Hardware Vulnerabilities and Their Effects on CE Devices: Design for Security Against Trojans

By Anirban Sengupta

Hardware threats on consumer electronic (CE) devices are a serious concern for industry designers and researchers, and the impact of such threats (or tampering) is distressing from the commercial perspective of the CE industry. Thus, it is important to understand the impact of hardware-based vulnerabilities on a CE device from the perspective of functionality and performance.

BACKGROUND AND PURPOSE

CE devices such as mobile phones, tablets, digital cameras, printers, televisions, and web and surveillance cameras play major roles in our everyday life ranging from office work, entertainment, medical purposes, communications, and home activities. Since early 2010, most CE products have become digital in nature and have largely amalgamated with the computer industry, which is increasingly referred to as the consumerization of information technology.

One of the dominant features of CE products is the trend of ever-falling costs due to the introduction of automation (driven by electronic design-automation tools), cheaper labor cost, and the rapid evolution of semiconductor technology that allows the doubling of chip density (with the integration of transistors) every two years for a given cost, in accordance with Moore's law [1], [2]. Therefore, at the heart of a successful CE product design are computers in the form of a system-on-chip (SoC) or very-large-scale integrated (VLSI) circuits (see Figure 1).

Since early 2010, most CE products have become digital in nature and have largely amalgamated with the computer industry.

VLSI chips are mounted in nearly all modern CE devices, including mobile phones, palmtops, tablets, laptops, televisions, set-top boxes, and cameras. However, due to rapid strides made by semiconductor technology, a new VLSI chip based on a new process node or other advancement is released almost every month. Thus, the development cycle of a VLSI chip is shortened by involving third-party intellectual property (IP) cores from across the globe.

Furthermore, for economic reasons, most of these chips used in CE devices are manufactured in off-shore fabrication facilities. In addition, outsourcing design and test services and involving EDA software tools supplied by different vendors are quite common practices when designing and manufacturing the VLSI chips used in CE devices.

MOTIVATION: UNDERSTANDING HARDWARE VULNERABILITY IN CE DESIGN

LOCATIONS OF VULNERABILITIES

A typical design flow of a VLSI chip from a hardware-threat perspective is shown in Figure 2(a), and its corresponding life cycle is shown in Figure 2(b). The green boxes indicate trusted sources where the possibility of hardware vulnerability/threat/attack is nonexistent. The specification stage in a VLSI-chip flow is trustworthy, as there are no third-party sources, unspecified designers, engineers, or adversaries. However, the red boxes indicate the possibility of untrusted sources or processes involving third-party software/tools. Thus, the entire design process is vulnerable to hardware threats.

Although vendor watermarks or buyer fingerprints are embedded in the IP cores to resolve ownership conflict/ piracy (which subsequently is embedded in the chip design after integration), the chip-design/fabrication cycle remains open to vulnerabilities of hardware threats (watermarks and fingerprints only guard against piracy, counterfeit devices, and false ownership). The yellow box indicates a mix of trusted and untrusted processes, such as the fabrication step in the chip manufacture process.

Digital Object Identifier 10.1109/MCE.2017.2684940
Date of publication: 14 June 2017

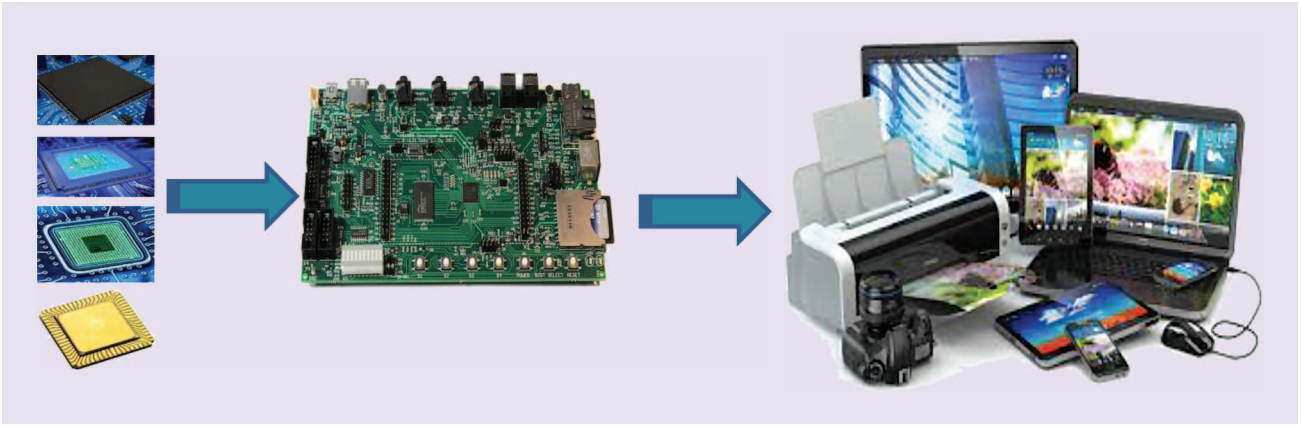


FIGURE 1. The VLSI chip at the heart of most CE products.

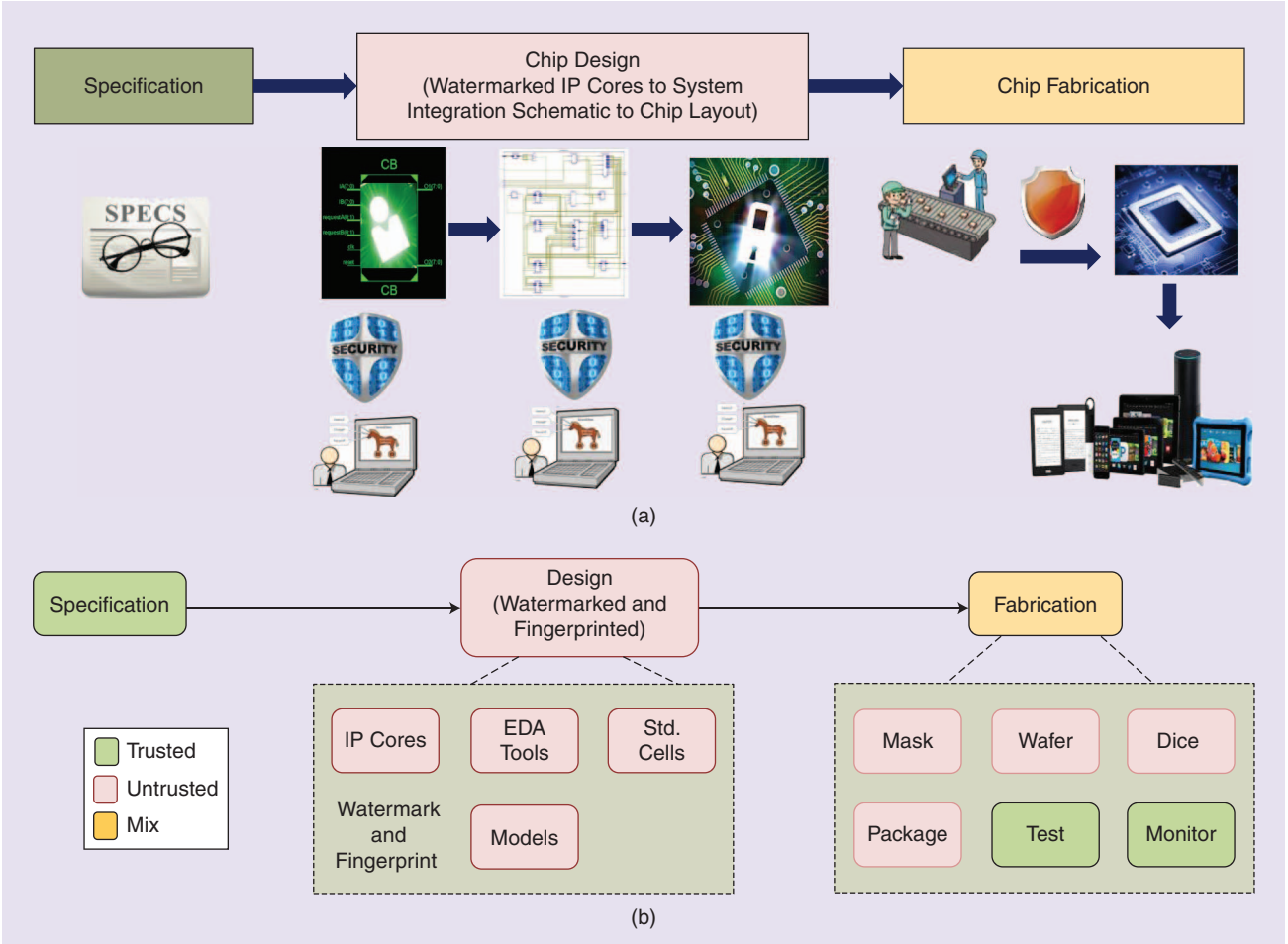


FIGURE 2. (a) A watermarked chip-design cycle indicating possible hardware vulnerabilities in each step. (b) The life cycle of a chip used in a CE product.

Each party associated with designing and manufacturing a chip can be considered a potential adversary (or attacker) who inserts malicious modifications (called *hardware Trojans*) or performs reverse engineering to gain access for tampering. Grave concerns about hardware vulnerability and the possibility of compromised CE devices when SoCs are used in such devices have been expressed by researchers in academia, industry, and government agencies.

PRE- AND POST-SILICON TESTING MAY NOT HELP IN DETECTING VULNERABILITIES

The major challenge is the difficulty in detecting such modifications through normal presilicon verification (such as a

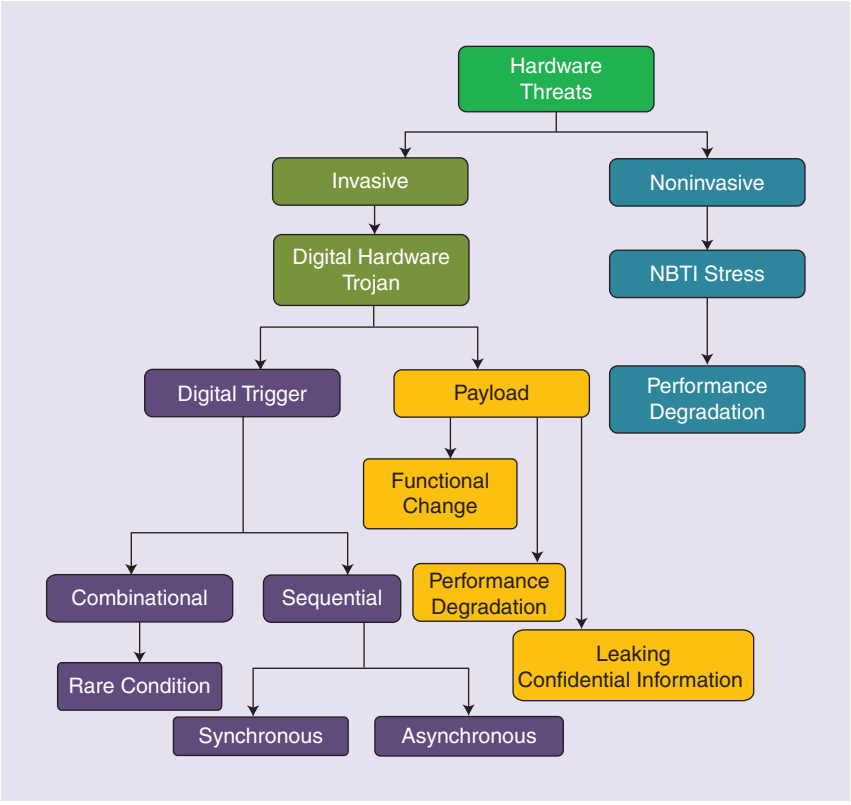


FIGURE 3. The classification of hardware threats for CE devices.

register transfer level simulation) and postsilicon testing. This is because, first, in most cases, presilicon verification requires a golden model for the entire chip, which is not available in the case of third-party IP-based designs. Second, there may be several IPs that are imported from third-party vendors as assumed preverified design blocks. Third, exhaustive verification may not be amenable for large multimodule designs. Finally, the third-party IP blocks operate correctly under normal circumstances but behave incorrectly under rare conditions.

Thus, normal validation tests such as simulations cannot detect malicious alterations. Furthermore, not all signals of an IP block may indicate the triggering of a Trojan (only a single signal may show different outputs during simulation, which could be bypassed if the validation process is not performed thoroughly). In contrast, postsilicon testing can be accomplished using destructive depackaging and reverse engineering of a chip. However, destructive depackaging is not scalable and highly expensive.

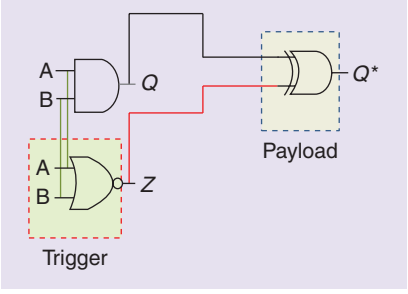


FIGURE 4. A combinational triggered-hardware Trojan under a rare event.

Additionally, postmanufacturing logic testing is also not efficient for detecting hardware Trojans due to their stealthy nature, numerous possible varieties, and activation under a rate condition or event. Finally, detection through side-channel parameters such as power, delay, and energy is also limited due to a large process variation effect in nanoscale chip technologies and measurement noise [3], [5].

CLASSIFICATION OF HARDWARE THREATS AND THEIR EFFECTS ON CE DESIGN

This article provides a taxonomy of the major hardware threats for CE devices (see Figure 3; inspired by [3] and [4]). Hardware threats can be categorized into two types: invasive and noninvasive. Invasive threats represent attacks that involve direct modification to the circuit and typically refer to digital-hardware Trojans. Yet, noninvasive threats represent attacks that are possible without altering the circuit.

One of the most effective noninvasive techniques to launch an attack on a chip is through the negative bias temperature instability (NBTI) stress method. NBTI stress results in the accelerated aging of nanoscale components, yielding to functional failure. A circuit vitiates over the lifetime of the chip and ultimately results in chip failure.

Circuit degradation is highly influenced by the operating conditions of the circuit. Rogue designers who are aware of the aging mechanism on circuit-performance degradation can exploit this to accelerate the aging process to decrease the lifetime, resulting in hardware-functional failure. Therefore, an attacker who is aware of the direct impact of NBTI stress on a hardware device can use this to his advantage to accelerate the failure mechanism.

However, digital-hardware Trojans can be classified into two types: trigger-activation based and payload based. Trigger-based Trojans can be triggered in combination or sequentially triggered. *Payload based* refers to hardware Trojans resulting in either functional changes or leaking confidential information or performance degradation (with no change in functionality).

THE TRIGGERED HARDWARE TROJAN: THE RARE CONDITION COMBINATIONAL TYPE

This type of hardware Trojan is activated under the combinational triggering of a rare-event input. Figure 4 shows an example of such a Trojan, where the occurrence of condition $A = B = 0$ at the trigger inputs (A and B) causes the circuit to have an incorrect output (Q*) instead of a correct output (Q). The red wire carries the output due to the trigger circuit in the payload, while the green wires carry the triggering inputs.

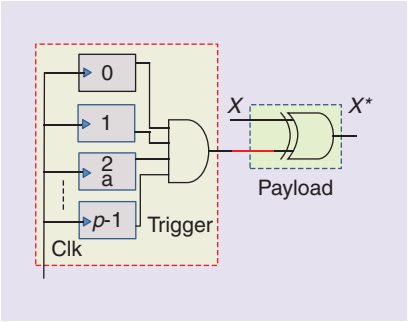


FIGURE 5. A synchronous counter sequential-triggered hardware Trojan.

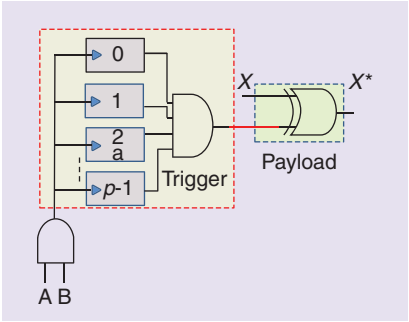


FIGURE 6. An asynchronous counter sequential-triggered hardware Trojan.

THE TRIGGERED HARDWARE TROJAN: THE SYNCHRONOUS AND ASYNCRONOUS SEQUENTIAL TYPES

The synchronous and asynchronous sequential types of hardware Trojans are triggered when the counter reaches a certain amount. For either type, once triggered, the output functionality of the circuit starts malfunctioning [3].

In Figure 5, for the synchronous sequential type, there are p counters working asynchronously under the influence of the clock signal. In Figure 6, for the asynchronous sequential type, the p counters are working asynchronously under the rising transitions of AND-gate output (fed with inputs A and B) instead of a common clock. For either type, the red wire carries the triggering output for affecting the payload block.

When the count reaches $2^p - 1$, the Trojan is triggered and the circuit produces an incorrect output (X^*) instead of a correct output (X). Such types of modifications done by an adversary are also called the *Time-Bomb Hardware Trojan*, as the triggering occurs after a certain predefined time event.

THE PAYLOAD-BASED HARDWARE TROJAN: RESULTING IN FUNCTIONAL CHANGE

This type of hardware Trojan covertly affects the payload of the circuit, resulting in functional changes such as an alteration in the produced computational value at the output. Figure 7 shows such an example, where a binary coded decimal (BCD) adder circuit black box comprises of three input signals, Enable (En), A (4 b), and B (4 b), and the output signal $O_3O_2O_1O_0$.

During a functional verification/simulation test, En is always kept on (by feeding logic 1 in the test bench) by the design verification team, as they would falsely assume it to be a circuit-enable signal (without being aware that it is designed in disguise to act as a trigger signal for the Trojan). Under En = 1, the BCD adder circuit would behave normally, thus passing the register transfer level simulation/functional verification test. This is because the Trojan would have only been activated under a rare event (e.g., resetting of the enable, i.e., En = 0, which the design integrator would normally never make during a verification test). Thus, after integration, whenever the IP En signal remains off, it continues to work maliciously (due to Trojan logic activation) and impacts the remaining system by providing the wrong output value [5].

THE PAYLOAD-BASED HARDWARE TROJAN: RESULTING IN PERFORMANCE DEGRADATION

The circuit itself is stealthily affected by this type of hardware Trojan through triggering, but this results in no functional change besides an occasional performance degradation. This means that after the Trojan block is activated, although the functional output always remains the same, a few occasions produce outputs after some delay, resulting in the aforementioned performance degradation. Thus, the Trojan goes undetected during functional verification.

Figure 8 provides an example of this type of hardware Trojan. The figure shows that, for inputs A = 0 and any input value of B (1 or 0), there will be a delayed output of Sum (S) because of the triggering in the adder block. This is because a black box design appears normal when it is purchased from a third-party vendor, but, internally, there are extra logics (i.e., the Trojan) appended with the 1-bit adder by some adversary.

Nevertheless, when A = 1, there is no delay because the Trojan block is not triggered (the delay that is caused by the inverter output would be ignored and considered as the input to the multiplexer). This type of Trojan may be secretly inserted by a third-party adversary into an adder block (or similar blocks) during product designing to cause occasional, unexpected timing violations (and

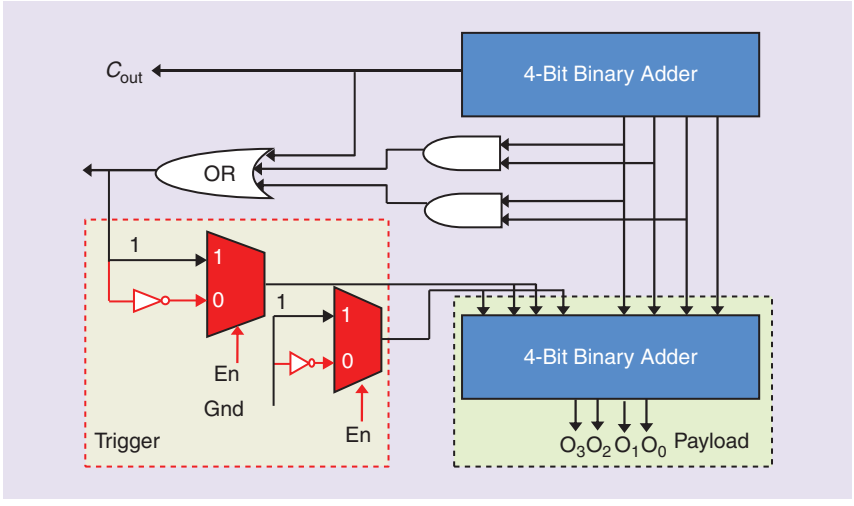


FIGURE 7. The Trojan logic inserted by an adversary in a BCD Adder IP (inputs A and B); Note: When Enable (En) = 0 (rare event), then Trojan blocks get activated and produces incorrect computational output ($O_3O_2O_1O_0$) [3].

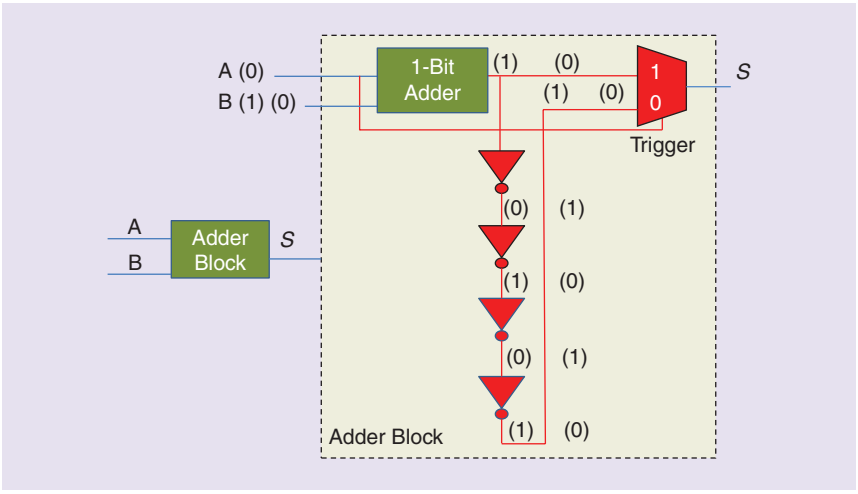


FIGURE 8. The hardware Trojan resulting in performance degradation.

subsequent time loss during the product-design cycle).

THE PAYLOAD-BASED HARDWARE TROJAN: RESULTING IN LEAKING CONFIDENTIAL INFORMATION AND DENIAL OF SERVICE

The Payload-Based Hardware Trojan yields the leakage of confidential information by bypassing encryption circuits. Figure 9 shows such scenarios, where the plaintext may skip the encryption block if the Trojan trigger is activated (adopted from [6]), and the red block indicates the Trojan block. Furthermore, Figure 10 shows an example of a hardware Trojan, which, on activation (En = 0), locks part of the circuit (in this case, it does not allow anything to pass through the muxes as the second input because both multiplexers are left uninitialized), which produces no output [denial of service (DoS)] at the output of the 4-bit binary adder.

HARDWARE ATTACK: NBTI STRESS (THE NONINVASIVE TYPE)

The attacker's objective is to accelerate the aging process of the CE device

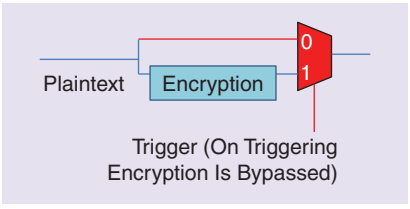


FIGURE 9. The hardware Trojan resulting into leakage of confidential information [4].

> The DFS flow is based on high-level synthesis framework driven through particle-swarm-optimization exploration.

when the product is under replacement warranty [4]. NBTI is one of the major hardware threats that results in performance degradation, leading to eventual device functional failure based on the input-vector pattern applied. These specific input patterns put stress on p-type metal-oxide-semiconductor (PMOS) transistors by increasing their threshold voltage (causing them to degrade with performance delay and fail). An attacker who is aware of the direct consequences of the NBTI stress may use it to an advantage to launch this type of hardware attack. The stress can expedite the aging-related performance degradation that NBTI causes when a PMOS device is on. It is noninvasive in nature, i.e., no modification to the circuit is made to launch the attack.

HOW IS AN ATTACK LAUNCHED?

In this threat, an attacker manages to obtain the netlist of the target hardware design through the help of a rogue insider in the design house or by reverse engineering. Subsequently, with the aid of VLSI-chip-testing tools, the attacker determines the input-vector pattern that produces the accelerated aging of components used in the CE hardware. Subsequently, a software program is made that automatically keeps applying these determined input patterns to produce stress and accelerated aging.

CASE STUDY 1

Figure 11(a) shows an example from [4] of NBTI stress on gate G3 only when the input vector 11111 is applied. For this circuit, only 1 PMOS (of gate G3) is on and experiences NBTI stress (aging). Thus, the performance degradation occurs for G3. However, if an attacker wants to increase the delay degradation further, then NBTI stress must be increased. This is possible by applying different input vectors, which is demonstrated in the next case.

CASE STUDY 2

Figure 11(b) shows an example from [4] of NBTI stress on gates G1 and G4 when the input vector 100111 is applied. For this circuit, 2 PMOS (of gates G1 and G4) are on and experience NBTI stress (aging). Thus, the performance degradation increases.

CASE STUDY 3

Figure 12 shows an example from [4] of NBTI stress on gates G1, G3, and G4 when the input vector 10111 is applied. For this circuit, 5 PMOS (2 PMOS due to G1, 2 PMOS due to G3, and 1 PMOS due to G4) are on in the critical path and experience maximum NBTI stress. Thus, the performance degradation is very high for this circuit.

A DESIGN FOR SECURITY (DETECTION) AGAINST COMBINATIONAL RARE-EVENT TRIGGERED TROJANS RESULTING IN FUNCTIONAL CHANGE

In this article, we introduce a design-for-security (DFS) flow against combinational rare-event triggered Trojans (that results in functional change). A DFS indicates designing with detection capability embedded into it against Trojans. The DFS flow is based on high-level synthesis framework (for

chip design) driven through particle-swarm-optimization (PSO) exploration. Figure 13 shows the DFS flow adopted from [7]. The DFS flow accepts the library details of each component imported from third-party vendors, inputs the application in the form of a control data-flow graph (CDFG), and produces an optimized design that guards against this specific type of Trojan. The DFS flow is described next.

The major block in the DFS flow is a Trojan detection block that provides security by first generating a dual-modular redundant (DMR) structure of the application (CDFG) and then imposing certain hardware-allocation rules on the operations. Creating a DMR structure means generating an original copy and a duplicate copy of the application simultaneously. Once the DMR design is created, then it is concurrently scheduled using a LIST scheduling algorithm based on the data (resource configuration) received from a PSO-driven design space exploration (DSE).

Subsequently, the Trojan specific hardware-allocation rules are imposed, which implies that sister operations of the original and duplicate copies must be assigned to the hardware from distinct vendors. This allocation process can be accomplished by using multiple techniques; thus, optimization here is also performed using a PSO-DSE. Once hardware allocation is performed, then binding and cost evaluation of this DMR design is evaluated and fed to the PSO-DSE block to generate the next resource configuration using the concept of velocity and position upgradation. A particle in the PSO-DSE block is initialized as $X_i = (R_1, \ldots R_D, A_v, UF)$, where R_1 to R_D are the various hardware resource types, A_v indicates the mode of possibilities of achieving distinct vendor-hardware allocation, and UF indicates the loop unrolling factor (in this case, the application is a loop-based CDFG). This process of iteration continues until the PSO-DSE produces a low-cost optimal Trojan secured design. Figure 14 shows the vulnerabilities detectable by the DFS flow in Figure. 13. More details are available in [7] and [8].

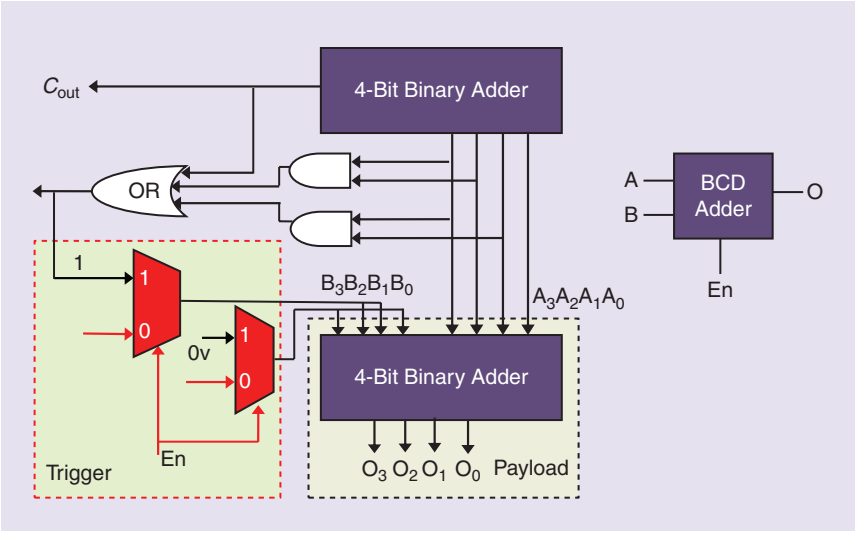


FIGURE 10. The DoS Trojan inserted by an adversary in a BCD adder. When Enable (En) = 0, the designer assumes that this signal enables the adder block and assumes it to be off. However, En is the trigger signal of the Trojan, and at En = 0, the Trojan logic gets triggered and produces no value at the output of muxes ($B_3B_2B_1B_0$). Thus, ($O_3O_2O_1O_0$) yields no output or DoS. The black box BCD adder available to a designer is shown on the right.

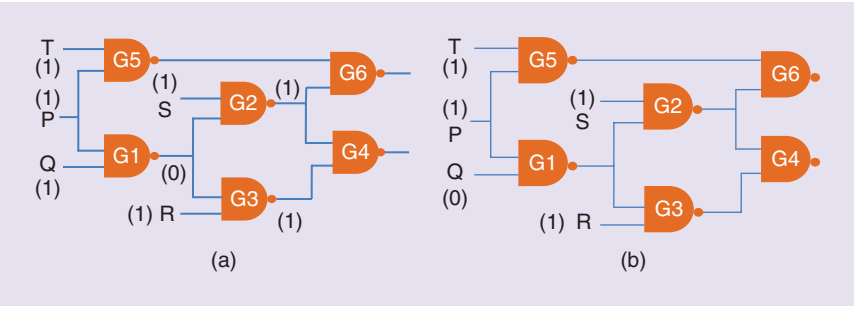


FIGURE 11. (a) The NBTI stress on G3 when PQRST = 11111 is applied as the input vector. (b) NBTI stress on G1 and G4 when PQRST = 10111 is applied as the input vector.

OTHER TROJAN DETECTION TECHNIQUES

There are two types of Trojan detection techniques: invasive and noninvasive. Invasive-type Trojan detection aims to prevent the insertion of Trojans during the design or fabrication of a chip. Trojan insertion may be likely in cases where there is dead space in the chip layout. Dead space is the only available space for inserting Trojans, because the total area of the chip cannot be modified by a rogue element. If a rogue element can extract the netlist of the design from its layout, then with rerouting and better placement/routing, some dead free space can be created, which may enable Trojan insertion. Thus, to prevent such a scenario, the authors in [9] proposed obfuscating the design to make it difficult for reverse engineering.

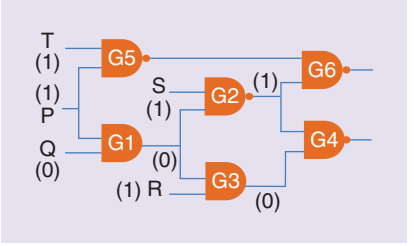


FIGURE 12. The NBTI stress on G3 when PQRST = 10111 is applied as the input vector.

Noninvasive techniques can be categorized into two types: a run-time test and a test-time test. The run-time test techniques employ an online monitoring system that tries to detect suspicious activity during an in-field operation,

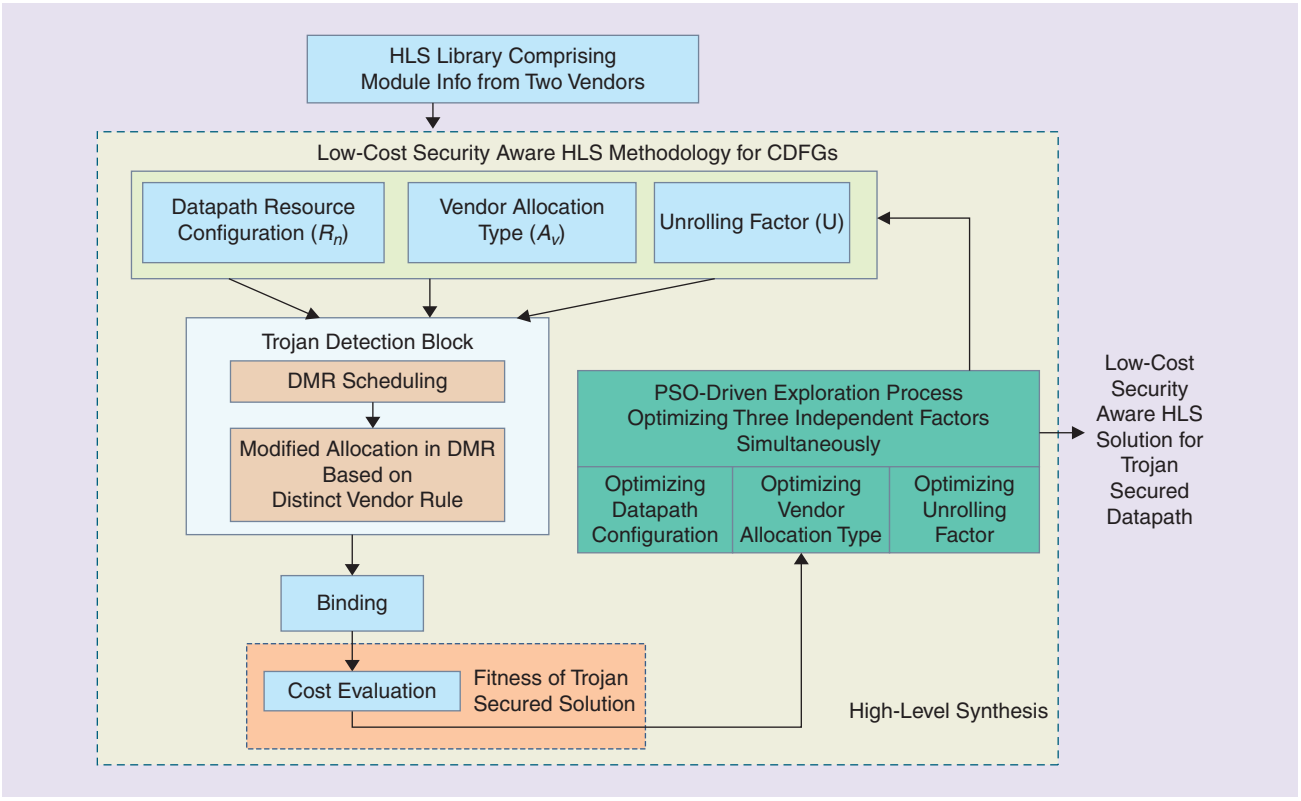


FIGURE 13. The DFS flow against a Trojan resulting in functional change [7].

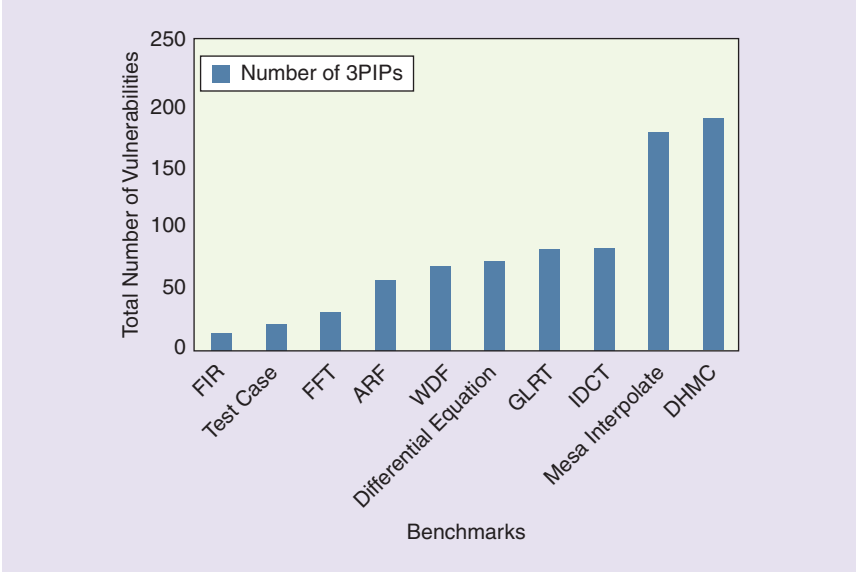


FIGURE 14. The Trojan vulnerabilities detected by DFS flow [7].

while test-time testing enables the detection of Trojan-infected designs before actual deployment. Test-time testing techniques are more advantageous than run-time techniques because they do not incur any hardware overhead; but the disadvantage is the requirement of a golden model (one that is Trojan free), which may be unavailable for IP-based chips.

Test-time techniques can be further classified as either logic-testing or a side-channel type. Logic testing requires many test vectors that are set for detection, as Trojans may vary in size and characteristics. However, side-channel test analysis for Trojan detection is based on observing the impact of a Trojan insertion on side-channel parameters such as power, delay, energy, and current. This is advantageous in the sense that it enables Trojan detection, even if the Trojan circuit does not produce a functionality change.

In contrast, the main bottleneck of this detection process is the effect of a large process variation and measurement noise, which may mask the impact of small Trojans on side-channel parameters. This process of detection, called *integrated circuit fingerprinting*, has been discussed in [10], and more discussion on Trojan detection is available in [11].

CONCLUSION

This article presented hardware vulnerabilities of a CE chip with its motivation from the attacker's perspective. Hardware threats ranging from combinational-triggered Trojans, sequential-triggered Trojans, and NBTI stress are used by an attacker for inducing performance

degradation, and Trojans resulting in functional change, confidential information leakage, and DoS were described with examples provided. Furthermore, DoS against Trojans (resulting in functional change) was presented, and various Trojan-detection techniques for chips/circuits used in CE devices were also highlighted.

ABOUT THE AUTHOR

Anirban Sengupta (asengupt@iiti.ac.in) works in computer science and engineering at the Indian Institute of Technology Indore. He is involved in various capacities including professor, researcher, editor, scientist, and consultant. He earned his B.S. degree from West Bengal University of Technology and his M.S. and Ph.D. degrees from Ryerson University, Toronto, Canada. He is the program chair for the 36th IEEE International Conference on Consumer Electronics 2018 and the Third IEEE International Symposium on Nanoelectronic and Information Systems 2017.

REFERENCES

[1] M. Yoshikawa, R. Satoh, and T. Kumaki, "Hardware Trojan for security LSI," in *Proc. IEEE Int. Conf. Consumer Electronics* 2013, pp. 29–30.

[2] I. Takato, N. Yusuke, Y. Masaya, and K. Takeshi, "Detection technique for hardware Trojans using machine learning in frequency domain," in *Proc. IEEE 4th Global Conf. Consumer Electronics*, 2015, pp. 185–186.

[3] R. S. Chakraborty, Seetharam Narasimhan, and Swarup Bhunia, "Hardware Trojan: Threats and emerging solutions," in *Proc. IEEE Int. High Level Design Validation and Test Workshop*, 2009, pp. 166–171.

[4] O. Sinanoglu, N. Karimi, J. Rajendran, R. Karri, Y. Jin, K. Huang, and Y. Makris "Reconciling the IC test and security dichotomy," in *Proc. 18th IEEE European Test Symp.*, 2013, pp. 176–181.

[5] A. Sengupta, D. Roy, and S. Bhadauria, "Low cost optimized Trojan secured schedule at behavioral level for single & Nested loop control data flow graphs (Invited Paper)," *Elsevier Integr., VLSI J*, Sept. 2016. doi:10.1016/j.vlsi.2016.09.007.

[6] R. Karri, J. Rajendran, K. Rosenfeld, and M. Tehranipoor, "Trustworthy hardware: Identifying and classifying hardware trojans," *Comput.*, vol. 43, no. 10, pp. 39–46, 2010.

[7] A. Sengupta, S. Bhadauria, and S. P. Mohanty, "TL-HLS: Methodology for low cost hardware Trojan security aware scheduling with optimal loop unrolling factor during high level synthesis," *IEEE Trans. Comput. Aided Design Integr. Circuits Syst.*, vol. 36, no. 4, April 2017, pp. 655–668.

[8] A. Sengupta, S. Bhadauria, and S. P. Mohanty, "Low cost security aware high level synthesis methodology," *IET J. Comp. Digit. Technol.*, Aug. 2016. doi:10.1049/iet-cdt .2016.0014.

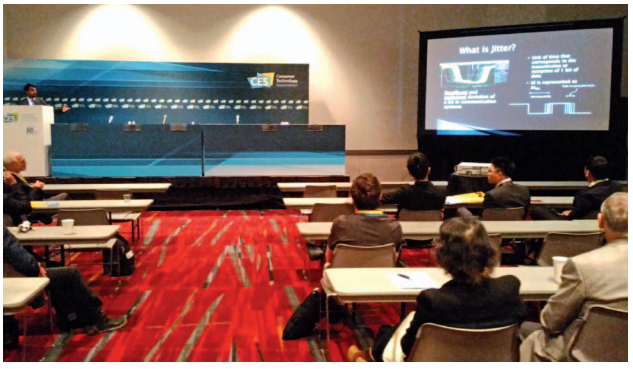
[9] R. S. Chakraborty and S. Bhunia, "Security against Hardware Trojan through a Novel Application of Design Obfuscation," in *Proc. ACM Int. Conf. Computer Aided Design*, San Jose, CA, 2009, pp. 113–116.

[10] D. Agrawal et al., "Trojan detection using IC fingerprinting," in *Proc. IEEE Symp. Security and Privacy*, 2007. doi: 10.1109/SP.2007.36.

[11] A. Sengupta, "Hardware security of CE devices: Threat models and defence against IP Trojans and IP piracy," *IEEE Consum. Electron. Mag.*, vol. 6, no. 1, pp. 130–133, 2017.

Society News *(continued from page 9)*

Coughlin gave tips about how to make an impact with a presentation.



The YP session at the 2017 ICCE.

talked about the analysis of correlated random jitter. During their talks, some of the audience called out with flying toy animals—a duck, a pig, and a cow—and provided suggestions on how to give a better talk. In addition, all of the audience members learned by counterexamples how to effectively improve their presentations.

This session is the first YP event in 2017. The YP committee of the CE Society, of which I am chair, is committed to engaging graduates, postgraduates, and early-career professionals through a range of professional and social activities. The YP committee welcomes the active participation of young members at the CE Society conferences and encourages participants to become members of the CE Society to access a variety of benefits.

More information can be obtained by contacting me at shingo.yamaguchi@ieee.org. I hope to see you at one of the CE Society conferences!

—Shingo Yamaguchi