International Journal of Research in Engineering, Science and Management
Volume-2, Issue-2, February-2019
www.ijresm.com | ISSN (Online): 2581-5792

Resilient DFS Architecture for Enabling Trust in IC Manufacturing and Testing

M. Agila[1], S. Krithiga[2]
[1]*M.E. Student, Dept. of Applied Electronics, Thanthai Periyar Govt. Institute of Technology, Vellore, India*
[2]*Professor, Department of ECE, Thanthai Periyar Govt. Institute of Technology, Vellore, India*

***Abstract*: Due to the prohibitive costs of semiconductor manufacturing, most system-on-chip design companies outsource their production to offshore foundries. As most of these devices are manufactured in environments of limited trust that often lack appropriate oversight, a number of different threats have emerged. These include unauthorized overproduction of the integrated circuits (ICs), sale of out-of-specification/rejected ICs discarded by manufacturing tests, piracy of intellectual property, and reverse engineering of the designs. Over the years, researchers have proposed different metering and obfuscation techniques to enable trust in outsourced IC manufacturing, where the design is obfuscated by modifying the underlying functionality and only activated by using a secure obfuscation key. However, Boolean satisfiability-based algorithms have been shown to efficiently break key-based obfuscation methods, and thus circumvent the primary objectives of metering and obfuscation. In this paper, we present a novel secure cell design for implementing the design-for-security infrastructure to prevent leaking the key to an adversary under any circumstances. Importantly, our design does not limit the testability of the chip during the normal manufacturing flow in any way, including postsilicon validation and debug. Our proposed design is resistant to various known attacks at the cost of a very little (< 1%) area overhead.**

***Keywords*: Design for security (DFS), integrated circuit (IC) overproduction, obfuscation, piracy, reverse engineering (RE).**

1. Introduction

Counterfeiting and piracy have become major problems in the twenty-first century due to the globalization of the semiconductor industry. Because of the persistent trend of device scaling and the resulting increase in the complexity of the fabrication process, most companies designing system-on-chips (SoCs) no longer maintain a fabrication unit (foundry or fab) of their own. Costs for building and maintaining such foundries are reported to be more than several billions of dollars. This leads to the adaptation of horizontal integration in the semiconductor industry where the SoC designers contract foundries and assemblies for production. In parallel, the continuous trend of device scaling has enabled designers to fit more and more functionality on an SoC to reduce overall area and cost of a system. As the complexity of modern SoCs grows exponentially, it is virtually impossible to design a complete system by an SoC designer alone. Therefore, the semiconductor industry has shifted gears to the concept of design reuse rather than designing the whole SoC from scratch. Due to the lack of transparency and the resulting lack of trust may lead to the following vulnerabilities.

- *Integrated circuit (IC) overproduction:* An untrusted foundry/assembly can produce more number of unauthorized chips and can make illegitimately larger profits by selling them in the market as no research and development cost is incurred during production. Moreover, they can also practically over-build chips at zero cost by manipulating the yield information.
- *Out-of-specification/defective ICs from manufacturing:* Due to the imperfect manufacturing and assembly processes, foundry/assembly discards defective chips and sends defect free chips to the market. In a trusted environment, these defective chips are always scrapped. However, an untrusted entity in the production process (a rogue employee) can source these rejected defective chips to the gray market. The application of these chips in a critical infrastructure can cause significant damage.
- *Intellectual property (IP) piracy and reverse engineering (RE):* An untrusted foundry or its rogue employee can pirate the details of an SoC (e.g., test patterns and mask information) to a competitor company or make one or more illegitimate copies of the original IPs. The design details of an SoC can be reconstructed from the musk information by RE, which ultimately help to make cloned ICs. An untrusted foundry can also add some extra features to the SoC to introduce a backdoor or a hardware Trojan into these clone chips.

In this paper, we present a novel design-for-security (DFS) architecture to prevent the aforementioned attacks by obfuscating a netlist. The chips must be activated to unlock their full functionality before shipped them to the market. We insert locks in the netlist in such a way that the commercial automatic test pattern generation (ATPG) tools can generate test patterns without having the obfuscation key. This will provide support for performing manufacturing tests before the activation of the chips. We have added a scan flip-flop (FF) to drive a key bit such that an ATPG tool can reach to the obfuscated portion of the circuit. We have provided the support such that an unlocked

circuit (fully functional) blocks the scan out capability when an adversary attempts to dump the functional responses captured in the FFs through the scan chains. Due to the unavailability of scan data that contains the obfuscation key, existing attacks become unfeasible. Note that the chips can be fully functional and structural tests can be carried out with the help of the key only in a secure environment, which can provide post silicon debug and diagnosis support.

A. Contributions

The key contributions of this paper are as follows.

- *Support for test before activation:* Our locked design does not require the obfuscation key during manufacturing tests and allows full scan-based structural manufacturing tests at the potentially untrusted foundry on the obfuscated design. Such structural tests can comprehensively test the circuit for virtually all faults (e.g., stuck-at, transition, and path delay faults), even though the circuit is still locked. A foundry can perform the complete range of manufacturing tests on the locked chips without the need for any change in the normal IC fabrication and test flow.
- *No capture of keys during scan tests:* The chip can also be tested, using both structural and functional tests, by untrusted end users, after the IC has been unlocked by programming the obfuscation key into the chip in a secure environment. This is allowable because our pro-posed solution prevents the capture of any information related to the keys during scan testing performed on even an unlocked chip. Any structural tests applied at this stage still operate on the locked obfuscated design. Basically, the programmed keys are disabled during scan-based tests. However, normal functional tests can obviously be performed on an unlocked chip configured for full functionality. The lack of access to a scan shift capability in conjunction with unlocked full functionality is a design feature that prevents an adversary from using scan to perform satisfiability (SAT)-based attacks on an unlocked IC to recover the key.
- *Disabling scan dump after functional mode:* Further-more, our DFS design blocks any direct transition from functional mode to scan mode. This is also a necessary feature to achieve complete protection against SAT-based attacks. Note that blocking the possibility of a scan dump in the midst of functional operation eliminates the availability of a “golden” functional circuit or “oracle” (an unlocked functional IC) with internal state visibility. In SAT-based attacks, a small set of distinguishing input scan test patterns (DIPs) are obtained from the locked circuit and incorrect keys are ruled out by any observed mismatches when the responses using candidate keys are compared with those from an unlocked functional IC or oracle. However, in the absence of visibility into the many internal FF states of a sequential circuit, any comparison of just the observable output signals provides very minimal information foreach applied input pattern. This dramatically increases the complexity of any SAT-based attack, making it virtually impossible to apply against a large sequential design.
- *Post-Si validation and debug support:* However, blocking any scan dump in the middle of functional operation can greatly complicate design error diagnosis and debug. Observe that logic design bugs and the obfuscation keys have a similar impact on the circuit; they both transform a good functional design into a faulty one. Consequently, preventing discovery of the obfuscation keys while at the same time providing support for logic error discovery and debug is inherently contradictory goals. Our proposed DFS architecture overcomes this problem with a novel design feature that necessitates availability of the actual obfuscation key for scan dump activation in the functional mode; having an unlocked chip is not sufficient. Recall that the key cannot be recovered from an unlocked chip. Thus, design debug, as well as key discovery using SAT or other formal methods, cannot be performed by an untrusted user. However, as described later in this paper, a full design debug capability is supported at a trusted site where the actual obfuscation key is available.

In summary, our DFS architecture is the first secure design that provides complete support for structural manufacturing tests, post silicon validation and debug, and full in-system test capability, all with a very small area overhead. With respect to pinout, our proposed architecture requires only one additional global signal pin (Test).

While studying the robustness of the DFS architecture against SAT-based attacks, we have also developed a novel new attack that can discover the obfuscation keys for a sequential circuit in an efficient manner. This is a second significant contribution of this paper. We call this as the greedy attack. In this attack, an adversary simulates an obfuscated logic cone with just a few random patterns to rule out a hypothesis key. Note that the number of key bits can be very small for a logic cone when the key gates are uniformly distributed though out a sequential circuit, which consists of thousands of cones for a modern design. This is a probabilistic attack, and it cannot guarantee the elimination all possible incorrect key combinations. However, our experimental results show that we can eliminate a hypothesis key with few random patterns in most cases. Furthermore, greedy attack can be performed in combination with SAT-based attacks to efficiently find the key. Fortunately, this attack can also be prevented completely by our proposed DFS architecture

2. Attacks on existing logic obfuscation techniques

Modern electronic designs are sequential in nature and consist of combinational logic and memory elements. The outputs of a sequential circuit depend both on the inputs and its internal state. Generating test vectors to test a sequential circuit is extremely challenging as it is required to initialize the internal state before applying a pattern and then carry the response to the primary output (PO). This leads to adopt scan design, where controllability and observability are provided for the memory elements (FFs). The basic idea of scan is to convert the sequential circuit to its combinational equivalent. Each combinational block can be tested simultaneously through the scan chains. It is now very relevant to analyze the security of the obfuscated sequential circuits. In this section, we present two different attacks that can partially (full) recover the obfuscation key for sequential circuits.

A. Brute Force Attack Based on Logic Cones

For the uniform obfuscation of a net list, it is required to distribute the key throughout the net list such that the circuit produces incorrect result most of the time. This can create a new vulnerability that an adversary can estimate the key by using exhaustive search when a key gate is placed in a smaller cone. We call this attack as brute force attack based on logic cones, which was first introduced by Lee and Touba. Brute force attack is very important to evaluate the security strength of an obfuscated design.

Brute force attacks can be performed through the scan chains, which are inserted into a design to provide manufacturing test support. This insertion of scan chains converts a sequential circuit to its combinational equivalent and contains hundreds/thousands of cones with varying input sizes. If a key gate is placed in a cone with smaller number of inputs, an adversary can perform an exhaustive search to estimate the key value. In order to get a better understanding of brute force attack, it is necessary to analyze attacker's effort (AE), which can be defined as the total number of trials to estimate the key. In this attack scenario, an adversary tries all possible combinations of key and input values of a cone and observes the output of the locked circuit. For a correct key, the output must be equal to the output of that cone of an unlocked functional IC (oracle).

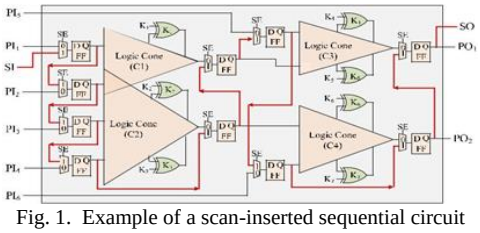

Fig. 1. Example of a scan-inserted sequential circuit

We will now present a short example to describe the complexity of this attack. Fig. 2 shows a sequential circuit, where seven key gates are placed. We assume that the circuit contains four logic cones, namely, C1, C2, C3, and C4 where, C1 and C2 have one overlapping input. The circuit has six inputs and two outputs. For simplicity, we assume that the circuit contains one scan chain (highlighted in dotted red). To find the correct key, an adversary will try all possible combinations. Thus, the AE for C1 (AE_1) will be 2^3.

Table 1
PC in IWLS benchmarks

Bench-mark	# Gates	# Cones	PC ≤ 16	16<PC ≤ 32	32<PC ≤ 64	64<PC ≤ 128	PC >128
S35932	16.065	2066	100.00%	0.00%	0.00%	0.00%	0.00%
S38584	19.253	1332	78.49%	19.88%	1.64%	0.00%	0.00%
S38417	22.179	1559	58.33%	16.25%	9.42%	16.00%	0.00%
b17	37.117	1311	11.06%	4.36%	11.76%	22.43%	50.39%
b18	92.048	3075	6.97%	5.04%	12.91%	14.60%	60.47%
b19	174,157	6074	6.80%	5.14%	12.59%	14.55%	60.92%

Similarly, AE for cones C2, C3, and C4 will be $AE_2= 2^5$, $AE_3= 2^4$, and $AE_4= 2^4$, respectively. It is interesting to note that an adversary can brute force all the cones simultaneously by shifting the appropriate patterns through the scan chain. The number of such scan shift operations (the overall AE) is the $\max(A_i)= 2^5$, which is much smaller than the exhaustive key search (2^{6+7}) to find 7-bit obfuscation key. However, an adversary can find some key bit much quickly if they are placed in a smaller cone (e.g., C1). However, a designer can route one key to multiple cones. For example, the *m*-bit obfuscation key can be distributed into *r* cones, where all cones receive *m*-bit key. In this case, the AE becomes $O(2^{n+m})$. We call this complexity as "worst case" as an adversary cannot perform brute force attacks like previously. Note that the routing congestion will increase significantly if we want to route all the key bits to different targeted cones. A compromise can be made, where a key be can be routed to few cones without increasing the routing congestion. However, due to the large number of cones, a designer can only obfuscate a very small portion of the circuit. Thus, it may be very tempting for him/her to distribute the key in different cones to maximize the effect of obfuscation. In summary, an adversary can perform brute force attacks to all the cones simultaneously through scan chains to estimate the complete *m*-bit key, when the key bits distributed across the circuit. He/she can find a part of key if those keys are placed in a small cone. The strength of the obfuscation depends only on the cone size, rather than the total number of bits in the obfuscation key and the primary inputs (PIs) of a complete net list.

3. Proposed design for security implementation

A. Requirements of DFS implementation

This section provides an in-depth analysis for all the requirements for successfully preventing IC overproduction, manufacturing rejection, and IP piracy.

- *Attack Resistance:* The netlist must be designed in such a way that the chip never leaks the key (during either tests or normal functions), which makes the design resistant to various known attacks. Finding of a key must satisfy NP completeness, and the key must be kept long enough such that brute force attacks

become impractical. In addition, the key must be resistant to RE attack, where an attacker must not find the key by looking at the circuit netlist. Direct mapping of the key bits to XOR or XNOR gates are prohibited.

- *Uniform Distribution of the Key:* The key gates need to be placed uniformly to a design to obfuscate its significant part. As the modern designs are sequential in nature, care needs to be taken to place a key gate. It can be subjected to brute force attacks. It can also be vulnerable to greedy attacks irrespective of the size of the cone. In addition, any cones are subjected to SAT-based attacks. The obfuscation scheme must address all these attacks.
- *Structural Test Capability Without the Key:* Allowing structural tests before the activation is one of the key requirements for preventing the overproduction of chips. It is necessary to add capability which permits a foundry or assembly to perform structural tests right after manufacturing and discard the defective chips. One can argue that tests can be performed at the SoC designer's site. However, it requires additional test setup for the SoC designers, which they may not have. In addition, it is not wise to send chips to the SoC designers without tests which require addition transportation. However, the greater challenge is that the foundry cannot stabilize the process unless they monitor the outcome. Thus, it is absolutely required that the tests have to be performed at the manufacturing site.
- *Post silicon Validation and Debug Capability:* The circuits must be modified in such a way that it does not impact the post silicon validation and debug, where the chips generally run at speed and scan dumps may be required to obtain high observability of internal nodes.

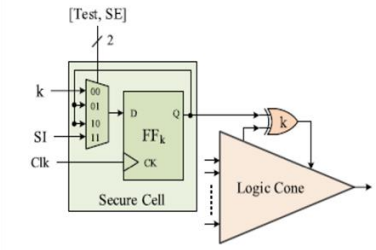

Fig. 2. Proposed SC architecture.

- *Full In-System Test Capability:* The obfuscated circuit must support in-system test capability. It is absolutely necessary that a chip does not leak key information to its POs while it is in functional mode. In this mode, a set of functional test vectors is required to test a design. While testing, it is required that each module (IPs) to be initialized to the desired state. Setting that state of a complex industrial circuit through PIs becomes a major challenge and could potentially take millions of clock cycles. Thus, test engineers often shift the state through existing design-for-test (DFT) structure. It is thus required that keys do not impose any limitation to this hybrid testing.

Table 2
Modes of operation

Test	SE	Mode	Description
0	0	M0	The chip is in functional mode. The secure cell applies key to the logic.
0	1	M1	The secure cell holds its previous value. The rest of the circuit is in functional/shift mode depending on the *SE*.
1	0		
1	1	M2	The SC becomes scan cell and it becomes a part of the scan chain.

B. Proposed Design-for-Security Architecture

The objective in designing the new DFS architecture is to prevent the key getting exposed during manufacturing tests. We have mentioned in Section IV-A that if the key information is captured during a test, it will eventually be exposed to the POs of a working (unlocked) chip and an adversary can effectively retrieve the key our proposed SC architecture used for design for security. We modify the scan cell in such a way that it can hold its previous state. The output of FFk is fed back to the its input through a multiplexer (MUX). As the MUX has four inputs, we need one additional Test pin for the MUX control. Depending on the value of Test and SE pins, a particular input is selected. The key bit (k) and scan in (SI) are connected to the first and fourth inputs of the MUX, respectively. The output of FFk is connected to the second and third inputs, which provides the capability to hold its previous state. The SC operates in three different modes based on Test and SE, which is shown. In mode M0, FFk captures the key k, which represents the normal functionality of the unlocked chip. The chip will be operated in this mode while it is in the field. In mode M1, the SC continues to hold its previous state. This mode provides test and debug capability without letting the key to be exposed as FFk continues to hold its previous state. Thus, no key information is captured in M1. The rest of the circuit becomes functional mode when SE = 0 and scan mode (shift in or shift out) when SE = 1. Finally, SC becomes the scan cell at mode M2 and FFk becomes a part of a scan chain.

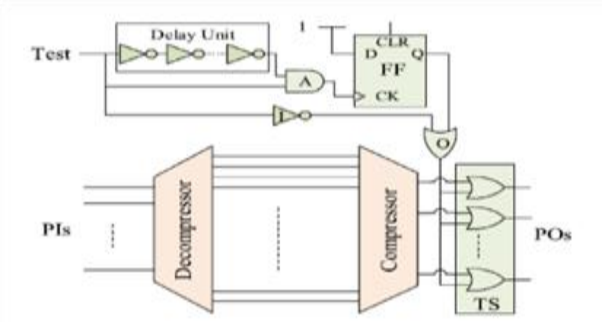

Fig. 3. Proposed architecture to restrict scan data access

- *Manufacturing Test:* The implementation of manufacturing tests using our proposed SC does not require any additional modifications in the existing test infrastructure. Note that the key is not programmed at this stage. It is required to keep Test

pin active high (logic 1) during the test. During the scan shift-in phase, the SCs become a part of a scan chain({Test, SE} = {1, 1} =M2) and receive values generated by the ATPG tool. Note that the key gate (k) is directly reachable from the SI. During the test response capture, the rest of the circuit becomes functional while the SCs hold their current state ({Test, SE} = {1, 0} =M1). No key bits are captured in the SCs as they continue to hold the states received during scan shift-in phase. This helps us to eliminate all the attacks completely. Finally, the captured functional responses are shifted out through the scan chain ({Test, SE} = {1, 1} =M2).

- *Post silicon Debug and Validation:* Complex modern designs can suffer from subtle logic and electrical design bugs that escape design verification and are only discovered in first silicon. This necessitates support for post silicon validation, and if a bug is discovered, its diagnosis followed by design changes to correct the problem. Post silicon debug is extremely challenging, and at a minimum requires both a fully functional test (on the activated design) as well as extensive scan test support. This extent of intrusive testing of the fully functional circuit can make it vulnerable to key discovery through SAT-based attacks or other formal tools. We therefore allow such full testing only in a secure design environment, with the key again applied through the scan chain, even if it is already programmed. Full scan tests on the fully functional circuit are performed in mode M1. Recall that in this mode, with the scan enable low (functional mode), the programmed key bits are not captured in the SCs from where they are presented to the logic; instead, the SCs are designed to hold and retain their current value. Thus, if the key bits are shifted into the SCs during the scan shift in M2, and the scan enable (SE) is then lowered to the functional mode (M1), the scanned in key bits will be retained in the SCs ensuring unlocked functional operation as long as the scan enable stays low. Single or multicycle tests can be performed and the results shifted out (M2).
- *Functional Tests:* The functional test can only be performed after the activation of the chips. Mode M0 supports functional tests. Functional patterns are applied to the PIs of a chip, and the responses are collected at the POs. It is required to initialize the finite state machine of a design before actual tests are performed, and sometimes could lead to millions of clock cycles. Test engineers often shift this initialization state through existing scan architecture. Mode M2 can be used to shift this state to the design, and then, it is switched to mode M0.
- *Mode Control:* An important restriction on switching between different operation modes for the SC is absolutely necessary for maintaining security. Switching from M0 to M2 (M0 →M2 or M0 →M1(Test = 1)→M2) cannot be permissible. To be specific, any positive transition at the Test pin will not be permitted. The key will be captured in M0 and be shifted out while the cell is in M2, if we allow this to happen. In addition, we will not allow shift out when Test is not asserted (i.e., Test = 0), which will prevent an adversary getting scan data (from SC to end of the scan chain may be shifted out while setting SE = 1) during Test = 0. our proposed architecture to restrict scan data access. We have added a series of OR gates at the output of the compressor (Test data compression is widely accepted by the industry), which is highlighted in green. The output of the test suppressor (TS) block becomes always 1 when the output of the OR gate (denoted by O) is asserted. One can place TS block before the compressor; however, the number of OR gates will be increased significantly. The output of the OR gate O becomes 1 when one or both inputs become 1. This ensures that an adversary cannot access scan data while Test = 0, which is one of the requirements for protecting the key. Now, we need to make sure that there is no positive transition on the Test pin.
- A pulse generator, when it experiences a positive transition of the Test pin. The delay unit consists of odd number of inverters and is fed to an AND gate A. A pulse with duration t is generated at the output of gate A. The width of this pulse t can be controlled by manipulating the number of inverters. The output of the AND gate A is fed to the clock input of the FF shown in Fig. When the FF detects a pulse, logic 1 will be captured and its output becomes 1 permanently. This FF can be cleared once during power up or after certain clock cycles (length of the scan chain) depending on one's choice. It is worth mentioning here that the Test pin can also be fed to the clock input of FF.
- *Secure Cell Placement:* Higher fault coverage (FC) is often required to ensure the high yield of good chips. In a circuit, there are many untestable faults due to the controllability and/or observability issues. Test point insertion is widely used to detect many of these untestable faults and thus increase the FC of a circuit. Our proposed SC can be used as a test point. Thus, the objective of placing a SC (e.g., one key bit) in the net list in such a way that it provides the detection capability of untestable faults.

Algorithm 1 determines the key location such that FC can be increased by sorting the nets (fault locations) based on the number of faults present in them. The two scenarios may arise. First, the number of such nets (L) is greater than key size, |K|. This may arise when the design has many unstable faults and

we have enough nets to place the key gates. Second, the number of such nets, L is less than key size, |K| (lines 11–18). L key gates are placed first (lines 11– 14), and the remaining key gates are placed randomly (lines 15–18). Note that the SC introduces few new faults to the design that can reduce the overall FC.

4. Proposed flow for enabling trust in IC manufacturing and test

The primary requirement for preventing IC overproduction and IP piracy is to obfuscate a design with a key which sign and fabrication.

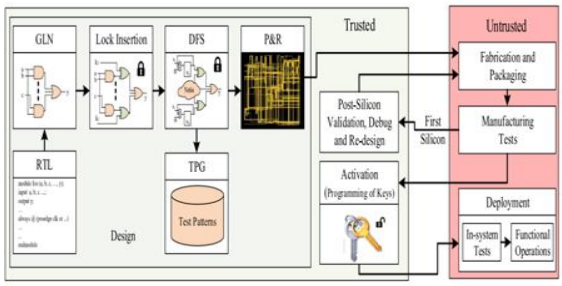

Fig. 4. Proposed flow for enabling trust in IC manufacturing and test

For the obfuscation key. The manufacturing tests can be performed at the foundry and/or package assembly as these tests do not require any key. It is also important to implement manufacturing tests before the activation of chips as an untrusted foundry can manipulate the yield information (the ratio of the defect free chips to the total number of chips) with the SoC designer and stockpile a large number of chips without contributing any design costs. In addition, it can source defective or out-of-specification chips to the market if the chips are activated before the manufacturing tests. Thus, it is necessary to activate the chips in a trusted environment such that an untrusted entity cannot get any undue advantages. In summary, manufacturing tests can be performed at any untrusted site; however, activation must be executed at a trusted site. It is also important to note that post silicon validation and debug require the chip to be fully functional (activated) with full structural test capability enabled.

Fig. 4 shows the overview of our proposed flow for enabling trust in IC manufacturing and test. This flow is exactly the same as existing IC design process, except for the lock insertion, DFS, and activation stages. The process starts with the RTL design phase, and then, it goes through synthesis to obtain gate level netlist (GLN). A set of key gates are now inserted to lock the GLN, which can only be unlocked through a proper key. We recommend adopting one existing RE-resistant lock insertion technique such that an adversary cannot find the key by simply observing the key gates. Note that the left half of highlighted in green (design phase, activation, and post silicon validation and debug) is under designer's control and is trusted. On the other hand, the right half of highlighted in red (fabrication and packaging, manufacturing tests, and deployment) is untrusted.

Even if the keys are resistant to RE, an adversary can still discover the key by using brute force, greedy, and SAT-based attacks. To avoid the key being exposed to these attacks, we propose to insert novel secure scan cells (see Section IV-B for details) to the key gates. This makes the keys resistant to known manufacturing test-related attacks. As all the key gates are reachable through these SCs, it is not required to provide key information to the ATPG tool for generating test patterns. It is worth mentioning that the keys now satisfy all the requirements mentioned. After DFS stage, the design is moved to the place and route (P&R) stage, and then Graphic Database System II files are created. Finally, they are sent to foundry for fabrication and packaging.

After manufacturing the first batch of chips, the foundry performs manufacturing tests and sends to the SoC designer for post silicon validation and debug, where it validates correct behavior in actual application environments. Any bugs may have been undetected previously during presilicon verification. During this stage, the SoC designer performs many different tests (combination of structural and functional) and observes the internal states to detect and diagnose any bugs. Our proposed DFS architecture provides the postsilicon validation and debug support which is absolutely required for SoC design and fabrication process.

Once the post silicon validation and debug is complete, the SoC designer provides the contract to a foundry to fabricate a certain number of chips. After fabrication, the foundry performs manufacturing tests and sends the defect-free dies to the assembly for packaging. The assembly performs final tests and sends back the chips to the SoC designer for activation. Finally, SoC designer activates the chips and sends them to the market for deployment. In-system functional tests can be performed on these activated chips in the field to test for their correct functionality.

Note that the left half of Figure highlighted in green (design phase, activation and post-silicon validation and debug) is under designer's control and is trusted. On the other hand, the right half of Figure highlighted in red (fabrication and packaging, manufacturing tests, and deployment) is untrusted and the keys may get compromised due to various attacks (SAT, RE, etc.). As the key information is not leaked for our proposed design during any tests, we can safely say that these attacks are ineffective in extracting the key.

5. Results and analysis

A. *Security analysis*

Ensuring security by protecting the key being exposed to an adversary is our prime objective. In this section, we will present different known attacks for security evaluation.

- *Attack Resistance:* All different attacks (e.g., brute force, greedy, and SAT-based attacks) are primarily based on the actual observation of the response of a logic cone through the scan chains of a circuit. As long as the key information is captured during functional

mode and then dumping the responses through scan chains, the key will be exposed to the aforementioned attacks. Our proposed design is resistant to these attacks as the SC prevents an adversary to capture key information while testing. SC is designed in such a way that it holds its previous state when the chip experiences tests. In addition, we impose restrictions for mode switching (mode M0 to mode M2) to access scan data. An adversary cannot extract functional response through the scan chains. He/She can only observe all 1s, when he/she tries to dump the scan data which contain the key information. Now one may argue that an adversary can perform SAT-based attacks by observing the functional responses. The inability of the attacker in our approach to use the blocked scan chains in a legally acquired unlocked chip limits the attacker's ability to obtain the complete set of internal FF states from an available functional part. Only the small number of input/output signals, along with perhaps a few additional signals captured in accessible internal memory, is available as input data for the SAT solver in its attempt to evaluate the obfuscation key. The logic states of the much large number of internal FFs (in each execution state) are no longer available as traces and must be computed by the SAT solver through extensive analysis over many sequential time frames. Consequently, the number of unassigned variables explodes dramatically for the SAT solver. Informally, this increase in computational complexity can loosely be compared to the increase in complexity of sequential ATPG over that of single time frame scan-based combinational ATPG. Practical state-of-the-art SAT solvers today are unable to handle sequential analysis for large designs beyond a few dozen time frames due to the explosion in problem size. Thus, our approach clearly makes mounting a SAT attack extremely difficult, if not impossible. Earlier proposals that allow scan tests in the unlocked mode can be attacked by repeatedly analyzing a single time frame using SAT for multiple test inputs.

- The security of our proposed approach lies on the length of the key. A key must be long enough such that it can withstand exhaustive key search, as our proposed design is resistant to brute force, greedy, and SAT-based attacks, and maintains NP
- completeness. As no key information is captured during the test, an attacker must try at least 2|K| combinations to make the circuit completely functional. Here, |K| is the length of the key. It is computationally unfeasible to find a correct key when |K| is greater than 128 considering current computing resources. However, one can use 256 or higher bit keys for obfuscating a netlist considering future computing resources.
- *Tampering:* An untrusted foundry can modify the masks to bypass the mode control logic (see Fig. 5 and write a permanent "zero" value at the output of the OR gate, O. In this case, an adversary has the full control of changing the modes (M0 to M2) and perform SAT-based attacks to find the key. Fortunately, this attack can easily be detected by the SoC designers and can be prevented. If the foundry manufacture chips with the tampered masks and send chips to the SoC designer for activation, he can easily detect the tampering by switching the modes and observe data. The scan data will be all 1s if it not tampered. Now an untrusted foundry can maintain two (one tampered and one genuine) sets of masks, and send chips to the SoC designer those are manufactured with genuine masks. For the worst case, it can send one tampered chip (fabricated with the tampered masks) along all genuine chips hoping that the SoC designer will burn the key and thus can get hold of the scan data (key) from this working chip (bypassed our security measures). To circumvent this attack, the SoC designer needs to verify the chip before activating. It is important to note that the reputation of a foundry will be demolished if the SoC designer detects tampering. Moreover, it is extremely expensive to design a new set of masks, we believe that there is little economic incentive for an untrusted foundry to maintain two different sets of masks.

B. *Area Overhead Analysis*

The area overhead for our proposed approach is primarily resulted from four parts.

- *SC module:* This SC contents two parts: a 4-to-1 MUX and a scan FF. The SC can switch among three modes: functional mode, hold mode, and scan mode. Based on our proposed structure, this will not disclose the key during any time. For a single SC (a 4-to-1 MUX and a scan FF), it usually contents 20 gates. The number of SCs are equal to the key length |K|, as each key bit is fed to a different SC. We require 256 (128) SCs when |K | is 256 (128) to maintain long-term security. The approximate gate count for an SC is around 20.
- *Keys gates:* The size due to keys also depends on the length of chip unlock key. To implement one key bit, we need one XOR/XNOR gate.
- *Test suppression:* The number of OR gates is equal to the compressor output. We can safely assume that we require 100 OR gates for Test Suppression.
- *Mode control:* We need approximately 20 gates to implement this module.

From the above analysis, we conclude that the majority of the overhead results from the SCs (number of key bits). The total gate count for our proposed approach is approximately 5200 when we consider 256-bit key. This can be reduced

significantly to 2700 when the key is 128-bit long. For a large benchmark (e.g., b19), the area overhead is less than 1%. However, it can be very less (1%) for a modern industrial design with millions of gates. Note that we need one additional pin (Test) to provide DFS support.

C. Simulation Results

To evaluate the effectiveness of our proposed DFS architecture, we use Synopsys tools to perform the simulation by using Synopsys 32-nm SAED32 EDK Generic Library on IWLS 2005 benchmark circuits.

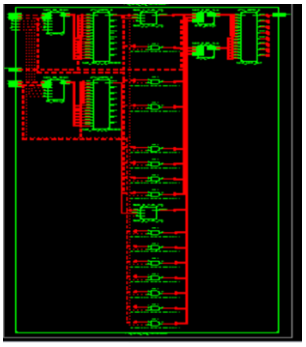

Fig. 5. RTL Schematic diagram for secure cell for IC design

The test metrics comparison between different methods. We compare the test coverage and pattern counts among the original netlist with no locks (denoted as ORG), key gate-inserted netlist (KEY) and our proposed DFS-inserted netlist (DFS). Column 2 shows the size of the obfuscation key, which represents the number of key gates to be insert into the netlist. Columns 3–5 represent the test coverage. The test coverage for KEY is computed by applying a preset key (all 0s) during test pattern generation; on the other hand, we do not need any key information for DFS. Column 6 represents the percentage change of the test coverage from KEY to DFS. We do not expect any significant change in the test coverage. However, we lose a small percentage for some benchmarks. This can be due to the added faults from the MUX of the DFS architecture. Similar analysis can be performed for the test pattern counts. We see a minor increase in the pattern counts for moderate and large size benchmarks.

6. Conclusion

In this paper, we have proposed a novel SC design for implementing DFS infrastructure that successfully prevents the leaking of obfuscation key to an adversary, and thus establishes trust in semiconductor manufacturing. First, our proposed SC disables scan dump after functional mode. This provides a complete protection against all different attacks that require an oracle (an unlocked functional chip) to compare the simulation responses.

References

[1] U. Guin, Z. Zhou, and A. Singh, "A novel design-for-security (DFS) architecture to prevent unauthorized IC overproduction," in *Proc. IEEEVLSI Test Symp. (VTS)*, Apr. 2017, pp. 1–6.
[2] U. Guin, K. Huang, D. DiMase, J. M. Carulli, M. Tehranipoor, and Y. Makris, "Counterfeit integrated circuits: A rising threat in the global semiconductor supply chain," *Proc. IEEE*, vol. 102, no. 8, pp. 1207–1228, Aug. 2014.
[3] U. Guin, D. DiMase, and M. Tehranipoor, "Counterfeit integrated circuits: Detection, avoidance, and the challenges ahead," *J. Electron. Test.*, vol. 30, no. 1, pp. 9–23, 2014.
[4] R. Chakraborty and S. Bhunia, "Hardware protection and authentica-tion through netlist level obfuscation," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, Nov. 2008, pp. 674–677.
[5] Y. Alkabani, F. Koushanfar, and M. Potkonjak, "Remote activation of ICs for piracy prevention and digital right management," in *Proc. IEEE/ACMInt. Conf. Comput.-Aided Design*, Nov. 2007, pp. 674–677.
[6] J. Huang and J. Lach, "IC activation and user authentication for security-sensitive systems," in *Proc. IEEE Int. Workshop Hardw.-Oriented Secur. Trust*, Jun. 2008, pp. 76–80.
[7] A. Baumgarten, A. Tyagi, and J. Zambreno, "Preventing IC piracy using reconfigurable logic barriers," *IEEE Des. Test Comput.*, vol. 27, no. 1, pp. 66–75, Feb. 2010.
[8] M. Tehranipoor and C. Wang, *Introduction to Hardware Security andTrust*. New York, NY, USA: Springer-Verlag, 2012.
[9] U. Guin, Q. Shi, D. Forte, and M. M. Tehranipoor, "FORTIS: A com-prehensive solution for establishing forward trust for protecting IPs and ICs," in *Proc. ACM Trans. Design Autom. Electron. Syst. (TODAES)*, 2016, p. 63.
[10] Y. Alkabani, F. Koushanfar, and M. Potkonjak, "Remote activation of ICs for piracy prevention and digital right management," in *Proc. IEEE/ACMInt. Conf. Comput.-Aided Design*, Nov. 2007, pp. 674–677.
[11] J. Huang and J. Lach, "IC activation and user authentication for security-sensitive systems," in *Proc. IEEE Int. Workshop Hardw.-Oriented Secur.Trust*, Jun. 2008, pp. 76–80.
[12] A. Baumgarten, A. Tyagi, and J. Zambreno, "Preventing IC piracy using reconfigurable logic barriers," *IEEE Des. Test Comput.*, vol. 27, no. 1, pp. 66–75, Feb. 2010.
[13] M. Tehranipoor and C. Wang, *Introduction to Hardware Security andTrust*. New York, NY, USA: Springer-Verlag, 2012.
[14] U. Guin, Q. Shi, D. Forte, and M. M. Tehranipoor, "FORTIS: A com-prehensive solution for establishing forward trust for protecting IPs and ICs," in *Proc. ACM Trans. Design Autom. Electron. Syst. (TODAES)*, 2016, p. 63.
[15] G. K. Contreras, M. T. Rahman, and M. Tehranipoor, "Secure split-test for preventing ic piracy by untrusted foundry and assembly," in *Proc.Int. Symp. Fault Defect Tolerance VLSI Syst.*, Oct. 2013, pp. 196–203.