www.ietdl.org

Published in IET Computers & Digital Techniques
Received on 4th March 2013
Revised on 7th July 2014
Accepted on 6th August 2014
doi: 10.1049/iet-cdt.2014.0039

Special Issue: Hardware Security

ISSN 1751-8601

Hardware Trojans: current challenges and approaches

Nisha Jacob, Dominik Merli, Johann Heyszl, Georg Sigl

Hardware Security, Fraunhofer AISEC, Garching, Germany
E-mail: nisha.jacob@aisec.fraunhofer.de

Abstract: More and more manufacturers outsource parts of the design and fabrication of integrated circuits (ICs) for cost reduction. Recent publications show that such outsourcing can pose serious threats to governments and corporations, as they lose control of the development process. Until now, the threat of hardware Trojans is mostly considered during fabrication. Third party intellectual properties (IPs) are also gaining importance as companies wish to reduce costs and shorten the time-to-market. Through this study, the authors argue that the threat of Trojans is spread throughout the whole IC development chain. They give a survey of both hardware Trojan insertion possibilities and detection techniques. Furthermore, they identify the key vulnerabilities at each stage of IC development and describe costs of hardware Trojan insertion and detection. This way, the threat level based on feasibility of Trojan insertion and the practicability of Trojan detection techniques is evaluated. Lately, detection techniques address the issue of including third party IP. However, those techniques are not sufficient and need more research to effectively protect the design. In this way, the authors' analysis provides a solid base to identify the issues during IC development, which should be addressed with higher priority by all entities involved in the IC development.

1 Introduction

As integrated circuit (IC) designs are steadily becoming more complex, semiconductor companies are not always able to afford the fabrication equipment for small feature sizes. This leads to the outsourcing of parts of the IC design and fabrication to other companies around the world. The outsourcing enables adversaries to maliciously manipulate the design of an IC, without the knowledge of the semiconductor company. However, based on recent trends [1–4], we see that it is not just the outsourcing of fabrication that is prone to malicious manipulations, but also the entire IC development chain.

The different types of malicious manipulations in an IC are those that weaken the system, give unauthorised access to the system and directly modify the functionality of the system. Weakening refers to manipulations that compromise the system's performance or reliability without directly manipulating the IC design or endangering the data integrity. For example, Shiyanovskii *et al.* [5] showed that if a higher temperature is used during IC fabrication, it will lead to early aging in an IC. Unauthorised accesses, also known as backdoors in an IC, can be used for debugging purposes [6], but can also be fatal and maliciously used to insert, modify or leak information of the IC design. Modifying functionality through the malicious insertion, deletion or modification of the original IC design is known as hardware Trojans. Such manipulations are done covertly without affecting the normal operation of the IC. Hardware Trojans pose a threat not only to security sensitive devices used for military or financial systems, but also basic home appliances. In this paper, we mainly focus on hardware Trojans and their effects.

Hardware Trojans have been gaining a lot of attention in the past few years by both, researchers and governmental organisations. Various publications and projects undertaken by governments and defence organisations [7–9] are indicators of the growing fear of hardware Trojans among them. Hardware Trojans are built to gain access to secure devices and their data. The advantage of hardware Trojans is that access to a whole series or a batch of chips can be gained by manipulating the design or fabrication of an IC. Alternatively, side channel analysis (SCA) and reverse engineering (RE) are used to gain access to a secure device, but they do not scale well for multiple devices. There is a significant overhead to carry out the manipulations needed for the analysis on each device. Although in the case of hardware Trojans, the manipulations are inserted only once during design or fabrication and can be exploited with lower efforts.

A Trojan is composed of two parts: 'trigger' and 'payload'. The trigger is the activation signal, whereas the payload is the main Trojan functionality. Trojans are known to be dormant until they are triggered. Some of the typical characteristics of Trojans are (i) not changing the physical form and number of inputs and outputs of the original IC, (ii) being atleast 3–4 orders of magnitude smaller than the original circuit, (iii) remaining undetected during regular test phases and (iv) operating covertly during the normal IC operation [10].

As a lot of attention is drawn to this topic and a lot of research is being performed, we would like to see whether it is going in the right direction. There have been many Trojan designs and detection techniques proposed in the literature. In this paper, we give a survey of both hardware Trojan designs and detection schemes currently known.

Tehranipoor and Koushanfar [11] have also given a survey of Trojan taxonomy and detection techniques, but this does not cover the current threats and Trojan designs. They mostly focus on threat of hardware Trojans during fabrication and Trojan detection techniques post fabrication. Additionally, we also discuss the feasibility of Trojan insertion and the cost associated with it. We identify the Trojan types at each stage of IC development and production, and show how the separation of the different development processes gives an adversary the opportunity to maliciously modify ICs. We identify the entities capable of inserting Trojans and investigate the practicability of previously proposed Trojan detection techniques. This analysis allows us to judge the threat level that is essential for future researchers and semiconductor companies to develop efficient techniques to protect the integrity of their products.

The remainder of this paper is organised as follows: in Section 2 we give an overview of the IC development chain. In Section 3, we describe the vulnerabilities in the IC development chain and a survey of previously proposed Trojan designs. In Section 4, we classify the hardware Trojan detection techniques and provide a survey. Finally, we conclude this paper in Section 5.

2 IC development chain overview

Since today the semiconductor development is a quite complicated process, the possibilities of hardware Trojan insertion and the ways to protect them against malicious manipulations are very diverse and hard to understand. Therefore, we create Fig. 1, to give a comprehensive overview of the research area.

The IC development chain can be divided into four main phases that are described on the vertical axis in Fig. 1. Along with the different design phases, the possibilities of Trojan insertion vary significantly from manipulating the specification to highly technical manipulations as indicated by lower left axis in Fig. 1. The broad range of detection techniques is shown on the lower right axis in Fig. 1. Through the course of this paper, we will describe the threats of Trojan insertion at each development stage and the corresponding detection techniques that can be applied.

In this section, we give a brief explanation of the four main phases of IC development. The remaining two axes, the Trojan insertion phases and Trojan detection techniques, will be described in Sections 3 and 4, respectively.

2.1 Functional design

During this stage, the functional behaviour, interfaces and protocols are specified. First, a high-level device specification is described, followed by the device architecture. The specification and architecture are in most cases described in plain text leaving room for inconsistencies. Although in some cases they may be formally defined using architectural description languages. Later, the logical functionality is described in the register transfer level (RTL). This is done using hardware description languages like Verilog or VHDL. Furthermore, third party intellectual property (IP) RTL blocks, like processor cores, are integrated into the design in this stage.

2.2 Physical design

Once the logical functionality is described, the RTL description is synthesised to lower abstraction levels and

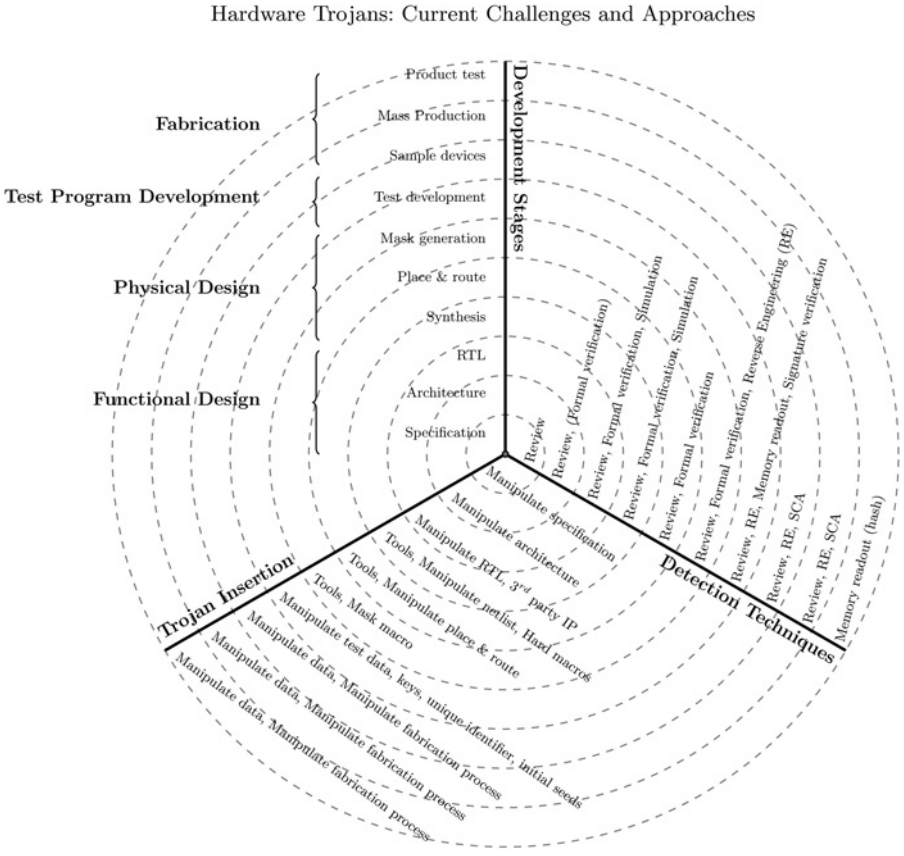

Fig. 1 *Vulnerabilities in an IC development chain*

converted to the technology specific netlist using electronic design automation (EDA) tools. Next, the placement and routing (P&R) is performed. The gates of the netlist are placed on a two-dimensional (2D) floorplan and then routed depending on the design constraints like timing, area and power. During this stage, third party hard macros such as analogue macros and memories are integrated into the design. Once it is complete, the mask layout is generated. The mask set is later sent to the foundry which produces the final IC. The graphic database system version II (GDSII) is a standard used in the industry to store the mask layout for a design. Some big foundries have their own mask shop but others rely on third party mask shops.

2.3 Test program development

In this phase, the test programmes are developed, which are later used to configure and test the IC. For example, boot loaders, keys, unique identifiers and seeds for pseudo random number generators (RNGs) are developed. Furthermore, scan chains used to detect manufacturing faults or built-in-self-tests used to test the memory and functional tests for the IC are designed during this phase. Analogue parameters are also developed to check for compliance. Electrical or optical fuses to be blown are defined by the test program. In the case of embedded read only memory (ROM)-based devices, the firmware is designed and sent to the foundry, which later downloads the firmware onto the devices [12].

2.4 Fabrication

Prior to the mass production, sample devices are fabricated to validate the desired functionality. Once the sample devices are fully characterised and tested, the mass production of the device is carried out. Firmware for embedded devices is downloaded during this stage and each device is then validated using the test programmes developed in the previous stage. Finally, the ICs are packaged and shipped for integration into the end product.

3 Hardware Trojan insertion and design

Hardware Trojans are designed to gain access to secure devices through malicious manipulations, insertion or deletion of circuitry of an IC design. In this section, we describe the vulnerabilities and the types of Trojans that can be inserted in each stage of IC development as shown in the 'Trojan Insertion' branch of Fig. 1. We also describe previously published Trojan designs based on the characteristics described below and summarise them in Table 1. Tehranipoor *et al.* [13] have proposed a similar taxonomy for Trojan insertion. However, we also evaluate the threat and feasibility of Trojan insertion at each development stage.

Hardware Trojan designs can be described based on their physical properties, activation and action characteristics [14].

Physical properties: They can be further divided with respect to the following attributes:

- *Type:* describes whether there has been addition/deletion of gates or modification of the wires.
- *Size:* the number of components in the Trojan circuitry.
- *Structure:* refers to whether or not modifications are made to the original layout.
- *Distribution:* describes whether the trigger and payload circuit are placed together or distributed.

Activation characteristics: Describes the characteristics of the trigger signal:

- *External signal:* this way the attackers can turn on the Trojan at a time of their choice.
- *Internal signal:* refers to Trojans which are always active or conditionally activated by a sensor value or an internal signal pattern.

Action characteristics: Describes the characteristics of the payload:

- *Modify functionality:* Trojans that alter the original functionality of the IC.
- *Transmit information:* Trojans that leak information.
- *Denial-of-service (DoS):* Trojans that temporarily or permanently deny or interrupt services of the device.

3.1 Functional design

An adversary can manipulate the functional design by altering the specification and architecture or by including malicious third party IP cores. There can be several entities affecting these stages, for instance, a malicious insider in the semiconductor company or even a third party like a competitor, research institution or governmental organisation.

- *Malicious insider in the semiconductor company:* A malicious insider has unrestricted access to the functional

Table 1 Hardware Trojan survey

	Possible insertion phases	Action characteristics			External trigger	Internal trigger	
		Modify functionality	Transmit information	DoS		Always on	Condition based
Jin and Makris [18]	functional design			•		•	
Chakraborty *et al.* [19]	functional design			•		•	
Agrawal *et al.* [10]	functional design	•					•
King *et al.* [20]	functional design	•			•		
Lin *et al.* [21]	functional design		•			•	
Kutzner *et al.* [23]	functional design		•				•
Muehlberghuber *et al.* [24]	physical design			•	•		
Bhasin *et al.* [25]	physical design	•			•		
Becker *et al.* [26]	fabrication	•				•	

doi: 10.1049/iet-cdt.2014.0039

design. They can modify the specification, so as to leave room for insertion of a Trojan during a later design phase. An insider can also manipulate the RTL and insert malicious code.

- *Third party:* Third parties have different opportunities to influence the specification. A third party contractor who develops parts of the specification for the semiconductor company can include malicious elements. Governmental organisations can force semiconductor companies to include malicious elements or include weaknesses in the functional design through laws [15]. Third parties can also directly influence the standard committees and the standards itself [16, 17]. The designers would be obliged to comply with the standards. Standards are not considered as Trojans but they can weaken the system just as much as a Trojan.

There have been several proposed Trojan designs that can be inserted in the RTL. The different Trojan designs that can be inserted range from DoS and modify functionality to those that transmit information. Trojans can be built to deny services of the device temporarily or permanently without compromising the integrity of the data, like DoS Trojans. For example, a counter-based Trojan is used to increase the power consumption of the circuit [18]. Agrawal *et al.* [10] use a counter-based Trojan that acts like a time bomb. When a particular count is reached, the system shuts down. The advantage of these Trojans is their small size because of which their effects are normally camouflaged by the circuit noise, hence making their detection very difficult. Chakraborty *et al.* [19] insert a ring oscillator (RO) network in an unencrypted configuration bitstream of an field programmable gate array (FPGA). The Trojan is inserted as a standalone circuit to increase the overall temperature, which results in early aging of the FPGA. This is achieved by disabling the cyclic redundancy check (CRC) function. A CRC check is generally performed to ensure the integrity of the bitstream.

Trojans are also designed so as to modify the original functionality of the IC. Comparator-based Trojans can be used to monitor two or more signals for a particular value. Once this value is detected, the signal values are altered. Agrawal *et al.* [10] describe such a comparator-based Trojan in an RSA circuit, which is conditionally triggered and then alters the signal values of the host system. King *et al.* [20] use a hardware–software co-design to insert a Trojan in the Illinois Malicious Processors. They design a hardware that supports post-fabrication Trojan insertion. A malicious software can be inserted post-fabrication by sending a specific sequence of bytes. Once the trigger sequence is received, the memory privileges are disabled giving the software access to restricted memory regions. To achieve this, they modify the data cache of the processor to add an additional state and circuitry in the memory management unit.

Trojans can be designed to actively leak data such as secret keys from the IC. Lin *et al.* [21] were the first to use the power side channel as a hardware Trojan that is always on and transmits information. The Trojan transmits the key of the block cipher advanced encryption standard (AES) from the cryptographic core. Furthermore, they use code division multiple access to encode the key so only the attacker would be able to decode it correctly. Jin and Makris [22] leak the encryption key of the block cipher data encryption standard (DES) through the wireless channel. The 56 bit DES key is hidden in the process variations that occur in amplitude and frequency of the wireless transmission. The authors propose to leak 1 bit at a time. After 56 consecutive encryptions, the whole key can be constructed. Kutzner *et al.* [23] design a Trojan that transmits the last round key of AES instead of the cipher text, thus transmitting sensitive information. Using this method, the whole AES key can be computed. This Trojan is conditionally triggered by a 64 bit trigger sequence.

3.2 Physical design

There are several different Trojan types that an adversary can insert during this stage:

- *Malicious insider in the semiconductor company:* A malicious insider can easily modify the design files and insert macros during the design synthesis. Furthermore, they can also modify the design tools or even alter the P&R so as to make room for the Trojan circuitry.
- *Third party:* Semiconductor companies buy third party IP cores, which may include malicious manipulations. Third party IP providers may not have a very controlled development chain. In addition, semiconductor companies use EDA tools for many critical design stages. A malicious entry in these tools can directly affect the hardware design of many designers. Semiconductor companies may outsource the mask generation to a third party company. This gives third party companies the opportunity to maliciously include mask macros in the GDSII.

Muehlberghuber *et al.* [24] reverse engineer the GDSII mask layout and insert a Trojan. They insert an externally activated DoS Trojan that is waiting for a 30 bit kill sequence.

Bhasin *et al.* [25] insert a Trojan by modifying the GDSII. They use an externally triggered Trojan that inserts a 1 bit fault in the eighth round of AES. With two pairs of correct and faulty cipher text, they are then able to retrieve the AES key using differential fault attacks.

3.3 Test program development

Test programmes can be manipulated by a malicious insider or third parties:

- *Malicious insider in the semiconductor company:* The malicious insider may modify the keys or add additional keys during the test development. They can also modify the boot loader of the device. Initial seeds to pseudo RNG can be manipulated and set to a number that is known to have certain weaknesses. The debug interface can also be left open, which can later be used to exploit the device.
- *Third party:* A third party providing IP cores to a semiconductor company can provide malicious test programmes.

3.4 Fabrication

Malicious manipulations at the time of fabrication may be because of malicious personnel in the foundry or third parties influence on the fabrication process:

- *Third party:* In the fabrication stage, again governments can regulate the production through laws. In addition, foundries use third party tools that can be manipulated.
- *Malicious personnel in the foundry:* The personnel have access to the fabrication process and tools. They can insert a Trojan into the IC by modifying the dopant level or the

mask layout provided to them either during the sample or mass production.

Becker *et al.* [26] design an always-on Trojan in the sub-transistor level by modifying the circuit type. This is done by modifying the polarity of the dopants, so that the transistors always have a constant output and thus modify the functionality. This modification does not require any additional gates. As a proof-of-concept, they stimulate this Trojan in an Intel Ivy Bridge RNG. They observe that their Trojan infected circuit passes the built-in-self-tests. They also find that, by having an entropy of at least 32 bit, the Trojan RNG even passes the National Institute of Standards and Technology (NIST) test suite. This is because the RNG uses AES, which is a strong post-processing scheme. This kind of Trojan may even go unnoticed during traditional RE. Sophisticated and expensive equipment would be required to detect such a Trojan.

3.5 Difficulty of Trojan insertion

In Table 2, we derive the difficulty of Trojan insertion by adversaries at each stage of the IC development. The difficulty of Trojan insertion is derived based on the cost, time and design knowledge needed by an adversary. Cost refers to the manpower, budget and tools required by an adversary. Time constraints refer to the time an adversary has between stages to insert a Trojan. Design knowledge is the insight the adversary would need to have about the design to insert a stealthy Trojan.

In our opinion, Trojan insertion in the functional design stage is relatively easy and has a low cost. The adversary has more freedom, as he/she is not limited by strict time constrains and there is no need to perform complex RE. Manipulations at this level will be carried over to the next stages. The adversary does not need to have a strong technical background or a high insight into the design. Thus, there is low difficulty with Trojan insertion.

Trojan insertion during the physical design stage has a strong impact at a medium cost. However, the attacker would need to have a medium understanding of the functional design and tools. The adversary is not bound by strict time constraints. Thus, this poses as a medium difficulty for Trojan insertion.

Inserting Trojans in the test program generation can greatly impact the integrity of the device. These manipulations can be carried out at a relatively low cost as it does not require many resources. This stage is bound by low time constraints, but the adversary should have high knowledge of the design. Hence, insertions at this stage pose as a medium difficulty.

Trojan insertion during fabrication is most difficult and most expensive as the adversary has very limited information about the design. To insert a complex Trojan, the adversary is required to perform RE of the mask layout in very limited time. In addition, the mask creation is the most expensive part of IC production. Masks can cost anything between 20 K and 2 M US dollars for small feature sizes. This makes it harder for organisations with a limited budget to insert a Trojan during fabrication. However, it is still possible for organisations with big budgets and manpower. It would also be necessary that the attacker has a high design knowledge so that he/she understands the design well and is able to successfully insert a Trojan. Even small mistakes could lead to the failure of the design. In summary, the difficulty of Trojan insertion at this stage is high.

Table 2 Difficulty of Trojan insertion

Development stages	Cost	Time constraints	Design knowledge	**Insertion difficulties**
functional design				
specifications	low	low	low	**low**
architecture	low	low	low	**low**
RTL	low	low	medium	**low**
physical design				
synthesis and P&R	medium	low	medium	**medium**
tools	medium	low	medium	**medium**
mask	high	high	high	**high**
test program development				
test development	low	low	high	**medium**
fabrication				
fabrication	high	high	high	**high**

4 Hardware Trojan detection

In Section 3, we looked the different Trojan types at each stage of IC development (Trojan Insertion branch of Fig. 1). Now we try to identify how Trojans can be detected at different stages (Detection Techniques branch of Fig. 1). This does not include organisational measures which can be applied in order to avoid Trojan insertion. We also provide a survey of previously proposed detection techniques and this is summarised in Table 3.

4.1 Functional design

Trojan-free designs can be obtained only if the specification is established as the root-of-trust. With this, a chain-of-trust must be built in all the following development stages. Malicious inclusions or modifications at this phase can be detected by thorough review. Additionally, the architecture level can be verified using formal verification. Formal verification of the architecture is possible only if the specification is formally defined. Detection at the RTL level can be done using formal verification techniques provided the architecture is formally defined. Simulation of the RTL can be used for evaluating code coverage, that is, to check if all parts of the design have been tested or not.

4.1.1 Formal verification: Banga and Hsiao [27] propose a technique to detect Trojans in third party RTL using automatic test pattern generation (ATPG) tools and equivalence checking. They first eliminate easy-to-detect signals and then use ATPG tools to detect the hard-to-excite signals, since Trojans are known to have rare triggers [28, 29]. Finally, the suspicious signals are compared against the specification provided by the IP provider using equivalence checking to determine if there are any malicious inclusions. Zhang and Tehranipoor [30] propose to use line, toggle, statement and finite state machine coverage algorithms to detect parts of code that have not been executed. The RTL is then synthesised and redundant code is removed to reduce the number of suspicious signals using formal verification and ATPG tools. They also propose to insert scan chains in the netlist and ATPG patterns to check for untestable stuck-at faults. Later, the gates with stuck-at faults are removed from the original design. If any

doi: 10.1049/iet-cdt.2014.0039

Table 3 Survey of Trojan published detection techniques

Papers	Detection stages	Detection techniques	Reference models		
			Golden IC	Design simulation	Specification
Banga and Hsiao [27]	functional design	formal verification			•
Zhang and Tehranipoor [30]	functional design	formal verification		•	
Waksman *et al.* [32]	physical design	functional analysis		•	
Bhasin *et al.* [25]	fabrication	optical inspection		•	
Jha and Jha [33]	fabrication	functional analysis	•		
Chakraborty *et al.* [34]	fabrication	functional analysis		•	
Agrawal *et al.* [10]	fabrication	SCA-power	•		
Banga *et al.* [35]	fabrication	SCA-power	•		
Banga and Hsiao [36]	fabrication	SCA-power	•		
Potkonjak *et al.* [39]	fabrication	SCA-power		•	
Alkabani and Koushanfar [40]	fabrication	SCA-current		•	
Wang *et al.* [37]	fabrication	SCA-current	•		
Rad *et al.* [38]	fabrication	SCA-current	•		
Jin and Makris [18]	fabrication	SCA-delay	•		
Li and Lach [42]	fabrication	SCA-delay		•	
Cha and Gupta [41]	fabrication	SCA-delay	•		
Li *et al.* [43]	fabrication	SCA-delay		•	
Salmani *et al.* [44]	fabrication	structural modification	•		
Salmani and Tehranipoor [45]	fabrication	structural modification	•		
Zhang and Tehranipoor [46]	fabrication	structural modification	•		
Narasimhan *et al.* [47]	fabrication	structural modification		•	

suspicious signal still remains in the design, it is considered to be malicious.

4.2 Physical design

Trojan detection techniques that can be deployed in this phase are review, formal verification and functional analysis of the design. Formal verification techniques can be used to verify the results after synthesis and P&R. After the mask generation process, the design should be extracted and RE to guarantee that no malicious component has been included or malicious modifications of the design have not occurred during the mask generation. EDA tools must be RE to check its correctness.

4.2.1 Functional analysis: Potkonjak [31] developed a technique to design trusted IC using untrusted EDA tools. This is achieved by having no unused resources in the design. For example, using all the functional units in all the control steps and not having any dont-care states in the finite state machine.

Waksman *et al.* [32] propose to use boolean functional analysis to detect nearly unused circuit in third party IP. For this, they construct truth tables for all output wires with its corresponding input wires. Wires with probability below the control value are flagged as suspicious. The control value is the fraction of the rows of the truth table that affects a column in the truth table, that is, the switching probability of the output. Once, the wires are flagged as suspicious, code review is done to determine whether the suspicious wires are malicious.

4.3 Test program development

The next stage of the IC development is the test pattern development. Authentication and integrity checks can be performed inside a chip when the test programmes are downloaded. Then, only authorised parties can test the chip. However, this requires cryptographic modules to be present in the tested chip. All memory contents and fuse bits that have been programmed in the test program can be read out from the sample devices. However, this will not be possible, if the test program has switched the chip into the operating mode which does not permit reading all memory contents.

4.4 Fabrication

The different analysis methods used for hardware Trojan detection in the fabrication stage are optical inspection, functional analysis, SCA and memory readout.

4.4.1 Optical inspection: Chips are reverse engineered; each layer of the IC is photographed and thoroughly examined for any addition, deletion or modification of the design. This technique offers a high degree of assurance. It is also effective in detecting minor changes like inclusion and deletion of transistors. The main drawback of this method is the cost of initial set up. For effective analysis, one requires dedicated equipment, which costs about USD 500 K [http://www.siliconfareast.com/]. Another drawback is that all the chips need to be destructively tested, thus rendering them useless. The whole process is also very time consuming. RE even simple circuits like sensors can cost about USD 15 000 [http://www.chipworks.com/]. Thorough optical inspection should be performed on all the sample devices to check for malicious manipulations. Agrawal *et al.* [10] suggested to use RE on a random set of devices to derive the golden chip. Bhasin *et al.* [25] also proposed to perform optical inspection for Trojan detection using medium cost hardware resources. The authors use optical imaging followed by normalised cross-correlation to compare the GDSII layout and the post-fabrication image of the IC, that is, layout against image comparison.

4.4.2 Functional analysis: Functional analysis is a conventional post-fabrication IC testing tool. It has a low cost but the tests are dedicated to verify the proper functionality of the IC. Trojans are designed to be stealthy and evade the conventional test phase [29]. Hence, these tools are not effective against Trojan detection.

Jha and Jha [33] perform functional analysis of a device under test (DUT). First, a unique probabilistic signature of the circuit is constructed. Once the fingerprint is obtained, a random set of input patterns are applied to the DUT. The DUT is treated as a black box. The probability distribution of the output is unique for each functionally different circuit. If the probability of logic 1 at an output is different for the DUT, then it indicates the presence of a Trojan. When a Trojan is detected, the fingerprint of the input pattern is returned. Such a set up works best on Trojans that modify functionality. In this method, only the implemented circuit is tested for the presence of a Trojan. A Trojan could still be inserted in an unused part of the IC.

Chakraborty *et al.* [34] perform logic testing to detect Trojans in IC that may have been inserted at the time of fabrication. They include an additional transparent mode of operation that triggers rare occurring events and generates a signature for the circuit. They add additional input/output ports, then a user defined key is given as an input and the signature is observed at the primary outputs. Using this technique, they hope that the Trojan will be triggered by the key and causing the output signature to change. Such a technique works for small circuits, as generating multiple signatures for a complex design would be difficult.

4.4.3 Side-channel analysis: SCA is the measurement of some physical property, like power, current and delay, along with some statistical analysis. SCA has a low initial cost and is effective. The main challenge with this technique is the noise caused by process and environmental variations. In the case of conditional Trojans, only the trigger circuit is active at all times. The trigger circuit can be very small and composed of very few gates, which may go unnoticed by conventional SCA because of the high noise levels. Hence, detecting those Trojans is most effective when the trigger circuit is not smaller than 3–4 orders of magnitude of the entire circuit [10]. A side-channel fingerprint can be obtained from the sample devices, which can later be used for comparison with the chips after fabrication.

SCA was first used for Trojan detection by Agrawal *et al.* [10]. They use power analysis to detect Trojans. First, a power fingerprint is generated from a set of randomly selected chips. Later, these chips are destructively tested, that is, reverse engineered, to make sure that they are Trojan-free. All the other chips are compared against this fingerprint. Results are simulated with different percentages of process variations. However, they show that small Trojans could not be detected successfully because of the process variations.

Banga *et al.* [35] also use power profiling for hardware Trojan detection. They first divide the circuit into partitions and then measure the power consumption in each partition. The activity in one partition is increased while reducing the activity of the remaining, in order to maximise the influence of the Trojan on the power consumption. Thus, the impact of the process variations is reduced. They generate test vectors in a way to minimise the hamming distance between adjacent regions.

Banga *et al.* [36] use gate logic inversion to activate Trojans. This is done by inverting the voltage supply of alternate levels so as to cause switching in rarely activated gates. For this, they first simulate a circuit without any voltage inversion, next they invert the voltage of odd levels and finally they invert the voltage of even level gates. They then go on to use power profiling to detect any suspicious behaviour in the circuit.

Wang *et al.* [37] show how Trojans can be detected using a multi-supply transient current integration technique. It is done by measuring the current on multiple ports of the chips. Random set of input patterns are applied to increase the switching activity. The current consumption from each port is integrated and compared with the worst-case charges of the golden chip. The authors assume that the Trojan is inserted only in a select number of dies. The golden chip is obtained by exhaustive testing of several randomly chosen dies. Later, Rad *et al.* [38] divide the chips into regions so as to minimise the current leakage and reduce the impact of process variation. A region is defined as the portion that receives maximum current from the surrounding power ports.

Potkonjak *et al.* [39] develop a technique to detect Trojans by extracting the gate level characteristics like delay and power. They use the measurements to solve a system of linear equations. The aim is to find the leakage scaling factors of for each gate. They simulate the model and measure the input arrival time for each input vector. They manipulate the constants to maximise the matrix rank and then compare it against the statistical model taking into consideration the measurement errors. Alkabani and Koushanfar [40] also propose a technique to measure the gate level characteristics of a circuit. They measure the static current leakage on the output port and compare it against the nominal leakage values of the gate. The nominal values are obtained by simulating the netlist with high coverage input vectors. They simulate a model of this technique and use a consistency metric to measure the difference between the real and estimated scaling factors of the gate leakage.

Measurement of path delay is also a widely proposed technique for Trojan detection. Jin and Makris [18] use path delay fingerprinting for Trojan detection. First, a set of high coverage input patterns are applied to collect the path delay fingerprint of the golden chip. Later, the path delay of the DUT is compared against this fingerprint. Cha and Gupta [41] improve this technique by measuring the path delay along the shortest path. The advantage of this technique is that there is a higher influence of the Trojan on the delay. And thus, the effects of the nominal delays and process variations are minimised. Li and Lach [42] develop a technique to measure the register-to-register path delays. For this they use a physical unclonable function like construction on the functional paths of the core circuit to measure the delay characteristics of the circuit. A negatively skewed shadow register is placed at the end of a number of combinatorial paths. The output of a combinatorial circuit is given to the destination register and a shadow register. These two registers are fed with two different clocks and the outputs from these two registers are compared to detect the presence of Trojans. The main drawback with this technique is the overhead in area. Li *et al.* [43] insert an array of sensors during design phase to measure the on-chip path delay. They correlate the predicted path delay with the actual path delay. A low correlation indicates the presence of a Trojan.

4.4.4 Structural modification: Structural modification refers to the changes in the original circuit at design time by inserting additional circuitry or reordering existing circuitry. This is done in order to enhance Trojan detection during fabrication by other detection techniques, like power, delay or logical testing.

A skillful adversary would design a Trojan to have a rare trigger so that they are not detected during the regular IC

doi: 10.1049/iet-cdt.2014.0039

testing [28, 29]. Hence, by eliminating low probability paths, the probability of Trojan activation can be increased. This also increases the probability of detection using other detection schemes like SCA, as the switching activity of the Trojan circuitry is increased. To increase the probability of rare occurring paths, Salmani *et al.* [44] propose to insert dummy scan flip-flops in paths with low transition probabilities. By inserting dummy scan flip-flops, one could improve the transition probabilities, thus also increasing the probability of rare occurring events during IC testing.

Salmani *et al.* [45] propose another technique by reordering scan cell chains. Owing to the complexity of current ICs, scan cells are generally used for IC testing to verify the internal circuitry. The authors used the scan cells to detect abnormal switching activity in the IC. They reordered the scan cells with respect to their position in the physical layout, thereby restricting the activity to a particular region. The layout is first divided into smaller regions and scan cells are reordered accordingly. By limiting the switching activity to one region, impact of the Trojan increases.

In another approach, Zhang and Tehranipoor [46] inserted an RO network. The ROs were distributed all over the IC to detect abnormal switching activity. A golden IC is used to generate a fingerprint of the RO frequencies and then the DUT is compared against this fingerprint. Switching because of Trojan activity affects the frequency of the RO network. As a result, the localised effects of the Trojan are captured from the closest RO.

Narasimhan *et al.* [47] suggest to distribute a set of current sensors in the design to detect Trojans. They use the current sensors to measure the on-chip transient current. This is done by applying a set of input vectors on the same die at several intervals of time. They then compare the transient supply current to detect malicious behaviour. The model was simulated and the results show better sensitivity of on-chip current measurement against off-chip measurements. However, this comes at the cost of area and performance. Furthermore, the authors also take the tampering of the current sensors into consideration.

Chakraborty and Bhunia [48] proposed to use design obfuscation to increase the complexity of RE for the attacker. They do this by adding additional states to design. Such a technique increases the complexity of the attacker to insert an ingenious Trojan in the design. However, this involves adding additional circuitry and thus reducing the performance of the whole design.

4.4.5 Memory readout: Since detection techniques like optical inspection and SCA might not be feasible for testing every fabricated IC. One could read out a unique hash from each IC after the mass production, which can then be verified by the semiconductor company.

4.5 Difficulty of Trojan detection

From our analysis, we see that current detection schemes publicly known mostly focus on post-fabrication detection techniques. Table 3 shows that most of the techniques use a golden chip as their reference model. A golden chip is an IC that is assumed to be Trojan-free. Detection techniques like optical inspection or exhaustive testing could be used to derive the golden chip. Comparison of the DUT against a golden chip is most reliable, but this is based on the assumption that the Trojan is not inserted in all the fabricated chips. Another reference model that can be used is a design simulation. Assuming to have a trusted specification and design phase, one can use the simulation of a pre-fabrication design for comparison. The behaviour of the simulated design and DUT might be significantly different because of the process and environmental variations.

Table 4 Difficulty of Trojan detection

Development stages	Costs	Time required	Technique accuracies	**Detection difficulties**
functional design				
specification	low	low	medium	**low**
architecture	low	low	medium	**low**
RTL	low	medium	high	**low**
physical design				
synthesis and P&R	medium	medium	high	**medium**
tools	high	high	medium	**high**
mask	high	high	high	**high**
test program development				
test development	medium	medium	high	**medium**
fabrication				
fabrication	high	high	high	**high**

In Table 4, we derive the difficulty of Trojan detection for each IC development stage based on the cost, time and accuracy of Trojan detection using a particular detection technique. Cost is the budget and tools required to set up the detection technique. Time required is the time needed for Trojan detection and finally accuracy defines the reliability of the technique. Whenever there are several detection techniques, we use the one with the highest accuracy for our evaluation. Through this evaluation, we try to determine the practicability of Trojan detection at each stage of development.

Trojan detection during the functional design stage can be carried out at a relatively low cost and time. Thorough review of the specification and architecture provides a medium accuracy for Trojan detection. Simulation of the RTL has a high accuracy of Trojan detection. Thus, in our opinion the difficulty of Trojan detection at this stage is low.

During the physical design stage, an evaluator would require higher resources and more time for Trojan detection. As in this stage, they might have to reverse engineer complex layouts and tools. Techniques like formal analysis and RE provide a high accuracy, but are expensive and time consuming. Hence, we conclude that there is a medium to high difficulty of Trojan detection.

Trojan detection at the test program development stage has a medium cost and time because of the complexity of the design. Techniques like authentication and integrity checks provide a high accuracy level. Thus, there is a low-to-medium difficulty of Trojan detection at this stage.

Current research mostly focuses on Trojan detection during fabrication. The most accurate technique here is RE, which is, however very expensive and requires a lot of time in order to perform a thorough analysis. However, this still does not guarantee the integrity of all the other fabricated ICs. For instance, detecting changes in the dopant level can be very expensive, as one needs sophisticated tools to conduct RE [26]. Thus, this stage has a high difficulty with regard to Trojan detection.

4.6 Trojan insertion against Trojan detection

In this section, we analyse the difficulty of Trojan insertion at each stage against the difficulty of Trojan detection. The cost and effort of Trojan insertion increases with each design step. Similarly, the cost of Trojan detection increases as the complexity of the design increases with each development stage. Also with the increase in complexity, the time required for reliable Trojan detection also increases and thus the difficulty of the Trojan detection also increases.

From our analysis in Tables 2 and 4, we see that the difficulty of Trojan insertion during the functional design and test program development stage is low and also the cost for Trojan detection is low. Furthermore, the difficulty of Trojan detection during the functional design stage is also low and a medium difficulty during the test program development stage.

On the other hand, the difficulty of Trojan insertion and detection is high during the fabrication stage. As here an adversary would need high budget and high design knowledge to insert a Trojan. Trojan detection after fabrication can be expensive and time consuming as each fabricated device would need to be thoroughly tested.

Looking back at Fig. 1, as we move from the origin outwards, the cost of both Trojan insertion and Trojan detection increases with every circle, that is, with every design stage. We identify that it is easier for an adversary to make malicious manipulation during the early design phases than during fabrication. And it is also easier for the evaluators to detect Trojans during the early design stages than post fabrication. Hence, we conclude that future research must focus more on securing the early stages of development against Trojan insertion. In addition, more effort should be spent on developing better detection techniques and especially during the test program development.

5 Conclusion

Through the course of this paper, we show that the threat of hardware Trojans is not just during fabrication, but the entire IC design chain is at risk. We discuss the threat of hardware Trojan insertion at different stages of IC development and the factors influencing it. We also provide a survey of both hardware Trojan design and detection techniques. Furthermore, we discuss the strengths and challenges of Trojan insertion and Trojan detection techniques at each development stage. We conclude that Trojans are relatively easier to insert during the functional design phase as compared with the fabrication stages. In addition, Trojan detection and prevention techniques are cheaper during the functional design phase than during the fabrication phase. From our assessment, we see that current research is mostly focusing on the detection techniques during the fabrication. This might not be very effective as Trojans inserted during the design phase will be difficult to detect by only using post-fabrication detection techniques. Work has already begun in this direction but this is not sufficient. Research should focus more on this stage, especially in regard with the secure integration of third party IP. As another result, we also show that the test development phase is also vulnerable to Trojan insertion and steps must be taken to prevent this. Future research must concentrate on developing efficient detection techniques for every stage of the IC development and production, especially for those in the design and test development phases.

6 References

1 Appelbaum, J., Horchert, J., Stöcker: 'Shopping for spy gear: catalog advertises NSA toolbox'. Der Spiegel, 2013
2 Ball, J., Schneier, B.: 'Explaining the latest NSA revelations Q&A with internet privacy experts'. The Guardian, 2013
3 Gellman, B., Nakashima, E.: 'U.S. spy agencies mounted 231 offensive cyber-operations in 2011', documents show, 2013
4 Perlroth, N., Larson, J., Shane, S.: 'NSA able to foil basic safeguards of privacy on web' (New York, New York Times, 2013)
5 Shiyanovskii, Y., Wolff, F.G., Rajendran, A., Papachristou, C.A., Weyer, D.J., Clay, W.: 'Process reliability based Trojans through NBTI and HCI effects', *2010 NASA/ESA Conference on Adaptive Hardware and System (AHS)*, June 15–18 2010, Anaheim, CA (USA), (IEEE, 2010), pp. 215–222
6 Skorobogatov, S., Woods, C.: 'Breakthrough silicon scanning discovers backdoor in military chip'. Cryptographic hardware and embedded systems CHES 2012, Berlin Heidelberg, 2012 (*LNCS*, **7428**), pp. 23–40
7 Adee, S.: 'The hunt for the kill switch', *IEEE Spectr.*, 2008, **45**, (5), pp. 34–39
8 D. S. B. T. Force: 'Report of defense science board task force on high performance microchip supply', 2005
9 Beaumont, M., Hopkins, B., Newby, T.: 'Hardware Trojans prevention, detection, countermeasures (a literature review)'. Defence Science and Technology Organisation, Command, Control, Communication and Intelligence Division, 2011
10 Agrawal, D., Baktir, S., Karakoyunlu, D., Rohatgi, P., Sunar, B.: 'Trojan detection using IC fingerprinting'. IEEE Symp. on Security and Privacy, IEEE Computer Society, 2007, pp. 296–310
11 Tehranipoor, M., Koushanfar, F.: 'A survey of hardware Trojan taxonomy and detection', *IEEE Des. Test Comput.*, 2010, **27**, (1), pp. 10–25
12 Europe Smart Card Industry Association: 'Security IC Platform Protection Profile with Augmentation Packages', 2014
13 Tehranipoor, M., Salmani, H., Zhang, X., *et al.*: 'Trustworthy hardware: Trojan detection and design-for-trust challenges', *IEEE Comput.*, 2011, **44**, (7), pp. 66–74
14 Wang, X., Tehranipoor, M., Plusquellic, J.: 'Detecting malicious inclusions in secure hardware: challenges and solutions'. Proc. of the 2008 IEEE Int. Workshop on Hardware-Oriented Security and Trust, HST'08, IEEE Computer Society, Washington, DC, USA, 2008, pp. 15–19
15 Ball, J., Borger, J., Greenwald, G.: 'Revealed: how US and UK spy agencies defeat internet privacy and security challenges'. The Guardian, 2013
16 ARS Technica: 'Stop using NSA-influenced code in our products, RSA tells customers'. Available at http://www.arstechnica.com/security/2013/09/stop-using-nsa-influence-code-in-our-product-rsa-tells-customers/, 2013
17 Schneier, B.: 'Elliptic curve crypto primer'. Available at https://www.schneier.com/blog/archives/2013/11/elliptic_curve.html, 2013
18 Jin, Y., Makris, Y.: 'Hardware Trojan detection using path delay fingerprint'. IEEE Int. Workshop on Hardware-Oriented Security and Trust, HOST 2008, Proc., Anaheim, CA, USA, 9 June 2008, pp. 51–57
19 Chakraborty, R., Saha, I., Palchaudhuri, A., Naik, G.: 'Hardware Trojan insertion by direct modification of FPGA configuration bitstream', *IEEE Des. Test*', 2013, **30**, (2), pp. 45–54
20 King, S.T., Tucek, J., Cozzie, A., Grier, C., Jiang, W., Zhou, Y.: 'Designing and implementing malicious hardware'. Networked Systems Design and Implementation, 2008
21 Lin, L., Kasper, M., Güneysu, T., Paar, C., Burleson, W.: 'Trojan side-channels: lightweight hardware Trojans through side-channel engineering'. CHES, 2009, (*LNCS*, **5747**), pp. 382–395
22 Jin, Y., Makris, Y.: 'Hardware Trojans in wireless cryptographic ICs', *IEEE Des. Test Comput.*, 2010, **27**, (1), pp. 26–35
23 Kutzner, S., Poschmann, A., Stöttinger, M.: 'Hardware Trojan design and detection – a practical evaluation'. Eighth Workshop on Embedded Systems Security, 2013
24 Muehlberghuber, M., Gurkaynak, F.K., Korak, T., Dunst, P., Hutter, M.: 'Red team vs. blue team hardware Trojan analysis: detection of a hardware Trojan on an actual ASIC'. Hardware and Architectural Support for Security and Privacy – HASP 2013, Second Workshop, Proc., Tel-Aviv, Israel, 23 June 2013, pp. 1–8
25 Bhasin, S., Danger, J.-L., Guilley, S., Ngo, X.T., Sauvage, L.: 'Hardware Trojan horses in cryptographic IP cores', *2013 Workshop*

doi: 10.1049/iet-cdt.2014.0039

on Fault Diagnosis and Tolerance in Cryptography, August 20 2013, Los Alamitos, CA (USA), (IEEE, 2013), pp. 15–29.
26 Becker, G.T., Regazzoni, F., Paar, C., Burleson, W.P.: 'Stealthy dopant-level hardware Trojans'. CHES, 2013, (*LNCS*, **8086**), pp. 197–214
27 Banga, M., Hsiao, M.S.: 'Trusted RTL: Trojan detection methodology in presilicon designs'. HOST, 2010, pp. 56–59
28 DARPA BAA06-40: 'TRUST for integrated circuits'. Defense Advanced Research Projects Agency, 2006
29 Wolff, F., Papachristou, C., Bhunia, S., Chakraborty, R.S.: 'Towards Trojan free trusted ICs: problem analysis and detection scheme'. Proc. of the Conf. on Design, Automation and Test in Europe, DATE '08, New York, NY, USA, 2008, pp. 1362–1365
30 Zhang, X., Tehranipoor, M.: 'Case study: detecting hardware Trojans in third party digital IP cores'. HOST, 2011, pp. 67–70
31 Potkonjak, M.: 'Synthesis of trustable ICs using untrusted CAD tools'. DAC, 2010, pp. 633–634
32 Waksman, A., Suozzo, M., Sethumadhavan, S.: 'FANCI: identification of stealthy malicious logic using boolean functional analysis'. ACM Conf. on Computer and Communications Security, 2013, pp. 697–708
33 Jha, S., Jha, S.K.: 'Randomization based probabilistic approach to detect Trojan circuits'. Proc. of the 2008 11th IEEE High Assurance Systems Engineering Symp., HASE'08, IEEE Computer Society, Washington, DC, USA, 2008, pp. 117–124
34 Chakraborty, R.S., Paul, S., Bhunia, S.: 'On-demand transparency for improving hardware Trojan detectability'. HOST, 2008, pp. 48–50
35 Banga, M., Chandrasekar, M., Fang, L., Hsiao, M.S.: 'Guided test generation for isolation and detection of embedded Trojans in ICs'. Proc. of the 18th ACM Great Lakes Symp. on VLSI, GLSVLSI'08, New York, NY, USA, 2008, pp. 363–366
36 Banga, M., Hsiao, M.S.: 'VITAMIN: voltage inversion technique to ascertain malicious insertions in ICs'. HOST, 2009, pp. 104–107
37 Wang, X., Salmani, H., Tehranipoor, M., Plusquellic, J.: 'Hardware Trojan detection and isolation using current integration and localized current analysis'. Proc. of the 2008 IEEE Int. Symp. on Defect and Fault Tolerance of VLSI Systems, DFT'08, IEEE Computer Society, Washington, DC, USA, 2008, pp. 87–95
38 Rad, R.M., Wang, X., Tehranipoor, M., Plusquellic, J.: 'Power supply signal calibration techniques for improving detection resolution to hardware Tatrojans'. ICCAD, 2008, pp. 632–639
39 Potkonjak, M., Nahapetian, A., Nelson, M., Massey, T.: 'Hardware Trojan horse detection using gate-level characterization'. DAC, 2009, pp. 688–693
40 Alkabani, Y., Koushanfar, F.: 'Consistency-based characterization for IC Trojan detection'. Proc. of the 2009 Int. Conf. on Computer-Aided Design, ICCAD '09, New York, NY, USA, 2009, pp. 123–127
41 Cha, B., Gupta, S.K.: 'Trojan detection via delay measurements: a new approach to select paths and vectors to maximize effectiveness and minimize cost', in Macii, E. (Ed.): 'DATE' (EDA Consortium/ACM DL, San Jose, CA, USA, 2013), pp. 1265–1270
42 Li, J., Lach, J.: 'At-speed delay characterization for IC authentication and Trojan horse detection'. HOST, 2008, pp. 8–14
43 Li, M., Davoodi, A., Tehranipoor, M.: 'A sensor-assisted self-authentication framework for hardware Trojan detection'. Proc. of the Conf. on Design, Automation and Test in Europe, DATE'12, EDA Consortium, San Jose, CA, USA, 2012, pp. 1331–1336
44 Salmani, H., Tehranipoor, M., Plusquellic, J.: 'New design strategy for improving hardware Trojan detection and reducing Trojan activation time'. Proc. of the 2009 IEEE Int. Workshop on Hardware-Oriented Security and Trust, HST'09, IEEE Computer Society, Washington, DC, USA, 2009, pp. 66–73
45 Salmani, H., Tehranipoor, M.: 'Layout-aware switching activity localization to enhance hardware Trojan detection', *IEEE Trans. Inf. Forensics Sec.*, 2012, **7**, (1), pp. 76–87
46 Zhang, X., Tehranipoor, M.: 'RON: an on-chip ring oscillator network for hardware Trojan detection'. DATE, IEEE, 2011, pp. 1638–1643
47 Narasimhan, S., Yueh, W., Wang, X., Mukhopadhyay, S., Bhunia, S.: 'Improving IC security against Trojan attacks through integration of security monitors', *IEEE Des. Test Comput.*, 2012, **29**, (5), pp. 37–46
48 Chakraborty, R.S., Bhunia, S.: 'Security against hardware Trojan through a novel application of design obfuscation'. Proc. of the 2009 Int. Conf. on Computer-Aided Design, ICCAD'09, New York, NY, USA, 2009, pp. 113–116