared to SRAM-based LUTs. The complementary style of MTJ reconfiguration helps increase the reliability in the presence of PV while mitigating the threat of the P-SCA. Finally, the Scan Enabled obfuscation circuitry helps in obfuscating the oracle circuit's responses by adding a 2-to-1 MUX controlled by a separate MTJ, thus adding an extra layer of security. Thus, the lightweight RIL-block has a higher output error as well as higher output corruptibility and can thwart both Approximate and Removal attacks.

REFERENCES

[1] M. Yasin and *et al.*, "On improving the security of logic locking," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 9, pp. 1411–1424, 2016.

[2] P. Subramanyan *et al.*, "Evaluating the Security of Logic Encryption Algorithms," in *Int'l Symp. on Hardware Oriented Security and Trust (HOST)*, 2015.

[3] M. Yasin and *et al.*, "SFLL-HLS: Stripped-functionality logic locking meets high-level synthesis," in *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2019, pp. 1–4.

[4] D. S. *et al.*, "Functional analysis attacks on logic locking," in *2019 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2019.

[5] F. Yang and *et al.*, "Stripped functionality logic locking with hamming distance-based restore unit (SFLL-HD)–Unlocked," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 10, 2019.

[6] B. Shakya and *et al.*, "CAS-Lock: A security-corruptibility trade-off resilient logic locking scheme," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 175–202, 2020.

[7] K. Z. Azar and *et al.*, "SMT Attack: Next generation attack on obfuscated circuits with capabilities and performance beyond the SAT attacks," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2019, no. 1, pp. 97–122, Nov. 2018.

[8] G. Kolhe and *et al.*, "On custom LUT-based obfuscation," in *Proceedings of the 2019 on Great Lakes Symposium on VLSI*, ser. GfLSVLSI '19. Association for Computing Machinery, 2019, p. 477–482.

[9] N. Rangarajan and *et al.*, "Opening the doors to dynamic camouflaging: Harnessing the power of polymorphic devices," *IEEE Transactions on Emerging Topics in Computing*, p. 1–1, 2020.

[10] H. M. Kamali and *et al.*, "Full-lock: Hard distributions of SAT instances for obfuscating circuits using fully configurable logic and routing blocks," in *Proceedings of the 56th Annual Design Automation Conference 2019*, ser. DAC '19. NY, USA: Association for Computing Machinery, 2019.

[11] H. M. Kamali and et al., "Interlock: An intercorrelated logic and routing locking," in *Proceedings of the 39th International Conference on Computer-Aided Design*, ser. ICCAD '20. New York, NY, USA: Association for Computing Machinery, 2020.

[12] G. Kolhe and *et al.*, "Security and complexity analysis of LUT-based obfuscation: From blueprint to reality," in *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2019, pp. 1–8.

[13] S. Salehi and *et al.*, "Clockless spin-based look-up tables with wide read margin," in *Proceedings of the 2019 on Great Lakes Symposium on VLSI*, ser. GLSVLSI '19, New York, NY, USA, 2019.

[14] I. Kuon, R. Tessier, and J. Rose, "FPGA architecture: Survey and challenges," *Foundations and Trends in Electronic Design Automation*, 2008.

[15] R. Zand, A. Roohi, S. Salehi, and R. F. DeMara, "Scalable Adaptive Spintronic Reconfigurable Logic using Area-Matched MTJ Design," *IEEE Transactions on Circuits and Systems II: Express Briefs, , revision pending*, 2015.

[16] J. Yang, X. Wang, Q. Zhou, Z. Wang, H. Li, Y. Chen, and W. Zhao, "Exploiting spin-orbit torque devices as reconfigurable logic for circuit obfuscation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 1, pp. 57–69, 2019.

[17] X. Cui, X. Li, M. Zhang, and X. Cui, "Design of the rram-based polymorphic look-up table scheme," *IEEE Journal of the Electron Devices Society*, vol. 7, pp. 949–953, 2019.

[18] K. Fazekas, "On SAT-based solution methods for computational problems," Ph.D. dissertation, Johannes Kepler University Linz, 2020.

[19] S. P. *et al.*, "Advancing hardware security using polymorphic and stochastic spin-hall effect devices," in *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2018, pp. 97–102.

[20] J. Kim and *et al.*, "A technology-agnostic MTJ SPICE model with user-defined dimensions for STT-MRAM scalability studies," in *2015 IEEE Custom Integrated Circuits Conference (CICC)*. IEEE, 9 2015.

Authorized licensed use limited to: Univ of Calif Davis. Downloaded on October 05,2022 at 21:44:15 UTC from IEEE Xplore. Restrictions apply.