2013 14th International Workshop on Microprocessor Test and Verification

Secure and Trusted SoC: Challenges and Emerging Solutions

Abhishek Basak[1], Sanchita Mal-Sarkar[2], Swarup Bhunia[1]
[1]Case Western Reserve University, Cleveland, OH, USA
[2]Cleveland State University, Cleveland, OH, USA
Email: skb21@case.edu

Abstract– Over the ages, hardware components, platforms and supply chains have been considered secure and trustworthy. However, recent discoveries and reports on security vulnerabilities and attacks in microchips and circuits violate this hardware root of trust. System-on-Chip (SoC) design based on reusable hardware intellectual property (IP) is now a pervasive design practice in the industry due to the dramatic reduction in design/verification cost and time. This growing reliance on reusable pre-verified hardware IPs and design automation tools that are often acquired from untrusted third party vendors severely affects the security and trustworthiness of SoCs, particularly when coupled with fabrication in untrusted offshore foundries. This paper presents an overview of the various security challenges in the SoC design cycle and possible solutions for protection.

1. INTRODUCTION

Security of electronic hardware at different stages of its life cycle has emerged as a paramount concern to integrated circuit (ICs) designers, system integrators, and end users. Over the past decade, with dramatic reduction in design/verification time/costs due to reusable pre-verified intellectual property (IP), System-on-Chips (SoCs) have become the prime driver of the embedded electronic domain. These IPs include soft IPs (RTL level), firm IPs ready to be used in hardware (synthesized gate-level netlists) and even hard IPs (GDS-II files after layout). Major security issues at different stages of SoC life cycle include piracy during IP evaluation and use, reverse engineering, cloning, counterfeiting, as well as malicious hardware modifications i.e., hardware Trojan attacks. Furthermore, use of untrusted foundries in a fabless business model greatly aggravates the SoC security threats. Due to ever-growing computing demands, modern SoCs tend to include many heterogeneous processing cores (e.g., MPSoC), scalable communication network, together with reconfigurable cores, e.g., embedded FPGA, in order to incorporate logic that is likely to change as standards and requirements evolve. Such design practices greatly increase the number of untrusted components in a SoC design and make the overall system security a pressing concern. Protection against these attacks is extremely challenging due to economic issues, the integration of the 3rd party IPs, emerging attack modes, as well as the sheer complexity of modern SoCs. To provide higher levels of assurance and trust to designers, system manufacturers and end users, there is an urgent need to integrate low-cost robust security measures during design and manufacturing test covering all aspects of security threats. This presentation will focus on diverse security issues in reusable IP-based SoC design and possible countermeasures to achieve secure trustworthy SoCs.

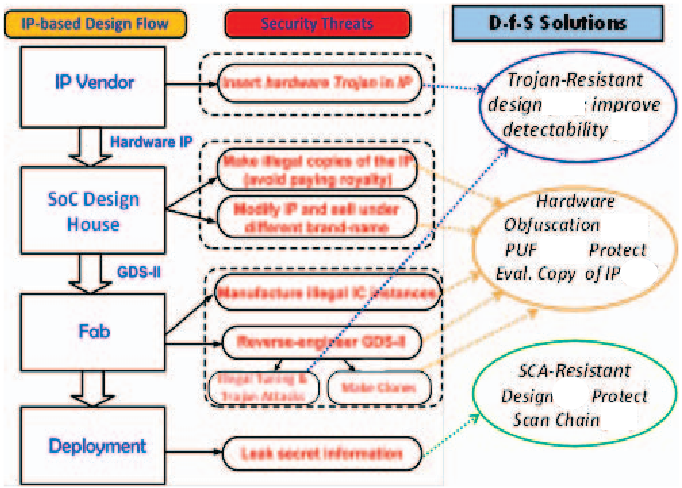


Figure 1: Different security threats spanning SoC life cycle and corresponding Design-for-Security (DfS) measures to protect against these threats.

2. THREATS AND CHALLENGES

Security is becoming an increasingly important parameter in current SoC design due to diverse threats and attacks at different stages of the design cycle as illustrated in Fig. 1. These threats include piracy, counterfeiting, reverse-engineering of the design coupled with malicious design modifications or hardware Trojan attacks and leakage of secret information. These security issues affect different parties including chip manufacturers, system designers and end users. For effective protection against these attacks, design and test time considerations, i.e. incorporation of ***Design-for-Security (DfS)*** features in the SoC cycle are becoming essential. However, DfS measures for different security threats require specific design modifications in individual IPs which significantly increases design effort, overhead and complexity leading to increased time-to-market. IP blocks used in a SoC design are often obtained from untrusted 3rd party vendors. In addition, the general heterogeneous architecture of current SoCs makes many IP-level DfS mechanisms unusable at SoC level. Finally, security primitives in SoC often compromise post-silicon debug and failure analysis by restricting signal observability. To effectively address the wide array of SoC security threats, many efficient D-f-S solutions have emerged, some examples of which are shown in Fig. 1. These include a secure design flow through hardware obfuscation, design techniques for effective detection of malicious modifications, Physical Unclonable Functions (PUF) for design authentication and other tamper-resistant designs. Next we review some of these solutions that target two security aspects:

1550-4093 2013

DOI 10.1109/MTV.2013.24

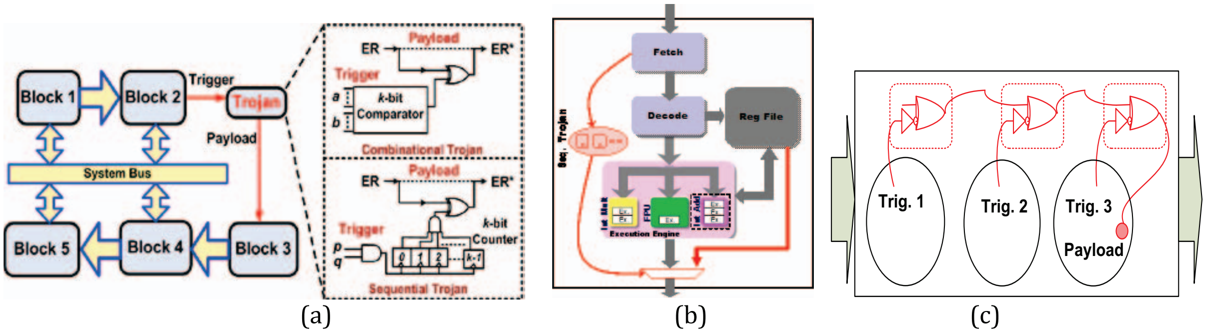


Figure 2: Hardware Trojan attacks of different forms – a) combinational and sequential Trojans causing malfunction; b) a Trojan with capability of leaking secret information from inside a crypto chip [3]; and c) a Trojan with distributed trigger condition [1].

i) Security against malicious modification (Hardware Trojans)
ii) Security against piracy and counterfeiting of designs

3. Hardware Trojan Attack

The emerging trend of outsourcing the design and fabrication services to external facilities as well as increasing reliance on third-party IP cores and electronic design automation (EDA) tools makes ICs increasingly vulnerable to hardware Trojan attacks at different stages of its life-cycle, where untrusted components/personnel are involved [1]. This raises a new set of challenges for trust validation with respect to malicious design modification. In particular, it brings in the requirement for reliable detection of hardware Trojan inserted in an untrusted fabrication facility during post-manufacturing test. It also imposes a requirement for trust validation in IP cores obtained from untrusted 3rd party vendors [2]. The Trojan insertion model and the different structural/functional forms of Trojans are illustrated in Fig. 2.

3.1 Trojan Detection Methodologies

Unfortunately, conventional test and validation approaches cannot be used to reliably detect these malicious changes or Hardware Trojans. Conventional test strategies focus on identifying undesired functional behavior due to manufacturing defects in ICs. They do not target detection of additional functionality in a design due to malicious modification. It is likely that an intelligent adversary would insert hardware Trojans, which are stealthy in nature. A cleverly-designed Trojan is a relatively unobtrusive circuit, which triggers a malfunction only under rare conditions in order to evade detection. It is also possible to design Trojans with no impact on functional outputs, i.e. one that leaks secret information through current signature [3, 4]. It is extremely challenging to exhaustively generate test vectors for triggering a Trojan and observing its effect for the inordinately vast spectrum of possible Trojan instances (with varying form and size) an adversary can employ [4]. This is especially true for sequential Trojans or "time-bombs", which are activated only on occurrence of a sequence of rare events as shown in Fig. 2(a). On the other hand, it is sometimes possible to detect Trojan circuits by observing their effect on a "side-channel parameter" such as power trace or path delay [5, 6]. An important advantage of such side-channel analysis is that it avoids the requirement of activating a Trojan and observing its effect in output logic values. Hence, test generation for side-channel analysis is expected to be less challenging. However, effectiveness of side-channel analysis is limited by the large process-variation effect in nanoscale technologies and measurement noise which reduce the detection sensitivity for ultra-small Trojans in large multi-million gate designs.

Since existing manufacturing test solutions, either logic testing or side-channel analysis based, may not provide comprehensive Trojan coverage, run-time monitoring may be employed to improve the level of assurance. Run-time approaches for

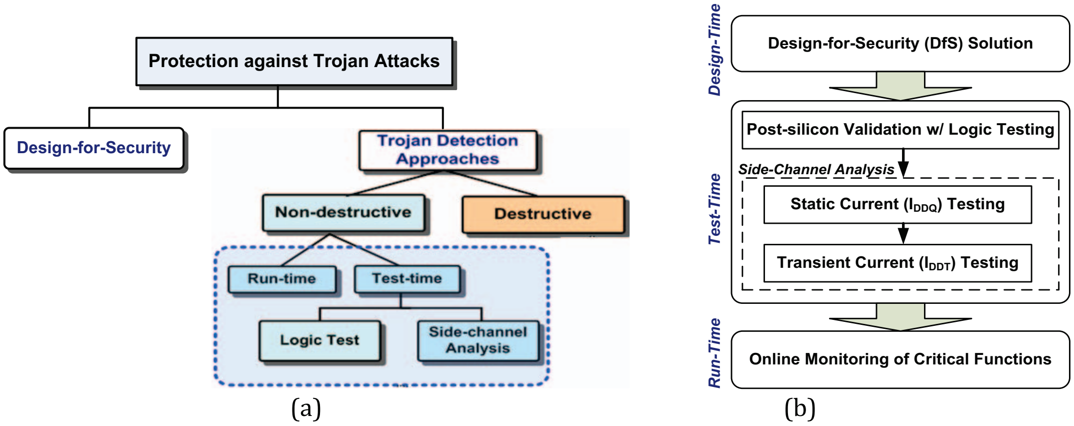
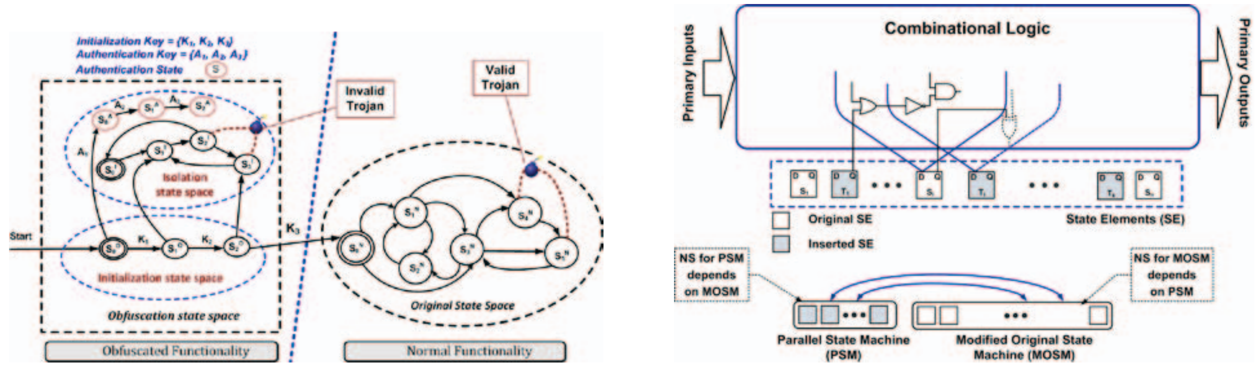

Protection against Trojan Attacks
Design-for-Security
Trojan Detection Approaches
Non-destructive
Destructive
Run-time
Test-time
Logic Test
Side-channel Analysis
(a)
Design-Time
Design-for-Security (DfS) Solution
Post-silicon Validation w/ Logic Testing
Side-Channel Analysis
Static Current (I_DDQ) Testing
Transient Current (I_DDT) Testing
Test-Time
Run-Time
Online Monitoring of Critical Functions
(b)

Figure 3: (a) Taxonomy of design and test techniques for protection against Trojan attacks; (b) an integrative protection approach that combines the benefits of design, test and online monitoring solutions.



(a) (b)

Figure 4: The obfuscation scheme for protection against hardware Trojans: (a) modified state transition graph and (b) modified circuit structure [3].

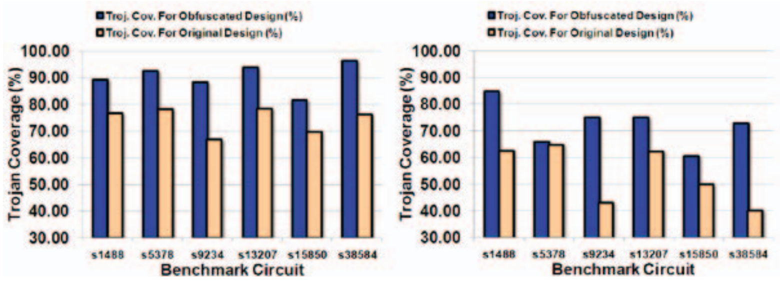
Trojan detection are based on monitoring execution of critical computations to identify specific malicious behaviors that trigger during long hours of in-field operation. For example, a Trojan, which leaks information from a crypto-chip via wireless channels, might consume large spikes in power during a relatively idle period when no computation/communications should be taking place. Hence, run-time monitoring of power traces can be used to detect such Trojan instances [4].

Trust verification in untrusted 3rd party IP cores involves additional challenges. Unlike ICs, it is difficult to obtain golden or reference models in case of IPs. For third-party IPs, one can only trust the functional specifications [2, 3]. Hence, one possible approach for trust verification in IP is to apply directed functional tests. However, such tests can only be generated if Trojan trigger conditions and Trojan effects are pre-characterized, which limits the effectiveness of functional validation for arbitrary Trojan circuits. An alternative approach is formal verification, such as sequential equivalence check with a high-level reference model, obtained from the functional specification of an IP or its high-level structural information, which do not scale well to large circuits.

3.2 Unified Security against Trojan Attacks

Note that existing Trojan detection approaches have their unique capabilities as well as limitations. There is yet no single “silver-bullet” technique available that can be applied to detect all classes of Trojans with high confidence. A possible solution to increase the level of confidence would be to combine various Trojan detection approaches with complementary capabilities. Logic testing approach can be combined with side-channel analysis [1, 4]. Similarly manufacturing test solutions can be combined with online monitoring. Finally, test approaches can be enhanced with design-for-security (DfS) solutions to achieve comprehensive protection against various malicious hardware security attacks.

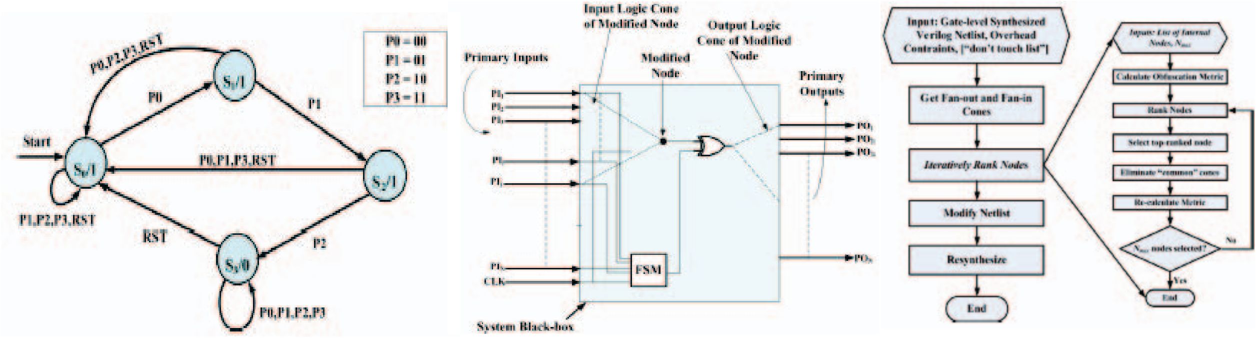
Existing research efforts for protection against Trojan attacks have focused on both design and validation techniques. Fig. 3(a) shows a broad classification of the protection techniques that apply at different stages of IC life-cycle. The design approaches make hard-to-detect Trojan insertion difficult or facilitate detection during post-silicon validation. Post-manufacturing Trojan detection approaches [1, 2] can be classified into two categories. Destructive testing by de-packaging, de-metallization and micro-photography based reverse-engineering of a chip is highly expensive and may not work if an attacker selectively tampers only a subset of the manufactured ICs [1]. Logic testing and side-channel analysis based validation provide complementary capabilities in detecting Trojans of different types and sizes. Hence, a post-manufacturing validation approach that combines their benefits can be effective in maximizing the Trojan coverage. For applications that require highest level of assurance against Trojan attack, one can combine post-manufacturing validation with online monitoring. Finally, validation approaches need to be complemented with low-cost design-for-security (DfS) solutions, which hardens a design with respect to Trojan



Benchmark circuit	Overhead (%)		Runtime (mins)
	Area	Power	
s1488	20.09	12.58	31
s5378	13.13	17.66	186
s9234	11.84	15.11	1814
s13207	8.10	10.87	1041
s15850	7.04	9.22	1214
s38584	6.93	2.63	2769

(a) (b) (c)

Figure 5: Improvement in Trojan coverage in obfuscated design as compared to original design with (a) 2-trigger nodes and (b) 4-trigger nodes; (c) low area and power overhead for obfuscation scheme in different benchmark circuits [3].



(a) (b)

Figure 6: (a) Example of net list level obfuscation in a sequential circuit. The key feature is to select a list of nodes to be modifed by an inserted FSM to produce wrong outputs until a particular sequence pattern (key) is applied at the primary inputs; (b) proposed obfuscated design flow and the iterative node ranking algorithm [2].

insertion and/or facilitates the test generation process during validation. Based on these observations, we propose an integrative protection approach, as illustrated in Fig. 3(b), which combines the benefits of multi-level protection and provides comprehensive coverage against Trojan attacks.

3.3 Obfuscation based Security against Trojan Attacks

Obfuscation is a technique that is conventionally employed to prevent piracy of both software and hardware IP. Our recent investigation has shown that a key-based structural and functional obfuscation can achieve protection against hardware Trojans triggered by rare internal circuit conditions [3]. The proposed obfuscation scheme is based on judicious modification of the state transition function, which creates two distinct functional modes: normal and obfuscated. A circuit transitions from the obfuscated to the normal mode only upon application of a specific rare input sequence, which defines the key, as illustrated in Fig. 4(a). We show that it provides security against Trojan attacks in two ways: 1) it makes some inserted Trojans benign, i.e. they become effective only in the obfuscated mode; and 2) it prevents an adversary from exploiting the true rare events in a circuit to insert hard-to-detect Trojans as shown in Fig. 4(b). The proposed design methodology can thus achieve protection from hardware Trojan attacks. Besides protecting ICs against Trojan attacks in foundry, we have observed that it can also protect against malicious modifications by untrusted computer-aided design (CAD) tools in both SoC and FPGA design flows. Simulation results for a set of benchmark circuits show that the scheme is capable of achieving high levels of security (Fig. 5 (a), Fig. 5(b)) against Trojan attacks at modest area, power and delay overhead (Fig. 5(c)).

4. PIRACY AND COUNTERFEITING

Evaluation of hardware IP cores is an important step in an IP-based SoC design flow. From the perspective of both IP vendors and IC designers, it is desirable that hardware IPs can be freely evaluated before purchase, similar to their software counterparts. However, protection of these IPs against piracy is a major concern for the IP vendors. Recent trends of IP piracy and counterfeiting can take several forms: (a) IPs being used without paying the requisite fees to the IP vendor; (b) design houses illegally selling IPs obtained from IP vendors to other parties; (c) fabrication houses manufacturing and selling illegal copies of a design without paying royalty fees to the design house, or (d) companies performing post–silicon reverse-engineering to derive the GDS–II database of an IC and manufacture illegal "clones". Whatever the form, IP piracy and counterfeiting affects IP vendors, chip designers and system manufacturers by depriving them of their revenue and market-share.

4.1 Existing Security Solutions

To prevent IP piracy, the IP vendors traditionally enforce a binding licensing agreement with the design house. Alternatively, they provide an IP in encrypted form. The decryption process is accomplished with a vendor specific design platform for simulation and synthesis [7]. The latter approach is prevalent in FPGA based design framework. On the other hand, some IP vendors allow simulation of the downloaded IP, but do not allow it to be synthesized to gate-level designs or bit-streams (for FPGA platforms) [8]. Unfortunately, such practices force a SoC designer to evaluate only the IP's functional behavior, but not the other important quality parameters. To overcome the shortcomings of existing IP evaluation practices, a recent industry initiative has resulted in a design platform that allows designers to download and use encrypted IPs from vendor websites. The synthesis and simulation tools in the design platform are capable of working in a user-transparent manner based on a technology undergoing IEEE standardization [9]. However, such a design technique mandates the use of a particular design platform throughout the design flow, which may not be acceptable for modern SoC designers, who typically rely on different software tools from diverse EDA vendors as well as in-house design tools.

On the other hand, several alternative approaches have been proposed to achieve protection of regular IPs (sales copy). A solution proposed, along the lines of obfuscation, is to modify the HDL (RTL) source code by removing the comments and changing the internal net–names following a simple string–substitution strategy. This is done in such a way that the functionality of the system remains unchanged, but the logic description becomes incomprehensible. However, these approaches do not modify the hardware structure and functionality and thus cannot provide hardware protection at all levels of design flow. Another protection scheme directed towards benefitting the IP vendor is to embed a *'Digital Watermark'* in the design, which helps to authenticate it at a later stage. Since this digital watermark (or signature) cannot be removed from the IP, it is easy to prove an illegal usage of such a component in litigation. However, these approaches do not obfuscate the design and hence, cannot prevent reverse engineering the design and unintended use of the IP.

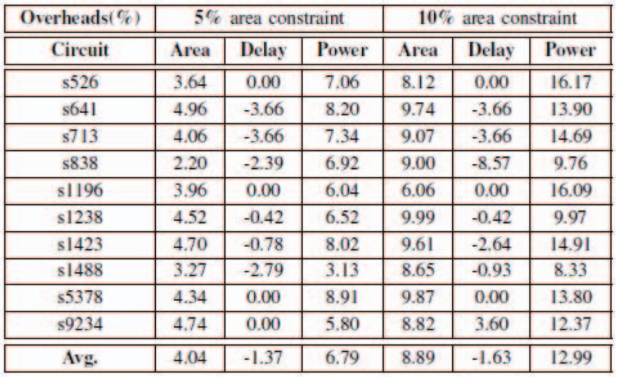
4.2 Obfuscation Based Solution

We have proposed a novel low-overhead, piracy-proof design flow for SoCs involving protection of the different 3rd party hardware IPs. In many scenarios, the IP vendors provide evaluation versions of their IPs to be downloaded and evaluated by IC designers. Although advantageous from both IP vendor and chip designer perspective, this practice makes the IPs extremely vulnerable to piracy. A low-cost solution for hardware IP protection during evaluation can be achieved by embedding a hardware Trojan inside an IP in the form of a finite state machine (FSM) with special structure [10]. The Trojan disrupts the normal functional behavior of the IP on occurrence of a sequence of rare events, thereby effectively putting an "expiry date" on the usage of the IP. The Trojan is structurally and functionally obfuscated after re-synthesis, thus protecting against potential reverse engineering efforts that target isolation of the Trojan.

Protection of regular sales copy of IPs can be achieved based on obfuscation of the design at the gate or higher level. This is by modification of a few selected nodes of the design and the state transition function, followed by a resynthesis of the flattened netlist. The objective of the modification process is to enforce undesired logic values at the primary outputs and state element inputs unless a sequence of patterns appears in the primary inputs [2]. The enabling pattern or key can vary from one instance of the IP to another. The gate level netlist obfuscation in a sequential circuit with 4 states in the inserted finite state machine (FSM) is shown in the schematic in Fig. 6(a). The FSM in the figure has some of the primary inputs as its inputs (besides clock and reset) and has one output signal. The FSM output is XORed with a few selected circuit nodes to invert the node values in the obfuscated state. The key effectively embeds authentication capabilities to the design, which helps to establish the legal source of the design in case of litigation, all the while ensuring that the degradation in the user experience is minimal. Obfuscation at the gate level is attractive since:

(i) the characteristic interpretability of the high-level behavioral description at the RTL level, makes any design modification much more observable as compared to gate level alteration; and (ii) the IP is usually transferred by the vendor at the gate level and the RTL description is not available.

The scheme provides high robustness to IP reverse-engineering at minimal hardware overhead. The nodes which are altered between the two function modes are selected judiciously in an iterative ranking scheme, shown in Fig. 6(b). The optimal set of nodes with larger fan–out and fan–in cones are preferred. This is because large fan–out and fan–in cones imply the modified node affects a large number of internal nodes and primary outputs. Fig. 6(b) portrays the obfuscated IP design flow. An optional don't-touch list of nodes (e.g. nodes on the critical path) can be provided at the start of the design flow, which are not to be modified. The IP vendor supplies the modified obfuscated IP to the design house, along with the activating sequence. The design house receives one or multiple IPs from IP vendors, and then integrates them on chip. To activate different IPs, the designer needs to include a low-overhead "controller module" in the SoC, that will steer different initialization sequences to the different IP blocks. The designer must also modify the test–benches accordingly to perform block–level or chip–level logic simulations. The manufacturing house manufactures the SoC from the design provided by the design house, and the test engineer performs post-manufacturing testing using the set of test vectors provided by the designer. The tested ICs are passed to the system designer along with different initialization sequences from the design house. In some scenarios, the IP vendor can design the IP with different activation sequences for designs supplied to different design houses. This will help to trace back to the source from where the IP was illegally leaked in case of litigation. The modest area, power and delay overhead for different benchmark circuits (number of modified nodes < 13% of all nodes) is shown in Table I.



Overheads(%)	5% area constraint			10% area constraint		
Circuit	Area	Delay	Power	Area	Delay	Power
s526	3.64	0.00	7.06	8.12	0.00	16.17
s641	4.96	-3.66	8.20	9.74	-3.66	13.90
s713	4.06	-3.66	7.34	9.07	-3.66	14.69
s838	2.20	-2.39	6.92	9.00	-8.57	9.76
s1196	3.96	0.00	6.04	6.06	0.00	16.09
s1238	4.52	-0.42	6.52	9.99	-0.42	9.97
s1423	4.70	-0.78	8.02	9.61	-2.64	14.91
s1488	3.27	-2.79	3.13	8.65	-0.93	8.33
s5378	4.34	0.00	8.91	9.87	0.00	13.80
s9234	4.74	0.00	5.80	8.82	3.60	12.37
Avg.	4.04	-1.37	6.79	8.89	-1.63	12.99

Table I: Design Overhead (%) with proposed obfuscation scheme for two different area constraints [2].

4.3 Physical Unclonable Function (PUF)

Physical Unclonable Functions (PUF) have emerged as an attractive security primitive for IC in a variety of applications like IP counter-plagiarism, IC authentication and cryptographic key generation in embedded system security. PUFs utilize a challenge response protocol that exploits the inherent random variations in a manufacturing process to generate unique signatures. Process variations cause fluctuations in device parameters, such as threshold voltage (Vth), channel length (L), leading to variations in circuit level parameters (e.g. path delay), which are typically used in a PUF for signature generation. PUFs have various security advantages- i) the challenge-response pairs are random and difficult to predict and copy for an attacker. In addition, challenge-response space can be large enough to render random trials ineffective. Many PUFs are designed such that any invasive tampering/reverse-engineering attempts would lead to alteration in the signature, leading to tamper-resistant hardware; ii) unlike digital keys stored in Non-Volatile Memories (NVM), PUFs are safer since the intrinsic signature is available only when the chips are running and a specific challenge vector is applied. NVM keys are vulnerable to invasive and probing attacks; iii) the cost of a PUF is expected to be lower than digital key storage accompanied by high tampering resistance environment.

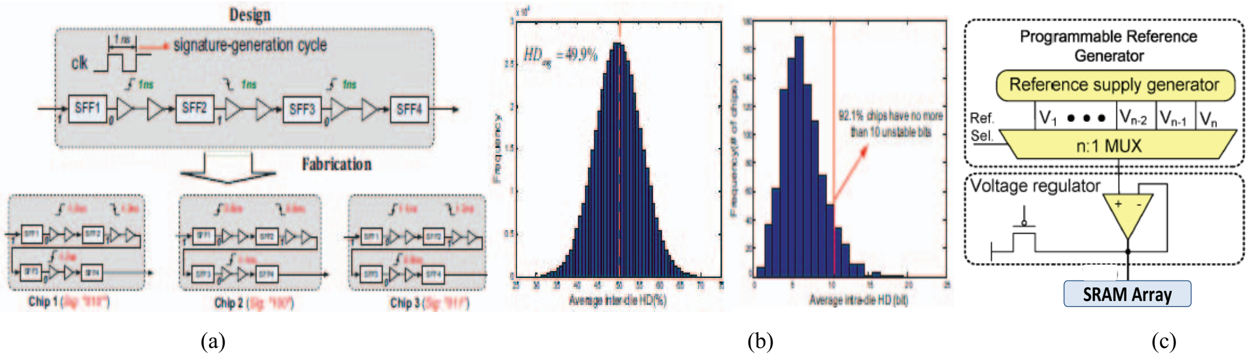


Figure 7: (a) PUF realization with on chip scan flipflops [11] (b) Normalized Inter and Intra Die Hamming distance of ~ 49.9% and ~6% for PUF signature in 1000 chips [11]; (c) Hardware architecture of PUF retrofitted into embedded SRAM [12].

PUFs have to be designed with minimal overhead, with respect to time and cost. A majority of existing PUFs require dedicated circuit structures. Apart from the substantial cost in silicon area, their integration into a system-on-chip (SOC) design needs extra effort on the placement, routing and verification, increasing design time. On the other hand, a separate class of relatively few PUF implementations generates a signature from existing on-chip structures, such as PUFs that exploit random mismatch in inner node voltages of memory elements (e.g. SRAM or Flip-Flops). This class of PUFs, however, often requires considerable modifications of the original design, leading to overhead in time-to-market and cost. Other disadvantages include comparatively smaller challenge-response space and Shannon entropy, leading to lower degree of signature uniqueness and higher predictability.

PUFs using already existing circuit structures in most chip designs, without major design modifications are attractive for security in SoC designs with minimal overhead. One such structure is the scan chain for testing and validation. [11] proposes to use this Design-for-Test structure to create large volume of signatures from a chip based on random delay variations in timing paths between two scan flip-flops (SFFs). Post manufacturing, the delays deviate (from chip to chip and within a chip) due to process variations. These deviations can be captured in the logic values latched in different SFFs by the signature-generation cycle controller. These values can then be shifted out from the scan chain to obtain a signature. The scan chain based PUF structure is shown in Fig. 7(a). The inter and intra die hamming distances for PUF signatures in 1000 chips is illustrated in Fig. 7(b), with a curve fitted Gaussian mean of ~ 49.9% and ~6% respectively. This signifies high uniqueness and reproducibility of the digital signature of 128 bits.

An alternative novel PUF structure utilizes the voltage scaling induced write failure in the already present embedded SRAM array to generate robust signatures, even for legacy ICs [12]. This exploits the fact that for a set of SRAM cells in an array, under scaled supply voltage, write access failure occurs only in specified cells depending on device- level process variations. Most modern ICs have isolated power grid for SRAM and hence the PUF structure can easily be implemented (Fig. 7(c)).

5. SUMMARY

With greater advances in the embedded and mobile electronics, the demand for increasingly complex SoC is ever-increasing with great constraints in cost and time-to-market. The life cycle of SoC typically involves multiple untrusted entities/tools, which make them extremely vulnerable to various security threats including piracy, counterfeiting, hardware Trojans etc. There is no single 'silver-bullet' solution for protection against these various security threats. Different methods, preferably with complementary capabilities, as reviewed in this paper need to be integrated into a design and validation approach for secure SoCs.

6. ACKNOWLEDGEMENTS

The work is supported in part by National Science Foundation grants # CNS-1054744 and # DUE-1245756.

REFERENCES

[1] Rajat Subhra Chakraborty, Francis Wolff, Somnath Paul, Christos Papachristou and Swarup Bhunia, "MERO: A Statistical Approach for Hardware Trojan Detection", CHES, 2009.

[2] Rajat Subhra Chakraborty and Swarup Bhunia, "Hardware Protection Through Netlist-Level Obfuscation", ICCAD, 2008.

[3] Rajat Subhra Chakraborty and Swarup Bhunia, "Security Against Hardware Trojan Attacks Using Key-based Design Obfuscation", JETTA, 2011.

[4] Seetharam Narasimhan, Xinmu Wang, Dongdong Du, Rajat Subhra Chakraborty, and Swarup Bhunia, "TeSR: A Robust Temporal Self-Referencing Approach for Hardware Trojan Detection", HOST, 2011.

[5] Seetharam Narasimhan, Dongdong Du, Rajat Subhra Chakraborty, Somnath Paul, Francis G. Wolff, Chris Papachristou, Kaushik Roy and Swarup Bhunia, "Multiple-Parameter Side-Channel Analysis: A Non-invasive Hardware Trojan Detection Approach", HOST, 2010.

[6] Seetharam Narasimhan, Dongdong Du, Rajat Subhra Chakraborty, Somnath Paul, Francis Wolff, Christos Papachristou, Kaushik Roy, and Swarup Bhunia, "Hardware Trojan Detection by Multiple-Parameter Side-Channel Analysis", IEEE TComp, 2012.

[7] T. Batra, "Methodology for protection and licensing of HDL IP". [Online], Available: http://www.us.design-reuse.com/news/.

[8] R. Goering, "Synplicity initiative eases IP evaluation for FPGAs", [Online], Available: http://www.scdsource.com/.

[9] "Recommended practice for encryption and [Use Rights] management of electronic design Intellectual Property (IP)". [Online]. Available: http://www.eda.org/twiki/bin/view.cgi/P1735/WebHome.

[10] Seetharam Narasimhan, Rajat Subhra Chakraborty and Swarup Bhunia, "Hardware IP Protection during Evaluation Using Embedded Sequential Trojan", IEEE D&T, 2011.

[11] Yu Zheng, Aswin Raghav Krishna, and Swarup Bhunia, "ScanPUF: Robust Ultralow Overhead PUF Using Scan Chain", ASP-DAC, 2013.

[12] Yu Zheng, MaryamSadat Hashemian, and Swarup Bhunia, "RESP: A Robust Physical Unclonable Function Retrofitted into Embedded SRAM Array", DAC 2013.