An Attribute Based Classification of Hardware Trojans

Samer Moein, Salman Khan, T. Aaron Gulliver, Fayez Gebali
Department of Electrical and Computer Engineering
University of Victoria, Victoria V8P 5C2, Canada
Email: {samerm, salmankh, agullive, fayez}@uvic.ca

M. Watheq El-Kharashi
Computer and Systems Engineering Department
Ain Shams University, Cairo 11517, Egypt
Email: watheq.elkharashi@eng.asu.edu.eg

***Abstract*—This paper considers the classification of hardware trojans in semiconductor chips. The phases of the chip production life-cycle are reviewed and opportunities for trojan insertion are discussed. Trojans are classified using a comprehensive attribute taxonomy based on eight categories. A matrix describing the relationships between these attributes is defined which can be used to identify hardware trojans.**

I. Introduction

We begin with several real world examples to motivate this work. In 2009, several countries participated in a multinational air manoeuvre. One of the exercises stipulated that airplanes from country B were to launch missiles against airplanes from country A. The aircraft used by country B were manufactured in country A. The missiles failed to launch even after numerous attempts. A thorough inspection of the airplanes from country B was unsuccessful in determining the reason for the failure of the weapons control systems. After the manoeuvre was over, the airplanes from country B used the missile systems and no problems were encountered. A possible reason for the failure is that a hardware trojan was used to disable these systems [1].

In 2007, Israeli jets bombed a suspected nuclear installation in northeastern Syria. Among the mysteries surrounding this airstrike is the failure of the Syrian radar systems. It has been suggested that the commercial off-the-shelf microprocessors used in these systems may have been fabricated with a hardware trojan which was used to temporarily disable them [2].

There are many documented cases of similar incidents which have caused significant government, industry, and consumer concerns. In response, the Defence Advanced Research Projects Agency (DARPA) initiated the trust in Integrated Circuits (ICs) program to develop techniques for trojan detection [2], [3]. This makes it clear that hardware designers and researchers must be vigilant to the insertion of hardware trojans during all phases of the chip production life-cycle. The classification of these trojans is the main goal of this paper.

Several researchers have proposed taxonomies for hardware trojans based on their attributes [4]–[7]. In [4], trojans were classified based on two categories: trigger and payload. These are in fact activation mechanisms for hardware trojans. In [5], [6], the classification was based on three categories: physical, activation, and action. Although this adds two categories to the previous taxonomy, the classification is not related to the chip life-cycle. In [7], a more detailed classification was developed based on five categories: insertion phase, abstraction level, activation mechanism, effect, and location. This classification considers the chip life-cycle and the target locations, but not the physical characteristics of trojans. A set of attributes was associated with each trojan considered, and some relationships between the attributes were identified.

The classifications proposed in the literature and the existing trojans indicate that numerous relationships exist among the trojan attributes. However, no classification has been developed based on these relationships. In this work, a comprehensive examination of these relationships is presented. As with any circuit, a hardware trojan goes through several production phases as it becomes embedded into the target system. Therefore, studying the life-cycle along with other attributes will provide insight into the relationships between the insertion phase, functionality, logic type, physical characteristics, and location of a trojan.

The approach to classifying hardware trojans presented in this paper has two major advantages over existing results. First, the relationships between the hardware trojan attributes are clearly defined based on a comprehensive taxonomy with eight categories. These relationships can be used to identify missing trojan attributes. This information can be employed to improve chip security during manufacturing, develop trojan detection techniques, and assess chips when covert attacks occur [8]. Second, the production life-cycle of a chip is used to determine how and where a trojan can be inserted into a chip.

The remainder of this paper is organized as follows. Section II reviews the chip production life-cycle. Section III provides a comprehensive hardware trojan taxonomy based on eight categories: insertion, abstraction, effect, logic type, functionality, activation, physical layout, and location. An algebraic approach to investigating hardware trojans is given in Section IV. Finally, Section V provides some concluding remarks.

II. Chip Life-cycle

The vulnerability of ICs to hardware trojan attacks is increasing due to design outsourcing, overseas fabrication and increasing reliance on third-party Intellectual Property (IP). In addition, hardware attacks can originate from the use of unverified design automation tools. Fig. 1 shows the modern IC production life-cycle phases: ***specification***, ***design***, ***fabrication*** (interfacing, masking, wafer fabrication and assembly, and packaging), ***testing*** and ***deployment*** [9]–[12].

Complete protection of a chip during its life-cycle is only possible if it is produced by a trusted source. The stages in

978-1-4673-9971-5/15/$31.00 ©2015 IEEE

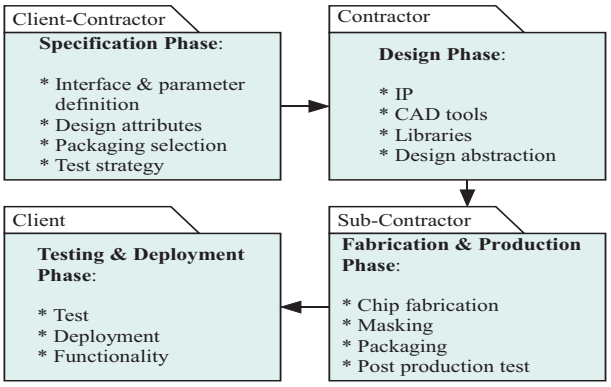

Fig. 1: The integrated circuit (IC) design life-cycle phases.

the production process can be classified as ***trusted***, ***untrusted***, or ***either*** [9]. This is based on the level of control over a given stage. A ***trusted*** stage indicates that the designer has complete control, whereas an ***untrusted*** stage means that the designer has only limited control. ***Either*** indicates that control may be complete or limited. This classification assumes that the specification, testing and assembly phases are trusted. However, the increased complexity of chip design and fabrication requires state-of-the-art infrastructure and very specialized processes at the production facilities. Often the *design* and *fabrication* must be carried out simultaneously, requiring support for multiple capabilities and mixed technologies. This creates significant economic challenges in meeting production quality and demand which has caused increased outsourcing to contractors and subcontractors during the design and manufacturing phases. Unfortunately, outsourcing significantly reduces trust [13]. For example, outsourcing the design phase reduces trust in the fabrication phase. Further, test vectors can be modified during the design phase, which leads to untrusted testing. Even if the testing phase is trusted, it does not imply that the assembly phase is trusted. This can only be ensured if it is done under client control.

There is typically a high level of control over the ***specification phase*** since the client and contractor must work closely to define the system blocks, design specifications, packaging, and test requirements. These are reviewed and validated to confirm the target die size and fabrication process [14]. However, deficiencies in the specification or incomplete validation can lead to vulnerabilities.

There is less control over the ***design phase*** primarily because many manufacturers outsource their designs. This makes standard cell design abstractions, design tools, design kits and design libraries untrusted [15]. The ***fabrication and production phase*** of an IC typically involves hundreds of steps. Computer-Aided Design (CAD) tools are used to define the doping, metallization and glass region masks. These masks are then used to transfer the design onto a silicon wafer. This process is repeated to embed the masks for the IC layers on the silicon. Each layer is tested for functionality and defective chips are discarded after the wafer is cut into individual ICs. The chips are then bonded to a mounting package and contact leads are attached. The mounting package is encapsulated with a plastic coating for protection and identifying part numbers and other data are added. This is followed by the ***testing and deployment phase*** in which ICs undergo final testing before they are distributed [15].

If the production life-cycle is not completely controlled by the designer, reverse engineering, malicious circuit insertion/modification, and IP piracy are possible. Of particular concern is an untrusted IC production facility where hardware trojans can be inserted into chips unknown to the designer.

III. Trojan Taxonomy

The growing application of semiconductor chips in critical applications such as military infrastructure, industrial automation, Supervisory Control And Data Acquisition (SCADA) systems, and the smart grid, has made security a serious concern. Tampering during the IC life-cycle can cause it to respond in an unexpected manner or worse in a manner determined by an attacker. A *hardware trojan* is a malicious component embedded in an integrated circuit which causes abnormal behaviour [16]. Hardware trojans can be implemented in microprocessors, microcontrollers, network and digital signal processors, Field Programmable Gate Array (FPGAs), Application Specific Integrated Circuits (ASICs), and other integrated circuits.

Any attempt to address the concern over hardware trojans should begin with an examination based on the processes involved in the IC production life-cycle. However, it is extremely challenging to characterize, classify, and detect hardware trojans. Trojans have previously been classified by their *physical* characteristics, by the *effects* they cause to a system, and by their *activation* mechanisms [6], [17]. These approaches are based on the assumption that the trojans are inserted only during *fabrication* [7]. Typically the specification and design phases are considered trusted [9], [13], but here the vulnerability of these phases is also considered. In addition, a comprehensive model for trojan classification is presented which is based on eight key categories: insertion (*when was it inserted?*), abstraction (*where was it inserted?*), effect (*what can it do?*), logic type (*what logic type does it employ?*), functionality (*what functionality does it have?*), activation (*what activates it?*), physical layout (*how does it appear?*), and location (*where is it located?*). This classification is illustrated in Fig. 2.

The **insertion** category represents the hardware manufacturing process. This process has many levels and each level is vulnerable to malicious insertions that can cause abnormal functionality [2], [16]. Modifications can be as early as the *specification phase* where parameters such as the size, structure, type, intended function, power, timing and delay can be targeted. Design constraints and deficiencies can be exploited by an attacker. Protocols and functions can also be modified to insert trojans [2], [16]. Alterations to the specification lead to changes in the design which result in the production of a modified IC. In the *design phase*, factors concerning the logical, functional, timing and physical constraints of realizing the design on the target hardware technology are considered. In this phase, designers typically use different IP blocks, design tools, and standard cell libraries and templates. These can contain malicious components if they originate from untrusted or commercial off-the-shelf sources. The *fabrication*

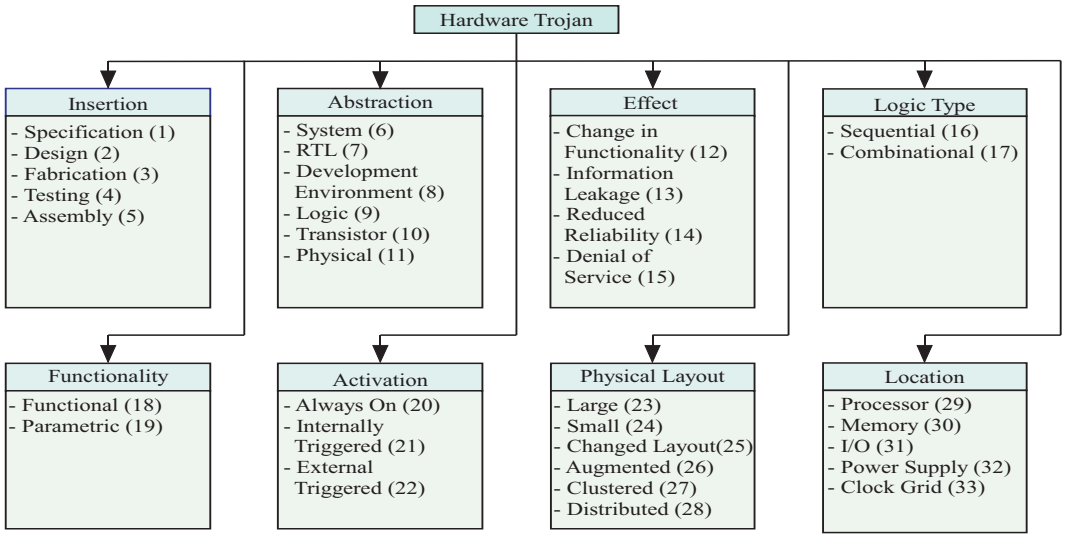

Fig. 2: The hardware trojan taxonomy.

phase involves preparation of the silicon wafers and processes such as masking, photolithography, doping, etching and interconnections to complete the manufacturing and packaging. An attacker can introduce trojans by altering parameters in these processes, modifying the mask geometry and layout, or changing chemical compositions.

In the *testing phase*, hardware trojans can be inserted, or trojans inserted earlier can be concealed from detection. Editing during the design and prototype stages can result in large blocks of unused circuitry from previous iterations. This latent circuitry can be exploited by attackers to reroute signals and trigger trojans [2], [15], [18]. Testing is a critical phase since it is the easiest phase to detect trojans. However, testing at an untrusted facility can lead to test vectors being revealed and detection deficiencies [17]. An attacker can exploit this information to omit or mask test vectors that will reveal hardware trojans. Trojans can also be inserted which will return acceptable values for test vectors [7].

In the *assembly phase*, hardware components are interfaced on the chip. Every interface between components is a possible trojan insertion point. At an interface, improper termination or improper shielding against phenomena such as electromagnetic coupling makes the chip susceptible to exploitation by an attacker. For example, deficiencies can be identified to inject faults or gather sensitive information.

The **abstraction** category represents the level at which a hardware trojan is introduced [2], [16]. The sophistication required to inject a trojan at a given abstraction level varies depending on the level.

A trojan can be introduced in the *system level* by altering interconnections, hardware modules, or communication protocols. This does not require a high level of sophistication. At the *Register Transfer Level (RTL)*, signals, registers and boolean functions are described in the function modules. Numerous trojan insertion opportunities exist in this level since an attacker can gain control over the hardware functionality [2], [7]. The *development environment* of an IC involves the simulation, verification, validation and synthesis. Software can be inserted into this abstraction level through CAD tools and scripts to hide the effects of hardware trojans [19]. The *logic (gate level)* is also prone to trojan insertion, but it is considered relatively secure since tampering requires a high level of sophistication. Logic design alteration is possible at the gate level or in the netlist [18].

The *transistor* level provides attackers with the opportunity to control circuit parameters such as power and timing. Trojans can be inserted by resizing or deleting existing transistors, or inserting new transistors, to modify the circuit functionality and characteristics. Modifications at the *physical* level involve the transistors and/or layout and are typically achieved by changing circuit parameters that affect the reliability and/or functionality.

The **effect** category describes the disruption or effect a trojan can have on a chip. These effects are: *change in functionality, information leakage, reduced reliability, and Denial of Service (DoS)*. A *change in functionality* is caused by trojans that introduce new logic or bypass existing logic to produce unexpected results. This can also be achieved by deleting logic [17]. A trojan can cause *information leakage* through a covert or existing channel. These channels may be radio frequency based or via JTAG or RS232 interfaces, and provide backdoor access to assist in compromising the chip. Information such as encryption keys can also be leaked through thermal or optical patterns created by the hardware [8], [18]. Hardware trojans can also cause *reduced reliability* by altering interface, functional or circuit characteristics such as path delay and power consumption. Increased power consumption may cause the ambient temperature of the circuit to rise above normal operating levels and/or cause faster battery depletion. Finally, a *DoS* trojan modifies device parameters to exhaust on-board resources such as power or memory, or introduces computational delays to degrade performance or create malfunctions.

The **logic type** category is based on the circuit logic that

triggers the trojan. A *combinational* trojan uses a particular logic value at one or more circuit locations as the trigger. A *sequential* trojan is triggered by a sequence of conditions after a given period of operation [20], [21].

The **functionality** category is based on whether hardware trojans are *functional* or *parametric*. A *functional* trojan introduces a change in the functionality of the device [22]. A *parametric* trojan exploits the parametric effects of the device circuitry such as power consumption and thermal and delay profiles [23]. This is achieved by weakening transistors, modifying the length and/or thickness of wires, or changing physical geometries [24].

The **activation** category is based on whether a trojan is always on or triggered. An *always on* trojan is always active. A triggered trojan will cause some unintended or malicious effect only when activated [18]. The trigger can be either *external* or *internal*. An *externally triggered* trojan is activated via an external signal received by an antenna or sensor that interacts with the outside world. An attacker can then be at a geographically distant location. An *internally triggered* trojan waits for an internal condition which can be a sequence of one or more events that occur in the system. This condition is typically an internal logic state or a pattern of input/output signals [17].

The **physical layout** category is based on the physical characteristics of hardware trojans. Hardware trojans have been developed with various sizes and layouts to evade detection. Some trojans are very *small* and thus virtually undetectable by inspection of the power consumption or heat dissipation [25]. Other trojans are *large* and often employ unused circuitry in the device to avoid detection. The size of trojans is determined by the number of deleted, added or compromised components in the chip. The distribution of a trojan is based on the layout of the trojan on the chip. A *clustered* trojan has a topology in which the components are close to each other. A *distributed* trojan has a sporadic topology and can be dispersed throughout the chip. Some trojans employ a *changed layout* structure where an existing layout is modified. If the layout is added to it is known as an *augmented* circuit [24].

The **location** category is based on where the trojan is located. As mentioned above, trojans can be *distributed* or *clustered* among the IC components. They can be located in components such as the *processor, memory, input/output, power supply, or clock grid*. A trojan injected into the *processor* can be located in the logical units so that it can change instruction or execution cycles. Trojans in the *memory* units or interfaces can create incorrect addresses, modify memory contents, or enable/disable read/write instructions. The *input/output* peripherals interface with external devices through communication and data buses such as serial ports. Trojans located here can modify the data or alter the way external devices communicate with IC components. Trojans in the *power supply* can create effects such as denial of service or reduced reliability. This is achieved by varying the current and/or voltage supply to the chip to cause malfunctions or abnormal behaviour. Finally, a trojan in the *clock grid* can cause variations in the clock frequency, or skip or freeze the clock signals supplied to chip modules [7].

Table I presents the hardware trojan attributes considered in existing taxonomies and those in the proposed taxonomy. This shows that the proposed classification illustrated in Fig. 2 is more comprehensive that those given in [4]–[7].

TABLE I: Comparison of Hardware Trojan Taxonomies

k	Attribute	[4]	[5]	[6]	[7]	Proposed
1	Specification				✓	✓
2	Design				✓	✓
3	Fabrication				✓	✓
4	Testing				✓	✓
5	Assembly				✓	✓
6	System				✓	✓
7	RTL				✓	✓
8	Development Environment				✓	✓
9	Logic				✓	✓
10	Transistor				✓	✓
11	Physical				✓	✓
12	Change in Functionality		✓	✓	✓	✓
13	Information Leakage			✓	✓	✓
14	Reduced Reliability		✓	✓	✓	✓
15	Denial of Service				✓	✓
16	Sequential	✓				✓
17	Combinational	✓				✓
18	Functional		✓	✓		✓
19	Parametric		✓	✓		✓
20	Always On	✓	✓		✓	✓
21	Internally Triggered	✓	✓	✓	✓	✓
22	Externally Triggered	✓	✓	✓	✓	✓
23	Large		✓	✓		✓
24	Small		✓	✓		✓
25	Changed Layout			✓		✓
26	Augmented			✓		✓
27	Clustered		✓	✓		✓
28	Distributed		✓	✓		✓
29	Processor				✓	✓
30	Memory				✓	✓
31	I/O				✓	✓
32	Power Supply				✓	✓
33	Clock Grid				✓	✓

IV. Algebraic Approach to Hardware Trojan Attributes

In this section, an algebraic approach to hardware trojan attributes is presented. Fig. 2 provides a comprehensive classification of these attributes. The attributes belong to one of four levels: chip life-cycle, abstraction, properties, and location, as shown in Fig. 3. The chip life-cycle level is equivalent to the insertion category in Fig. 2. This indicates the phase of the chip life-cycle that the trojan is inserted. Therefore, the abstraction category in Fig. 2 is the same as the abstraction level in Fig. 3. The trojan properties level contains the properties a trojan may have according to the chip life-cycle and abstraction levels. The possible location for a trojan is based on the chip life-cycle, abstraction, and properties levels. Each trojan has a particular combination of attributes which define its characteristics and the potential locations for insertion.

The attributes of a hardware trojan can be expressed using an $n \times n$ trojan matrix defined as

$$\mathbf{R} = \begin{bmatrix} R_1 & R_{12} & 0 & 0 \\ 0 & R_2 & R_{23} & 0 \\ 0 & 0 & R_3 & R_{34} \\ 0 & 0 & 0 & R_4 \end{bmatrix}$$

where n is the number of attributes.

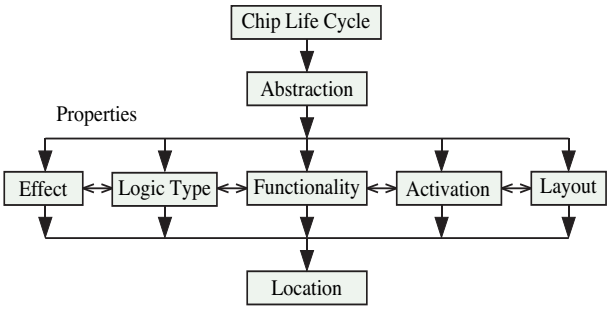

Fig. 3: The four hardware trojan levels.

This matrix is used to represent the set of attributes in Fig. 2 and the hardware trojan levels in Fig. 3. The trojan characteristics can be inferred from the paths in $\mathbf{R}$. The hardware trojan matrix $\mathbf{R}$ has four square submatrices $\mathbf{R_1}$, $\mathbf{R_2}$, $\mathbf{R_3}$, and $\mathbf{R_4}$ which represent the levels in Fig. 3. The transfer matrices $\mathbf{R_{12}}$, $\mathbf{R_{23}}$, and $\mathbf{R_{34}}$ represent the connections between the levels. An empty submatrix indicates that all entries are zero. Further, submatrix $\mathbf{R_4} = 0$ since the location attributes are terminal attributes and so are not interconnected. The usefulness of $\mathbf{R}$ in classifying hardware trojans is illustrated below.

A. Single Element Example

The matrix element $\mathbf{R}(17, 29) \equiv \mathbf{R}$(Combinational, Processor) = 1 indicates that if there is a trojan inserted in a processor, it may be a combinational circuit.

B. Row Example

For matrix row 6:
$\mathbf{R}(6,\ j) \equiv \mathbf{R}$ (System, j) = 1;
$\mathbf{R}(6,\ 7) \equiv \mathbf{R}$ (System, RTL),
$\mathbf{R}(6, 12) \equiv \mathbf{R}$ (System, Change Function),
$\mathbf{R}(6, 13) \equiv \mathbf{R}$ (System, Leak Information),
$\mathbf{R}(6, 15) \equiv \mathbf{R}$ (System, DoS),
$\mathbf{R}(6, 18) \equiv \mathbf{R}$ (System, Functional),
$\mathbf{R}(6, 19) \equiv \mathbf{R}$ (System, Parametric),
$\mathbf{R}(6, 20) \equiv \mathbf{R}$ (System, Always On) = 1,
implies that if a trojan is introduced at the system abstraction level, it may lead to a combination of RTL, changed function, leaked information, DoS, functional, parametric, or always on.

C. Column Example

For matrix column 6:
$\mathbf{R}(i,\ 6) \equiv \mathbf{R}$(i, System) = 1;
$\mathbf{R}(1, 6) \equiv \mathbf{R}$(Specification, System),
$\mathbf{R}(4, 6) \equiv \mathbf{R}$(Testing, System),
$\mathbf{R}(5, 6) \equiv \mathbf{R}$(Assembly, System) = 1,
implies that if a trojan is inserted at the system abstraction level, it may be because of an incomplete or modified specification, control of the testing to evade trojan detection, or modification in the assembly phase.

V. Conclusion

ICs are now a part of most systems including critical infrastructure and critical applications, which makes hardware trojans a serious concern. In this paper, a comprehensive taxonomy of hardware trojan attributes was presented, and a hardware trojan matrix developed to show their connections.

$\mathbf{R} =$

A	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
1	0	1	0	0	0	1	0	0	0	0	0																						
2	0	0	1	0	0	0	1	0	0	0	0																						
3	0	0	0	1	0	0	0	0	0	0	1																						
4	0	0	0	0	1	1	0	0	1	0	0																						
5	0	0	0	0	0	1	0	0	0	0	0																						
6						0	1	0	0	0	0	1	1	0	1	0	0	1	1	1	0	0	0	0	0	0	0	0					
7						0	0	1	0	0	0	1	0	0	1	1	1	1	0	1	1	1	1	1	0	0	0	0					
8						0	0	0	1	0	0	1	0	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1					
9						0	0	0	0	1	0	1	0	0	1	1	1	1	0	1	1	1	0	0	0	0	0	0					
10						0	0	0	0	0	1	1	0	1	0	0	1	1	1	1	0	0	0	1	0	1	1	0					
11						0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	0	0	1	1	1	1	1	1					
12												0	0	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
13												0	0	0	0	0	1	0	1	1	0	1	0	1	0	1	1	1	1	1	1	1	1
14												0	0	0	0	0	0	0	1	1	0	0	0	1	0	1	1	1	1	1	1	1	1
15												0	0	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1
16												1	0	0	1	0	0	1	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1
17												1	1	0	1	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
18												1	0	0	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
19												0	1	1	0	0	0	0	0	1	0	0	0	1	0	1	0	1	0	0	1	1	1
20												1	1	1	1	0	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1
21												1	0	0	1	1	1	1	0	0	0	0	1	1	0	1	1	1	1	1	1	1	1
22												1	1	0	1	1	1	1	0	0	0	0	0	1	0	1	1	0	1	1	1	1	1
23												1	0	0	1	1	1	1	0	1	1	0	0	0	1	1	1	1	1	1	0	0	0
24												1	1	1	1	0	1	1	1	1	1	1	0	0	0	1	1	0	1	1	1	1	1
25												1	0	0	1	1	1	1	0	1	0	0	1	0	0	1	1	0	1	1	1	1	1
26												1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
27												1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	0	1	1	1	1	1
28												1	1	1	1	1	1	1	1	1	1	0	1	0	0	1	0	0	1	1	1	1	1
29																																	
30																																	
31																																	
32																																	
33																																	

This matrix can be used to characterize and classify hardware trojans. Previous results in the literature examined some trojan characteristics while others are ignored, which limits their usefulness in describing hardware trojans. Further, they assumed that some phases of the chip life-cycle are trusted, but any phase can be vulnerable to hardware trojan insertion. This paper considered the attributes of hardware trojans for all phase of this life-cycle. Further, the classifications in [4]–[7] do not consider the relationships between hardware trojan attributes.

REFERENCES

[1] Anonymous soldier from country B, personal communication.

[2] S. Adee, "The hunt for the kill switch," *IEEE Spectrum*, vol. 45, no. 5, pp. 34–39, 2008.

[3] Defense Science Board. (2005) Task force on high performance microchip supply. [online]. Available: http://www.acq.osd.mil/dsb/reports/ADA435563.pdf

[4] F. Wolff, C. Papachristou, S. Bhunia, and R. S. Chakraborty, "Towards trojan-free trusted ICs: Problem analysis and detection scheme," in *Proc. Conf. on Design, Automation and Test in Europe*, 2008, pp. 1362–1365.

[5] R. M. Rad, X. Wang, M. Tehranipoor, and J. Plusquellic, "Power supply signal calibration techniques for improving detection resolution to hardware trojans," in *Proc. Int. Conf. on Computer-Aided Design*, 2008, pp. 632–639.

[6] X. Wang, M. Tehranipoor, and J. Plusquellic, "Detecting malicious inclusions in secure hardware: Challenges and solutions," in *Proc. Int. Workshop on Hardware-Oriented Security and Trust*, 2008, pp. 15–19.

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

[8] S. Moein, F. Gebali, and I. Traore, "Analysis of covert hardware attacks," *J. Convergence*, vol. 5, no. 3, pp. 26–30, 2014.

[9] B. Sharkey. (2007) TRUST in integrated circuit program - briefing to industry. [online]. Available: http://cryptocomb.org/DARPA-TrustinIntegratedCircuitsProgram.pdf

[10] T. Zhou and T. B. Tarim, "An efficient and well-controlled IC system development flow: Design approved specification and design guided test plan," in *Proc. Int. Symp on Circuits and Systems*, 2005, pp. 2775–2778.

[11] M. Tehranipoor and C. Wang (Eds.), *Introduction to Hardware Security and Trust*. Springer, New York, NY, 2012.

[12] S. Padmanabhan. (2013) Discover a better way to go from C-level to synthesis for SoC designs. [online]. Available: http://electronicdesign.com/technologies/discover-better-way-go-c-level-synthesis-soc-designs

[13] D. R. Collins, "TRUST, a proposed plan for trusted integrated circuits," in *Proc. Conf. on Microcircuit Applications and Critical Tech.*, 2006, pp. 276–277.

[14] StarChip, Product provider for semiconductor market. [online]. http://http://www.starchip-ic.com/Download/flyer_STARCHIP.pdf.

[15] K. M. Goertzel and B. A. Hamilton, "Integrated circuit security threats and hardware assurance countermeasures," *Crosstalk*, vol. 26, no. 6, pp. 33–38, 2013.

[16] M. Banga and M. S. Hsiao, "A region based approach for the identification of hardware trojans," in *Proc. Int. Workshop on Hardware-Oriented Security and Trust*, 2008, pp. 40–47.

[17] M. Tehranipoor and F. Koushanfar, "A survey of hardware trojan taxonomy and detection," *IEEE Design and Test of Computers*, vol. 27, no. 1, pp. 10–25, 2010.

[18] A. Iqbal. (2013) Security threats in integrated circuits. [online]. Available: http://sdm.mit.edu/security-threats-in-integrated-circuits/

[19] J. A. Roy, F. Koushanfar, and I. L. Markov, "Circuit CAD tools as a security threat," in *Proc. IEEE Int. Workshop on Hardware-Oriented Security and Trust*, 2008, pp. 65–66.

[20] L. Ni, S. Li, J. Chen, P. Wei, and Z. Zhao, "The influence on sensitivity of hardware trojans detection by test vector," in *Proc. Commun. Security Conf.*, 2014, pp. 1–6.

[21] S. Narasimhan, X. Wang, D. Du, R. S. Chakraborty, and S. Bhunia, "TeSR: A robust temporal self-referencing approach for hardware trojan detection," in *Proc. IEEE Int. Workshop on Hardware-Oriented Security and Trust*, 2011, pp. 71–74.

[22] G. T. Becker, A. Lakshminarasimhan, L. Lin, S. Srivathsa, V. B. Suresh, and W. Burelson, "Implementing hardware trojans: Experiences from a hardware trojan challenge," in *Proc. IEEE Int. Conf. on Computer Design*, 2011, pp. 301–304.

[23] R. Kumar, P. Jovanovic, W. Burleson, and I. Polian, "Parametric trojans for fault-injection attacks on cryptographic hardware," in *Proc. Workshop on Fault Diagnosis and Tolerance in Cryptography*, 2014, pp. 18–28.

[24] X. Wang, H. Salmani, M. Tehranipoor, and J. Plusquellic, "Hardware trojan detection and isolation using current integration and localized current analysis," in *Proc. IEEE Int. Symp. on Defect and Fault Toleranc"e of VLSI Systems*, 2008, pp. 87–95.

[25] W. Danesh, J. Dofe, and Q. Yu, "Efficient hardware trojan detection with differential cascade voltage switch logic," *VLSI Design*, vol. 2014, article ID 652187, pp. 1-11, 2014.