Proceedings of the International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT 2023)
IEEE Xplore Part Number: CFP23CV1-ART; ISBN: 978-1-6654-7451-1

2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT) | 978-1-6654-7451-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/IDCIoT56793.2023.10052782

Importance of Logic Locking Attacks in Hardware Security

[1]Ashika S V
Department of Electronics and Communication Engineering
Karunya Institute of Technology and Sciences
Coimbatore, India
ashikas@karunya.edu.in

[2]N.M. Sivamangai
Department of Electronics and Communication Engineering
Karunya Institute of Technology and Sciences
Coimbatore, India
nmsivam@gamil.com

[3]R Naveenkumar [A, B]
Department of Electronics and Communication Engineering
[A] Research Scholar- Karunya Institute of Technology and Sciences
[B] Assistant Professor- Karpagam Academy of Higher Education
Coimbatore, India
naveentamil256@gmail.com

[4]Napolean A
Department of Electronics and Communication Engineering
Karunya Institute of Technology and Sciences
Coimbatore, India
nepojustin@gmail.com

***Abstract*- A rise in the number and devastating capability of hardware-based assaults has brought attention to the necessity of protecting the hardware root of trust alongside improvements in power, cost, performance, and reliability. The whole design of an integrated circuit can be concealed from a suspect foundry or end- user via a key-based circuit obfuscation or logic-locking approach. The method is based on introducing "key" input bits into the circuit to introduce ambiguity within the original circuit, rendering the circuit unreadable without the proper secret key. The present level of knowledge in this developing area is reviewed in this study, which also includes a threat model classification such as hardware Trojans, reverse engineering (RE) and side channel analysis. Moreover, the traditional and strong logic locking techniques and its efficiency in terms of area, power, delay is reviewed in hardware-based attacks.**

Keywords - Logic Locking, SAT attack, Anti-SAT attack, Hardware Security

I. INTRODUCTION

Threats to hardware security can appear at any point in the semiconductor life cycle, from specification through fabrication to recycling. They may be the result of unintended design defects, unexpected system side effects, or maliciously deliberate design alterations[1]. to create effective security mechanisms, it is crucial to first understand the various hardware security threats. Traditional hardware security threats including hardware Trojans, reverse engineering (RE) and side channel are continually advancing lethal attacks [2].

Different applications sharing functional units and using rapid and slow execution paths creates side and covert channels that attackers can use. With real and idealised attacks already exceeding the 100-bps limit, covert and side channel capacity are continuing to grow. It is impossible to presume that newly functional units are immune to side or covert channel vulnerabilities. There are numerous functional units that don't have visible attacks but do help with the fast and slow test steps that might develop into side channels and covert channels in the future. Software defences that are now available include changing the state of various functional units, resource partitioning, generating noise, and changing application scheduling to prevent contention [3]. Hardware Trojans are designed to unlock secure devices and access their data. The benefit of hardware Trojans is that they can access a complete batch or series of chips by altering the IC's design or manufacturing process. Reverse engineering (RE) and side channel analysis (SCA) are other methods for breaking into a secure device, however they do not scale well for many devices. To perform the modifications required for the analysis on each device, there is a sizable overhead. However, alterations are only injected once during design or fabrication in the case of hardware Trojans, making it easier to attack them [4]. In order to reduce the effect of Trojan inputs on circuit side-channel signals like power and delay, nets with low transition probabilities are typically used to supply Trojan inputs. The netlist of the Trojan-free circuit is used by automatic test pattern generation (ATPG) techniques used in manufacturing tests to find flaws. Therefore, Trojans cannot be directly targeted by ATPG algorithms currently in use [5].

An IC is reverse engineered by determining the device technology utilised in it, obtaining its gate-level netlist, and/or determining its functionality. Reverse engineering ICs has been made possible by a variety of methods and tools. RE can be abused to identify the technology of the target device, illicitly construct the target IC, and/or steal and/or pirate a design. Reverse engineering a design to the necessary level of abstraction is the attacker's objective. By using the specified input pairs to verify the operational viability of the reverse-engineered design, he can direct reverse engineering (RE) to extract the gate-level netlist of a competitor's IP and use it in one's own IC or illegally advertise it as an IP [2].

To reveal Trojan impact on design attributes beyond process and environmental variables, efficient pattern creation is required following Trojan discovery. To maximize the Trojan contribution to the circuit power consumption, Trojan detection systems using transient power analysis require patterns that boost Trojan activity while maintaining low circuit activity[5].

This paper organises the knowledge for a number of significant modern hardware security issues within this context. To assess the efficacy of the developed defences,

it categorises hardware-based threats and logical locking techniques.

II. HARDWARE TROJANS

HTs are characterised as malicious, purposeful additions, deletions, and modifications or unintentional design flaws in intellectual property (IP) cores or integrated circuits (ICs) that can be used by skilled adversaries to further an attacker's goals, potentially causing enormous financial losses and serious societal harm [6]. The design can accommodate a variety of Trojans with various activation mechanisms (triggers) and outcomes (payloads). A typical HT construction can be seen in Fig. 1, which includes the payload, Trojan circuit, and trigger. The HTs are often made to remain silent for the majority of the time in order to be concealed in ICs, and their triggers are connected to uncommon signals or occurrences. The payload circuit is activated and performs harmful actions when the predefined signal or event occurs. Because of creative design, uncommon signals or occurrences are more likely to occur during long periods of field operation than during design simulation or testing [7]. The IC life cycle might include the implementation of HT at various stages. It is possible for HT risks to arise during interactions between any two of the parties in the IC market model, including: i) foundries and SoC designers; ii) IP vendors and SoC designers; iii) IP vendors and EDA vendors; and iv) end users and SoC designers.

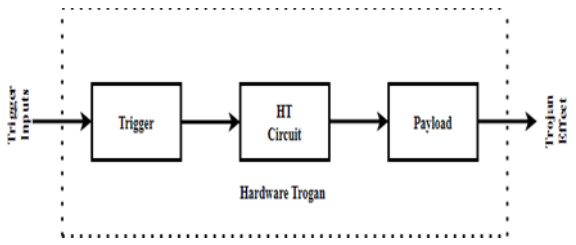

Fig. 1. Hardware Trojans Structure

The following are some of the detection methods ofHardware Trojans:

1) Presilicon oriented Countermeasures: Techniques for detecting HTs at the presilicon stage of design are available. These methods include security verification, structural inspection, and switching probability analysis [8].

Security verification: Certain kinds of HTs can be found via security verification. In order to prove security qualities like confidentiality and integrity, it derives formal security models for hardware designs and uses formal methods like SAT solving, model checking, and type checking. An inadvertent design defect or maliciously modified design is indicated by a security property violation. Formal verification, however, frequently encounters scalability issues and frequently only effectively detects HTs at the IP level.

Structural inspection: The goal of structural checking-based HT detection approaches is to identify structural characteristics unique to HT designs, such as gate type, gate count, and interconnection patterns, and then perform detection using these characteristics. To identify Trojan circuitry, they typically employ a scoring algorithm to compare such attributes to the circuit topologies being tested. These approaches, however, may give false- positive results and have scaling problems.

Changing the probabilistic approach: The concept that such Trojans trigger signal should get a very low switching potential in order to prevent HT from being triggered repeatedly has been the foundation for the development of HT detection systems. Through structural analysis or behavioural code analysis, these techniques attempt to pinpoint the signals of the behaviour of switching activity that are noticeably below average. These studies have demonstrated the intimate relationship between Trojan circuitry and low controllability or observability signals.

2) Post silicon-oriented Countermeasure: After chip production, presilicon HT detection techniques look for harmful design alterations. A popular method for post silicon HT detection is destructive RE, which entails delayering and depackaging the ICs as well as recovering the circuit structure from layout photos. However, if HT is only placed into a few chips, this time-consuming and costly process could fail. On the other hand, non-destructive techniques like functional testing and SCA are often thought to be more practical.

Functionality testing: aims to trigger the Trojan systems infrequently used circuitry and deliver the result to a visible area. An interesting research vector is how to create test vectors that can activate each node in a circuit that switches seldom. [9]. Statistical techniques have suggested one possible answer, while additional attempts involve assisted assessments against HTs in key design areas.

Side-Channel attack (SCA): based methods that examine physical IC metrics including power usage, route delay, and chip emissions are used to identify HTs. The absence of a gold chip and the impact of process variation on measurements SCA are these techniques' primary drawbacks. To increase HT detection sensitivity, researchers employ a number of SCA parameters or merge logic testing with SCA to guided pattern formation. By mimicking the golden signature or contrasting signatures collected at various times, there are additional attempts to generate "golden-free" solutions.

3) Trust-Based Design (TBS) Techniques: TBS approaches incorporate dedicated logic to aid with HT identification. Several ways enhance the internal node controllability and observability to include dummy flip flips and testing sites to speed up the Trojan triggering. To enable production screening, on-site monitoring of HT-infected chips, or to support SCA-based HT detection, another type of TBS approach incorporates circuit infrastructure, like as ring oscillators and current sensors. TBS techniques can also be used to render HT insertion harder while enhancing detection.

4) Third-party IPs (3 PIP): The bulk of HT identification systems evaluate their effectiveness using Trojan benchmarks [9].Because of the absence of understanding about the Trojan implementation, detecting unexpected HTs in 3PIP is more difficult. This method is made more complex by functional testing's scope, switching probability analysis's location, and process fluctuation's noise [10].The detection rate of functional and security verification procedures is depending on the quality of the attributes. Unfortunately, 3PIP finds it difficult to specify the appropriate property for Trojan detection. If the Trojans' features are part of the feature library, methods for recognizing HTs by comparing reliable design structural

or data flow characteristics to earlier templates.

Prevention strategies for HT:

Making HT insertion harder or impossible, is the aim of HT preventive strategies. Three common Trojan protection approaches are logic-obfuscation, structural obfuscation and split-manufacturing.

A. Logic obfuscation techniques, which were originally advocated for IP protection. Without the right key, the obfuscated circuit's functions is tied to the unreliable foundry. This makes HT implantation challenging. By introducing additional gates into the original design, logic locking conceals the functionality and implementation of a system. A proper key must be supplied to the locked design in order for it to function correctly (i.e. create correct outputs). The key-gates are the gates inserted for locking. When the incorrect key is used, the locked design will demonstrate incorrect functionality (i.e. create incorrect outputs). XOR/XNOR and MUX-based logic locking are common logic obfuscation techniques that can influence the hardware flow of information or the values of internal circuit nodes. Similarly, logic obfuscation can be achieved by providing programmable parts that retain a portion of the logic for later modification. However, iteratively searching for distinguishing input patterns (DIPs) in functional oracle-guided SAT assaults to weed out the incorrect keys can exploit these obfuscation techniques [11]. A number of SAT- resistant strategies, include Anti-SAT block (ASB), increased the number of SAT iterations necessary to make the SAT attack is unfeasible [12].

B. Split manufacturing can aid in preventing nefarious design alteration. It divides a methods for improving into components at the front end of the line (FEOL) and the back end of the line (BEOL), it will be produced by reputable foundries, unreliable foundries. Without knowledge of the BEOL component, it is challenging to the unreliable foundry for include a meaningful HT.

By altering the design structure, structural obfuscation hides the functioning of the structure and makes it difficult for an adversary to understand. This makes RE more challenging, preventing the installation of malicious software or Trojans. By considering the trade-off between design characteristics like area, latency, and Parhi and Lao discuss how power changes throughout major transformations were able to defend DSP circuits that underwent significant structural changes due to RE and HT attacks without losing their approaches applied .

III. LOGIC LOCKING

Logic locking encrypts the originality with a password by putting additional logic in a circuit. A locked circuit, as depicted in Fig. 2, has key inputs in addition to the existing inputs, which are powered by an on-chip tamper-proof memory.

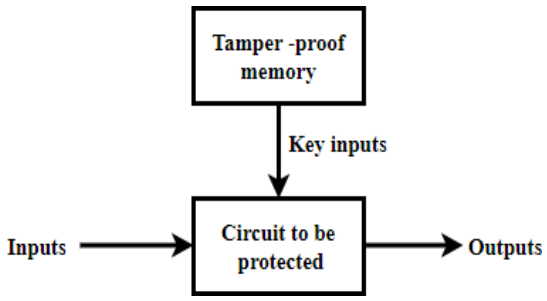

Fig.2 An illustrative example of a logic-lockedarchitecture

The design only yields accurate results when the secret key is used; otherwise, inaccurate results are provided.

Look-up tables (LUTs) or XOR key gates may make up the additional logic. The untrusted design phases are completed with the locked netlist. Without the secret key,(i) It is impossible to recover the design specs (for RE), (ii) The IC is not working properly since it generates the wrong outputs, and (iii) the secret key cannot be activated by loading it onto the chip's memory (for over-production). The secret key needs to be stored in the chip's memory in order to unlock a locked IC [13]. Table I shows about different locking methods and its effect against resistance approaches.

TABLE I: LOGIC LOCKING APPROACHES' RESISTANCE TO ATTACK.

Attack	RLL [15]	FLL [11, 16]	SLL [14]	Anti-SAT [12]	SAR [18]	SFLL [20]	Compound locking
Sensitization	✓	✓	×	×	×	×	×
SAT	✓	✓	✓	×	×	×	×
AppSAT	✓	✓	✓	×	×	×	✓*
Double-DIP	✓	✓	✓	×	×	×	✓*
SPS	×	×	×	✓	✓	×	×
AGR	×	×	×	✓	✓	×	✓
Bypass	×	×	×	✓	✓	×	×
Test-data mining	✓	✓	✓	×	×	×	×
DPA	✓	✓	✓	×	×	×	×
Desynthesis	✓	✓	✓	×	×	×	×

✓Signifies an effective attack,× Indicates resiliency, and ✓* represents a partial success [13].

The two most common types of logic locking are conventional logic locking and SAT attack resilient logic locking.

Traditional logic locking: The methods in this group were concentrated on creating effective algorithms for choosing the key gate positions. Strong logic locking [14], random [15], and fault analysis based [11] are examples of classic logic locking approaches. Although combinational elements are used in the majority of procedures, sequential logic locking techniques [16] are also available.

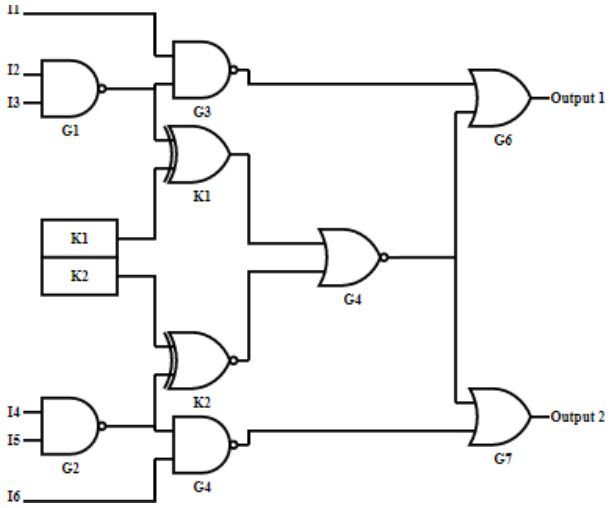

Fig.3 Key insertion based on Strong logic locking

By adding key gates with the greatest possible mutual interference and preventing the individual key bits from becoming sensitized, SLL thwarts the sensitization attack. A potential attacker is compelled to treat many key bits collectively rather than individually since the sensitization of key bits is hindered. The netlist in Fig. 3 is an example. Using the SLL algorithm, the netlist has two key-gates, Key A and Key B. The paths of Key A and Key B to the main outputs are shown to be interfered with by one another. A primary output cannot be individually sensitized by an attacker to Key A and Key B.

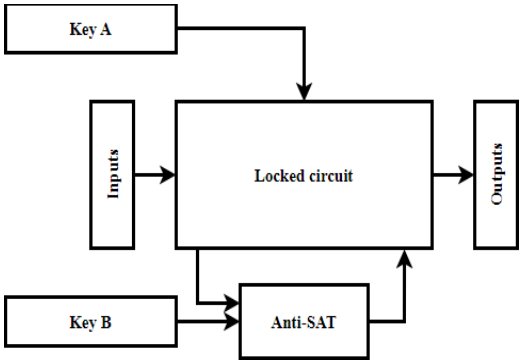

Fig.4 Anti-SAT Circuit Block

Strong logic locking against SAT attack:

Using a small number of carefully chosen input patterns and their accurate outputs as seen from an active functional chip, the SAT attack's crucial insight is to infer the proper key. When the SAT attack was developed [17], it completely changed the direction of logic locking research because it was capable of defeating all established methods. SARLock [18], Anti-SAT [12], TTLock [19], and SFLL [20] are recent research projects that concentrate on preventing the SAT attack.

Table II tells about the performance overhead of logic locking techniques in terms of area, power and delay.

When the key vector is properly set, the output of an Anti-SAT block is a constant (e.g., always equals to 0). Otherwise, depending on the inputs, output can either be 1 or 0. It can be incorporated into the original circuit thanks to this characteristic. The wires in the original circuit serve as the inputs of Anti-SAT block, as shown in Fig. 4. An XOR gate is used to connect the output to the original circuit.

TABLE II. PERFORMANCE OVERHEAD OF LOGIC LOCKING TECHNIQUES. ↓ DENOTES LOW, ↑ DENOTES HIGH

Logic locking techniques	Area	Power	Delay	Ref
RLL	↓	↓	↓	[15]
FLL	↑	↑	↑	[11][16]
SLL	↓	↓	↑	[14]
Anti-SAT	↓	↓	↓	[12]
SAR Lock	↑	↑	↓	[18]
SFLL	↓	↑	↑	[20]
Compound Locking	↓	↓	↑	[18]

An innovative method is put out by Roy et al. to stop the theft of integrated circuits (EPIC). Here, each chip creates a unique random identification number using methods that are commonplace. According to the evaluation, the overhead of EPIC on circuit delay and power is minimal, and the conventional test and verification workflows do not need to be altered [15].

FLL tries to stop an IC from being used as a black box. Even erroneous keys may result in the right output in random logic locking for specific input patterns. When the wrong keys are utilized, FLL guarantees optimum corruption at the output bits. The % Hamming distance between the proper output and the incorrect output, which was produced by using the wrong keys, is used to measure output corruption. Smaller designs (2000 gates) like C432, S510, and S641 have very high overhead in FLL since even a modest number of additional XOR gate 30 needed for logic encryption is on the order of the whole number of gates necessary to build these tiny circuits. On the other hand, the overhead for fault analysis-based approaches (both XOR and MUX) in big systems (>3000 gates) is less than 5%. This demonstrates how very viable and low-overhead fault analysis-based encryption is, particularly for bigger architectures [11].

Strong Logic Locking (SLL) has larger area, power, and delay overheads than the currently used methods. For instance, 1% key-gates in SLL circuits often have an area overhead of 2.94% as opposed to 2% in mux-based locking. Among logic locking approaches, SLL has the biggest delay overhead. This is due to the fact that SLL, in its current configuration, places the maximum amount of key-gates in specific circuit regions, which raises the delay overhead by two to three times [14].

The method put out by Rathor. V. S. et al. makes use of the circuitry that already exists to design and incorporate the ASB, which also lowers overhead. The ASBs, which use 50% existing circuitry in their design (excluding key-key gates), can eliminate an average of 38.4 transistors [12].

SAT Attack Resistant Logic Locking (SARLock) can be implemented selectively to only the essential components of the design to save overhead. The most valuable intellectual property (IP) in processors is often found in the controllers, which also take up very little space (less than 1%). SARLock+SLL has an average area, power, and delay

overhead that is similar to RLL and SLL at 35.2%, 61%, and 9.3%, respectively [18].

The versatility of stripped-functionality logic locking (SFLL) allows the designer to secure any number of cubes inside a specific set, resulting in a straightforward and scalable architecture for broad applications. Additionally, it supports specialized applications that need to secure IP-critical input cubes [20].

The main advancements in the field of logic locking are outlined in this paper. Initially focusing on effective gate selection techniques, logic locking research has steadily moved its attention to concurrently fending off SAT and structural attacks. It is crucial to compare a logic locking method to all existing ones.

IV. CONCLUSION

This article provides a survey of current developments in a few hardware security subfields. In particular, concepts for hardware Trojans, SCA, and RE threats are described. It also describes current, fast expanding work in hardware trojan detection and countermeasure techniques. Finally it focus about different logic locking techniques such as conventional logic locking and SAT attack resilient logic locking that are used recently and their efficiency in terms of area, power and delay.

REFERENCES

[1] W. Hu, C. -H. Chang, A. Sengupta, S. Bhunia, R. Kastner and H. Li, "An Overview of Hardware Security and Trust: Threats, Countermeasures, and Design Tools," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 40, no. 6, pp. 1010-1038, June 2021, doi: 10.1109/TCAD.2020.3047976.

[2] Massad, Mohamed El, Siddharth Garg, and Mahesh Tripunitara. "Reverse engineering camouflaged sequential integrated circuits without scan access," arXiv preprint (2017), arXiv:1710.10474.

[3] Szefer J, "Survey of Microarchitectural Side and Covert Channels,Attacks, and Defenses," Journal of Hardware and Systems Security, 3(3), pp.219-234, Sep 2019.

[4] Naveenkumar, R, N. M. Sivamangai, A Napolean and V. Janani. "A Survey on Recent Detection Methods of the Hardware Trojans." 3rd International Conference on Signal Processing and Communication (ICPSC) (2021): 139-143. https://doi.org/10.1109/ICSPC51351.2021.9451682

[5] Salmani H, Tehranipoor M, and Plusquellic J, "A novel technique for improving hardware trojan detection and reducing trojan activation time," IEEE transactions on very large scale integration (VLSI) systems, 20(1), pp. 112-125, Jan 2011 .

[6] Huang Z, Wang Q, Chen Y, and Jiang X, " A survey on machine learning against hardware trojan attacks: Recent advances and challenges," IEEE Access, 8, pp.10796-10826, Jan 2020.

[7] Li He, Qiang Liu, and Jiliang Zhang. "A survey of hardware Trojan threat and defense." Integration 55, pp. 426-437 Sep 2016.

[8] Haider S. K, Jin C, Ahmad M, Shila D. M , Khan O, and van Dijk M, "Advancing the state-of-the-art in hardware trojans detection," IEEE Transactions on Dependable and Secure Computing, 16(1), pp. 18-32, Jan 2017.

[9] Huang Y, Bhunia S, and Mishra P, "Scalable test generation for Trojan detection using side channel analysis," IEEE Transactions on Information Forensics and Security, 13(11), pp. 2746-2760, May 2018.

[10] Chen X, Liu Q, Yao S,Wang J, Xu Q, Wang Y and Yang H, "Hardware trojan detection in third-party digital intellectual property cores by multilevel feature analysis," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 37(7), pp. 1370-1383, Aug 2017.

[11] Rajendran J, Zhang H, Zhang C, Rose G. S, Pino Y, Sinanoglu O, and Karri R, " Fault analysis-based logic encryption," IEEE Transactions on computers, 64(2), pp. 410-424, Oct 2013.

[12] Rathor V. S, Garg B, and Sharma, G. K, "New lightweight Anti-SAT block design and obfuscation technique to thwart removal attack," Integration, 75, pp. 178-188, Nov 2020.

[13] Naveenkumar R, N. M. Sivamangai, A Napolean , A. Puvirasu and G. Saranya, "Preventive Measure of SAT Attack by Integrating Anti-SAT on Locked Circuit for Improving Hardware Security," 7th International Conference on Communication and Electronics Systems (ICCES), pp. 756-760,Jun2022.https://doi.org/ 10.1109/ICCES54183.2022.9835923.

[14] Yasin M, Rajendran J.J, Sinanoglu O, and Karri R, "On Improving the Security of Logic Locking," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 35, pp. 1411-1424, Dec 2015.

[15] Roy, Jarrod A, Koushanfar, Farinaz and Markov, "EPIC: Ending piracy of integrated circuits," Proceedings of the conference on Design, automation and test in Europe, pp. 1069-1074, Mar 2008.

[16] Baumgarten A, Tyagi A, and Zambreno J, "Preventing IC piracy using reconfigurable logic barriers," IEEE design & Test of computers, 27(1), pp. 66-75, Feb 2010.

[17] Subramanyan P, Ray S, and Malik S, "Evaluating the security of logic encryption algorithms," IEEE International Symposium on Hardware Oriented Security and Trust (HOST), pp. 137-143, May 2015.

[18] Yasin M, Mazumdar B, Rajendran J. J, and Sinanoglu O, "SARLock: SAT attack resistant logic locking," IEEE International Symposium on Hardware Oriented Security and Trust (HOST), pp. 236-241, May 2016.

[19] Naveenkumar R, N. M. Sivamangai, A Napolean and G. Akashraj Nissi, "Hardware Obfuscation for IP Protection of DSP Applications," Journal of Electronic Testing, 38(1), pp.9-20, Feb 2022. https://doi.org/10.1007/s10836-022-05984-2

[20] Yasin, M, Sengupta A, Nabeel M. T, Ashraf M, Rajendran J, and Sinanoglu O, "Provably-secure logic locking: From theory to practice," Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 1601- 1618, Oct 2017.