CHAPTER

ELECTRONICS SUPPLY CHAIN

6

CONTENTS

Hardware Security. https://doi.org/10.1016/B978-0-12-812477-2.00011-3

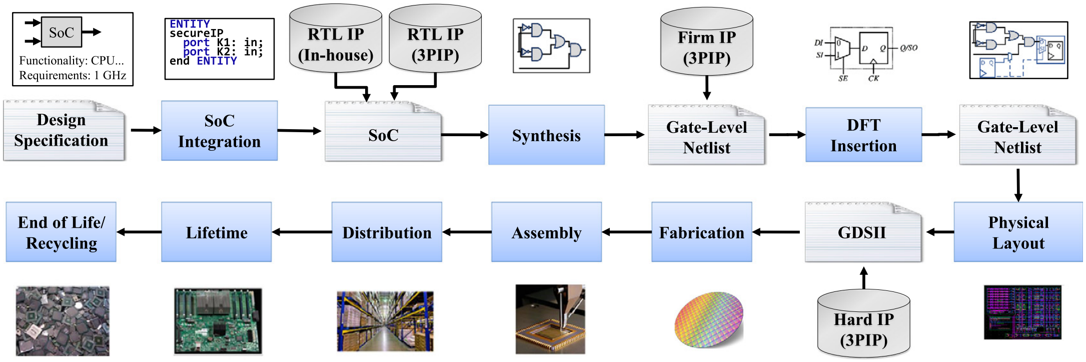
6.1 INTRODUCTION

The trend of transistor scaling has enabled designers to fit an increasing amount of functionality on a single chip. Integrating the overall functionality of a system into a single chip improves the performance (for example, speed and power) while reducing the cost by minimizing the required silicon area. Such a chip is referred to as a system on chip (SoC), and the vast majority of modern mobile and handheld devices contain SoCs, as do many embedded devices. In general, an SoC contains analog components (for example, radio-frequency receiver, analog-to-digital converter, network interfaces), digital components (such as a digital signal processing unit, graphics processing unit, central processing units, and cryptographic engine), and memory elements (for instance, RAM, ROM, and flash) [1,2].

The complexity of designing modern SoCs is amplified by time-to-market pressure, making it infeasible for a single design house to complete an entire SoC without outside support. Additionally, the cost to build and maintain a fabrication facility (commonly known as a foundry or fab) for modern technology nodes is now in the multi-billion dollar range. As a result, the majority of SoC design houses can no longer afford their own fab. Impacted by these factors, the semiconductor industry has shifted to a horizontal business model over the past two decades. In this model, time-to-market and manufacturing costs are lowered through outsourcing and design reuse. To be more specific, SoC design houses obtain licenses for third party intellectual property (3PIPs), design an SoC by integrating the various 3PIPs with their own IP, and outsource the SoC design to contract foundries and assemblies for fabrication and packaging. Although this model has lowered time-to-market and manufacturing costs through outsourcing and design reuse, it has also introduced security and trust issues in the final product. This chapter discusses the composition of modern electronic hardware supply chain, the security and trust issues associated with it, and the potential countermeasures to address these concerns [3].

6.2 MODERN ELECTRONIC SUPPLY CHAIN

Figure 6.1 shows the modern system on chip (SoC) design flow and its corresponding supply chain. The following subsections discuss the flow and supply chain in details.



```
ENTITY
secureIP
   port K1: in;
   port K2: in;
end ENTITY
```

FIGURE 6.1

Supply chain of system on chip (SoC) design.

6.2.1 DESIGN

The design of an SoC includes multiple steps, for example, design specifications, SoC integration, synthesis, insertion of test and debug structures, physical layout generation, and functional and performance verification.

6.2.1.1 Design Specification

In the first step, the SoC integrator (commonly called design house) specifies the high-level requirements and blocks of the SoC. For example, the SoC integrator first identifies what functionalities need to be incorporated in the SoC and what the targeted performance will be. It then identifies a list of functional blocks to implement the SoC. These functional blocks have intellectual property (IP) values and are commonly referred to as IPs. These IP cores are either developed in-house or purchased from 3PIP developers. This decision is mainly driven by economic factors. For example, if an SoC integrator decides to incorporate a GPU unit in the SoC, then he/she could direct his/her hardware designers to develop the GPU unit. However, often it is more economically feasible to procure this IP from third-party vendors who specialize in designing GPUs.

6.2.1.2 3PIP Acquisition

The third party intellectual property cores can be procured in the following three forms:

- Soft IP cores are delivered as synthesizable register transfer level (RTL) code written in hardware description language (HDL), for example, Verilog or VHDL. The soft IP cores are similar to a high-level programming code, such as C, with the difference being they are developed for hardware implementation. Most IPs are procured as soft IPs as they offer more flexibility.
- Firm IP cores are delivered as gate-level implementation of the IP, possibly using a generic library. The firm IP cores are synthesized from the RTL code, and represented as a netlist consisting of logic gates and wires. Unlike soft IP cores, a firm IP does not possess the behavioral information of the IP. Therefore, firm IPs offer less flexibility as compared with soft IPs.

- Hard IP cores are delivered as GDSII representations of a fully placed and routed design. The hard IPs are integrated at the last stages of the design process. They offer least flexibility, but at a lower cost. For example, most memory IPs are procured as hard IPs.

6.2.1.3 SoC Integration

After developing/procuring all the necessary soft IPs, the SoC design house integrates them to generate the RTL specification of the whole SoC. The RTL design goes through extensive functional testing to verify the functional correctness of the SoC and find any design bugs.

6.2.1.4 Synthesis

The SoC integrator synthesizes the RTL description into a gate-level netlist based on a target technology library. Synthesis is a process by which an RTL code is transformed into a hardware implementation consisting of logic gates. Synthesis process is performed by computer-aided design (CAD) tools, for example, Design Compiler from Synopsys. The CAD tools also optimize the design with the objective of minimizing area, timing, or power. The gate-level netlist then goes through formal equivalence checking to verify that the netlist is equivalent to the RTL representation. The SoC designers may also integrate a firm IP core from a vendor into the SoC netlist at this stage.

6.2.1.5 DFT Insertion

Design-for-test (DFT) refers to the addition of test infrastructure together with the use of test algorithms to generate effective tests to improve testability of an SoC. Higher testability leads to improved test coverage, test quality, and lower test costs. DFT enables the IC to be thoroughly tested during fabrication, package assembly, and in the field to ensure its correct functionality. To achieve these objectives, the SoC integrator integrates the DFT structure into the SoC. However, in many cases, the DFT insertion is outsourced to third party vendors who specialize in designing test and debug structures (e.g., scan, built-in self-test (BIST), and compression structures).

6.2.1.6 Physical Layout

In this step, the gate-level netlist is translated into a physical layout design. Here, each gate is translated into its transistor level layout. Physical layout also performs transistor placement and wire routing as well as clock tree and power grid placement. It is also possible to import hard IP cores from vendors and integrate them into the SoC at this stage. After performing static timing analysis (STA) and power closure, SoC integrator generates the final layout in GDSII format and sends it out to a foundry for fabrication. The generated GDSII file contains layer-by-layer information needed to fabricate the SoC on a silicon wafer.

6.2.2 FABRICATION

As the technology for integrated circuits and SoCs shrinks to very deep sub-micron levels, the complexity and cost of chip fabrication increase significantly. Therefore, only a few companies can afford to maintain state-of-the-art fabrication facilities. Most design houses have become fabless, that is, they fabricate their products by third-party offshore foundries. In this process, the SoC designers enjoy reduced cost and state-of-art fabrication technologies, however, at the cost of reduced control over product integrity and, therefore, reduced trust in the manufacturing process. The foundry also performs

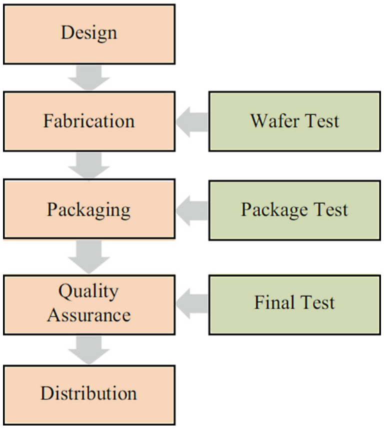


FIGURE 6.2

SoC design and test flow.

structural/functional tests on the die to find manufacturing defects. These defects are caused by imperfections in the fabrication processes. The fraction of defect-free chips produced in a manufacturing process is called *yield*. The faulty chips are discarded, and the good chips are sent to assembly to be packaged.

6.2.3 ASSEMBLY

After fabrication, the foundry sends tested wafers to the assembly line to cut the wafers into several die, and package the good ones to produce chips. Advanced assembly operation also includes wafer/die bumping, die placement, solder reflow, underfill, encapsulation and substrate ball attach. After these processes are done, Assembly performs structural tests to find defects in the chip that could be introduced during the assembly process. Figure 6.2 shows the test process, where the package test is performed at the assembly, followed by the final test for the quality assurance. After performing these tests, the chips without defects are shipped to the distributors, or to the system integrators.

Note that the Wafer Test and the Package Test performed by the foundry and the assembly, respectively, are mostly structural tests, for example, automatic-test-pattern-generation-based (ATPG-based) tests. These tests are performed to find defects in the chip introduced during the fabrication and assembly process. These tests do not necessarily test chip functionality, which ensures the proper functionality of the chips. On the contrary, the Final Test performed during the Quality Assurance process mostly focuses on testing chip functionality.

6.2.4 DISTRIBUTION

The tested ICs are sent either to the distributors or to system integrators. The distributors sell these ICs in the market. These distributors are of several types, including OCM authorized distributors, independent distributors, internet-exclusive suppliers, and brokers.

6.2.5 LIFETIME

The lifetime process starts by combining all the components and subsystems together to produce the final product, for example, a printed circuit board (PCB). This job is typically outsourced to a third-party company, which mounts all the necessary components into one or more PCB to make the final product. Once the final product is assembled, it is sent to the consumer.

6.2.6 END-OF-LIFE

When electronics age or become outdated, they are typically retired and, subsequently, replaced. Proper disposal techniques are highly advised to extract precious metals and prevent hazardous materials, such as lead, chromium, and mercury from harming the environment.

6.3 ELECTRONIC COMPONENTS SUPPLY CHAIN ISSUES

Due to the globalization of the electronics supply chain, many security vulnerabilities can be intentionally or unintentionally introduced by entities involved in the supply chain. Also, with most of these entities involved in the design, manufacturing, integration, and distribution located across the globe, original IP owners and the SoC integrators no longer have the ability to control and monitor the entire process. In other words, trust becomes a major concern in the modern design flow. The IP owners cannot have complete trust in the SoC designers, whereas the SoC designers may not trust IP owners, the foundries, or assemblies [1].

Here, security and trust vulnerabilities in the supply chain are classified into two different classes (shown in Fig. 6.3). Some design issues may cause security vulnerabilities in the integrated circuits and systems, whereas trust issues are mostly associated with factors, such as counterfeiters gaining illegal profit and attackers gaining control of the chip by malicious inclusions.

6.4 SECURITY CONCERNS

This section discusses vulnerabilities maliciously introduced by insertion of hardware Trojans and unintentionally introduced by CAD tools, design mistakes, and test/debug structures.

6.4.1 HARDWARE TROJANS

A hardware Trojan is defined as a malicious modification of a circuit design that results in undesired behavior when the circuit is deployed in the field [4]. Details of hardware Trojan, its structure and potential adversaries who are capable of inserting Trojan are discussed in Chapter 5.

6.4.2 CAD TOOLS

Computer-aided design (CAD) software used to design, test, and validate SOCs can unintentionally introduce vulnerabilities into SoCs [9], because they were not designed with security in mind; instead, their design is driven primarily by conventional metrics such as area, timing, power, yield, and testa-

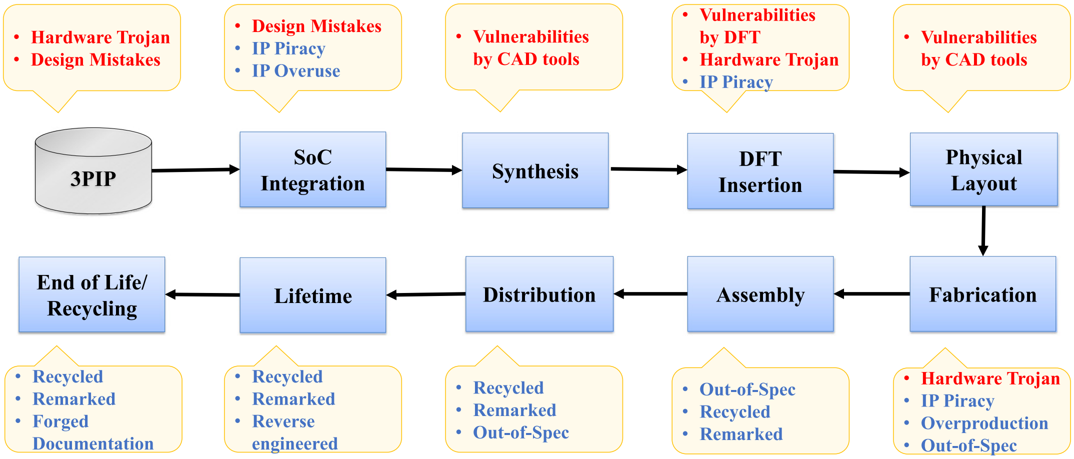


FIGURE 6.3

Vulnerabilities in the hardware supply chain. The red (dark gray in print version) and blue (light gray in print version) colored text represent security and trust issues, respectively.

bility. Designers who overly rely on these tools can, therefore, fall victim to "lazy engineering" [10], where the design is optimized without being aware of impacts on security. This can result in backdoors through which sensitive information can be leaked (that is, violation of a confidentiality policy), or an attacker can gain control of a secured system (violation of integrity policy). For example, finite state machines (FSMs) often contains don't-care conditions in which a transition, next state, or output is not specified. A synthesis tool will optimize the design by replacing don't-care conditions with deterministic states and transitions. A vulnerability will be introduced if a protected state (for example, kernel mode) becomes illegally accessible by the new states/transitions [11].

The controller circuit of an AES encryption module is used as another case study to demonstrate the vulnerability introduced by the CAD tools. The state transition diagram of the FSM shown in Fig. 6.4B implements the AES encryption algorithm on the data path shown in Fig. 6.4A. The FSM is composed of 5 states, and each of these states controls specific modules during the ten rounds of AES encryption. After ten rounds, the "Final Round" state is reached, and the FSM generates the control signal *finished* $= 1$, which stores the result of the "Add Key" module (that is, the ciphertext) in the "Result Register". For this FSM, the Final Round is a protected state, because, if an attacker can gain access to the Final Round without going through the "Do Round" state, then premature results will be stored in Result Register, potentially leaking the secret key. Now, during the synthesis process if a don't-care state is introduced that has direct access to a protected state, then it can create vulnerability in the FSM by allowing the attacker to utilize this don't-care state to access the protected state. Let us consider that the "Don't-care_1" state, shown in Fig. 6.4B, is introduced by the synthesis tool and this state has direct access to the protected state Final Round. Introduction of the Don't-care_1 state represents a vulnerability introduced by the CAD tool because this don't-care state can facilitate fault, and Trojan-based attack. For example, an attacker can inject a fault to go to the Don't-care_1 state, and access the protected state Final Round from this state. The attacker can also utilize the Don't-care_1 to

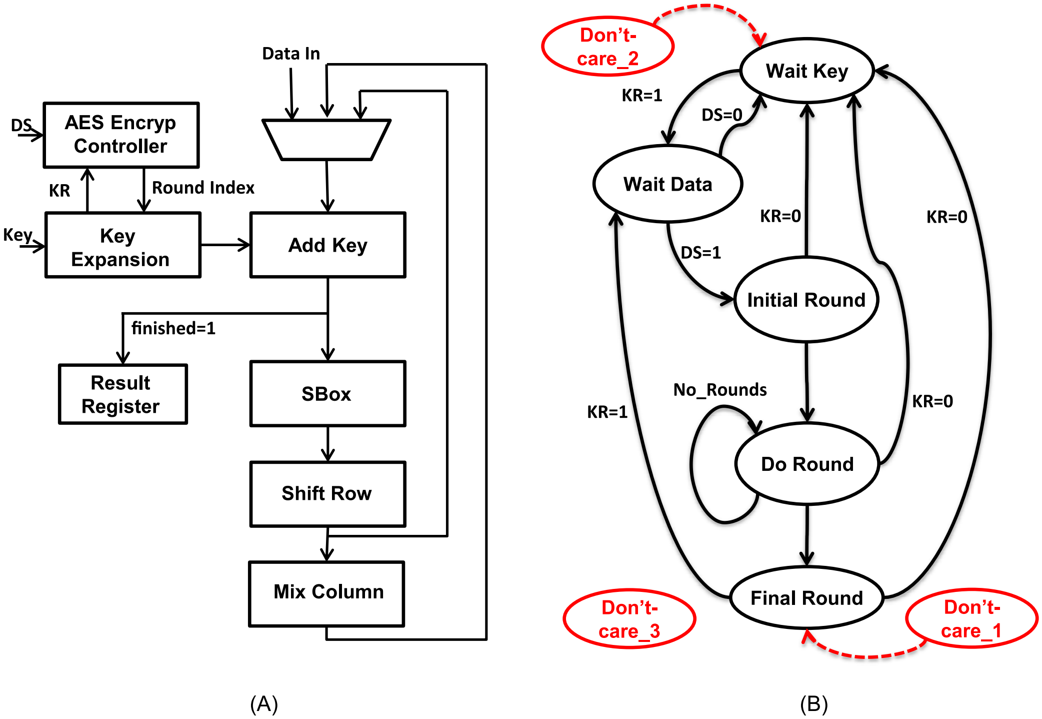


FIGURE 6.4

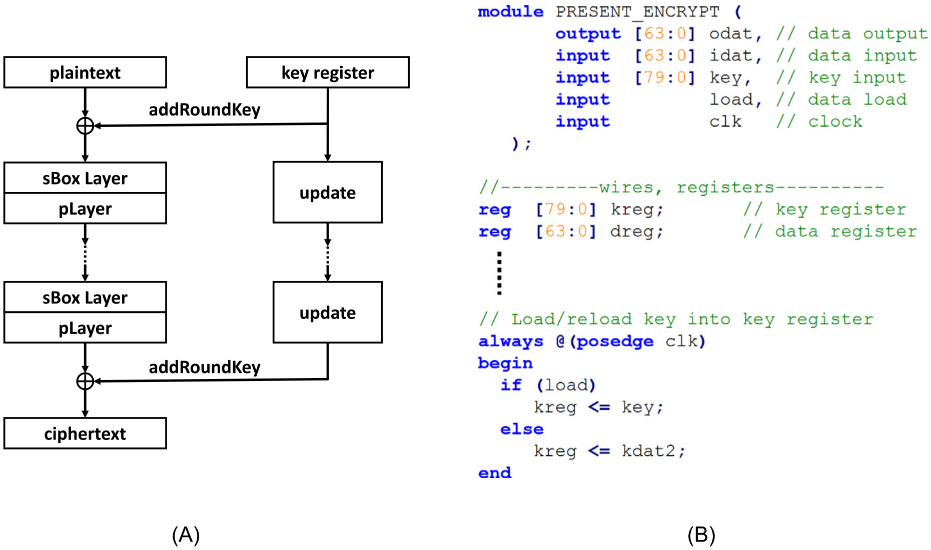
Unintentional vulnerabilities created by CAD tools. (A) and (B) show data path and finite state machine (FSM) of AES encryption module. KR and DS stand for Key Ready and Data Stable signals, respectively; the red (gray in print version) marked states and transitions represent the don't-care states, and transitions introduced by the CAD tool.

implant a Trojan. The presence of this don't-care state gives the attacker a unique advantage because this state is not taken into consideration during validation and testing; therefore, it is easier for the Trojan to evade detection.

Additionally, during the synthesis process, CAD tools flatten all the modules of the design together and try to optimize the design for power, timing, and/or area. If a secure module, such as encryption module is present in an SoC, design flattening and the multiple optimization processes can lead to merging trusted blocks with those untrusted. These design steps, which the designer has little control of, can introduce vulnerabilities and cause information leakage [12].

6.4.3 DESIGN MISTAKES

Traditionally the design objectives are driven by cost, performance, and time-to-market constraints; whereas, security is generally neglected during the design phase. Additionally, security-aware design practices do not yet exist. Thus, many security vulnerabilities can be created unintentionally by de-



```
module PRESENT_ENCRYPT (
       output [63:0] odat, // data output
       input  [63:0] idat, // data input
       input  [79:0] key,  // key input
       input         load, // data load
       input         clk   // clock
   );

//---------wires, registers----------
reg  [79:0] kreg;        // key register
reg  [63:0] dreg;        // data register
  ⋮

// Load/reload key into key register
always @(posedge clk)
begin
  if (load)
     kreg <= key;
  else
     kreg <= kdat2;
end
```

FIGURE 6.5

Unintentional vulnerabilities created by design mistakes. (A) Top-level description of PRESENT, (B) Verilog implementation of PRESENT.

sign mistakes or a designer's lack of understanding of security problems [15]. Design engineers may not have sufficient knowledge in hardware and information security due to the high complexity of the designs and diversity of security problems. For instance, security is often in direct conflict with the intuition that engineers have developed for IC testing. Design-for-test and design-for-debug infrastructures can, themselves, provide backdoors if not properly designed.

This is illustrated further with a case study [15]. Figure 6.5A shows the top-level description of PRESENT encryption algorithm [13]. A segment of its Verilog implementation is shown in Fig. 6.5B. One can see that the key is directly being assigned to the register, defined as "kreg" in the module. Although the encryption algorithm itself is secure, a vulnerability is unintentionally created in its hardware implementation. When this design is implemented, the "kreg" register will be included in the scan chain, and an attacker can gain access to key through scan-chain-based attack [16].

Also, different implementation style of a same algorithm can have different levels of security. In a recent study [17], two AES SBox architectures, PPRM1 [18] and Boyar and Peralta [19], were analyzed to evaluate which design is more susceptible to fault-injection attack. The analysis showed that P-AES is more vulnerable to fault-injection attack than the B-AES architecture.

6.4.4 TEST/DEBUG STRUCTURE

High testability is important for critical systems to ensure proper functionality and reliability throughout their lifetime. Testability is a measure of controllability and observability of signals (that is, nets) in a circuit. Controllability is defined as the difficulty of setting a particular logic signal to "1" or "0", and

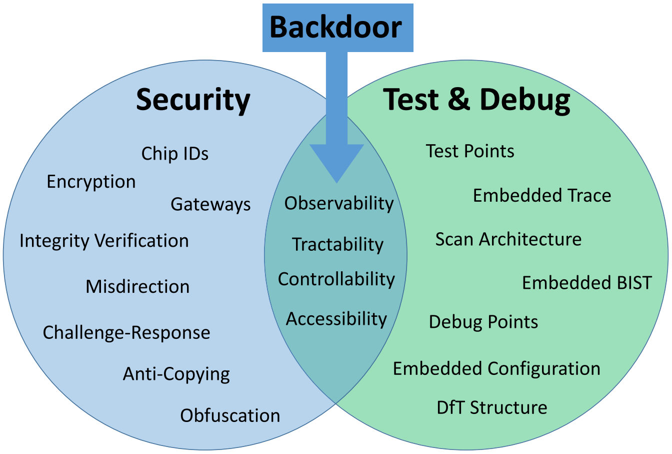


FIGURE 6.6

Requirement for high-quality test and debug contradicts security.

observability is defined as the difficulty of observing the state of a logic signal. To increase testability and debug, it is very common to integrate design-for-test (DFT) and design-for-debug (DFD) structures in a complex design. However, the increased controllability and observability added by DFT and DFD structures can create numerous vulnerabilities by allowing attackers to control or observe internal states of an IC [20].

In general, test and debug can be viewed as the opposite of security when it comes to accessing circuit internals, as shown in Fig. 6.6. Unfortunately, the DFT and DFD structures cannot be simply avoided in modern designs, because of large amount of unexpected defects and errors that occur during the fabrication deep sub-micron devices. Additionally, National Institute of Standards and Technology (NIST) requires that any design used in critical applications needs to be properly testable, both in pre- and post-manufacturing. Therefore, the DFT and DFD structures must be incorporated in ICs, though these structures may create vulnerability. Thus, it is necessary to verify whether any security vulnerability is introduced by the DFT and DFD.

6.5 TRUST ISSUES

The counterfeiting and IC/IP overuse issues in the hardware supply chain is discussed in this section. The US Department of Commerce defines a counterfeit component as one that

- is an unauthorized copy
- does not conform to original chip manufacturer (OCM) design, model, and/or performance standards
- is not produced by the OCM or is produced by unauthorized contractors
- is an off-specification, defective, or used OCM product sold as new or working; or

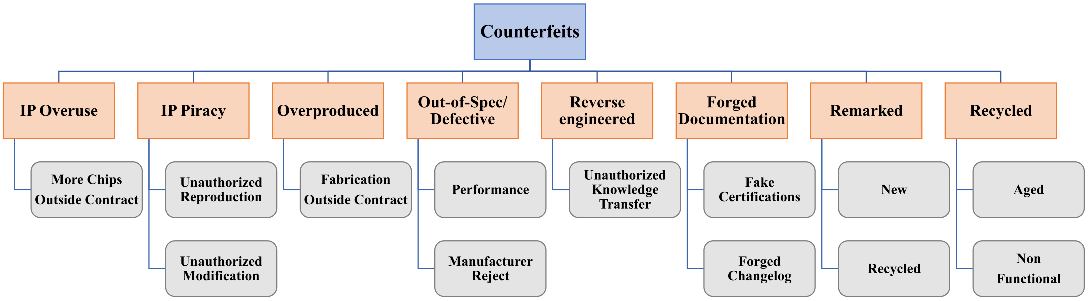


FIGURE 6.7

Taxonomy of counterfeit IPs and ICs.

- has incorrect or false markings and/or documentation

The above definition does not include all possible scenarios, where an entity in the component supply chain sources electronic components that are authentic and certified by the OCMs. For example, one may copy the entire design of a component by reverse engineering [21,22], manufacture them, and then sell them in the market under the OCM's identity. An untrusted foundry or assembly may source extra components without disclosing it to the OCM [23,24]. All these scenarios impact the security and reliability of a system utilizing such components. Thus, the above definition of counterfeiting was expanded using a comprehensive taxonomy of counterfeit types [25]. Figure 6.7 shows this taxonomy of counterfeit types. Descriptions of each type are given in the subsections below.

6.5.1 IP OVERUSE

The IP author/owner is the producer and legal owner of the IP. His/her interest includes providing a valuable product and preventing loss through disclosure of the IP to either competition or IP users [26]. The IP user/SoC integrator is the receiving party that seeks possession and rights to use the IP in their product. In general, the IP owner gives license to the SoC designer to integrate their IPs to a specific number of chips. A rogue SoC designer may produce more chips and report a lesser number to the IP owner in order to reduce licensing costs. Put simply, the problem is that the IP owners have little, if any, means to verify how many chips have been fabricated with their IPs. Profit is lost if the IP is used in more chips than the licensed number.

6.5.2 IP PIRACY

A dishonest SoC designer may legally purchase a 3PIP core from an IP vendor, but make clones (that is, illegitimate copies of the original IP) to sell to other SoC designers. Also, the SoC designer can make certain modification and sell the modified IP as a new IP. For example, an SoC integrator may purchase a crypto-accelerator IP from an IP owner. He/she then develops an cryptographic hash engine to calculate the digest. Then, the rogue SoC designer can sell the crypto-accelerator with a hash engine as a new IP to other SoC designers.

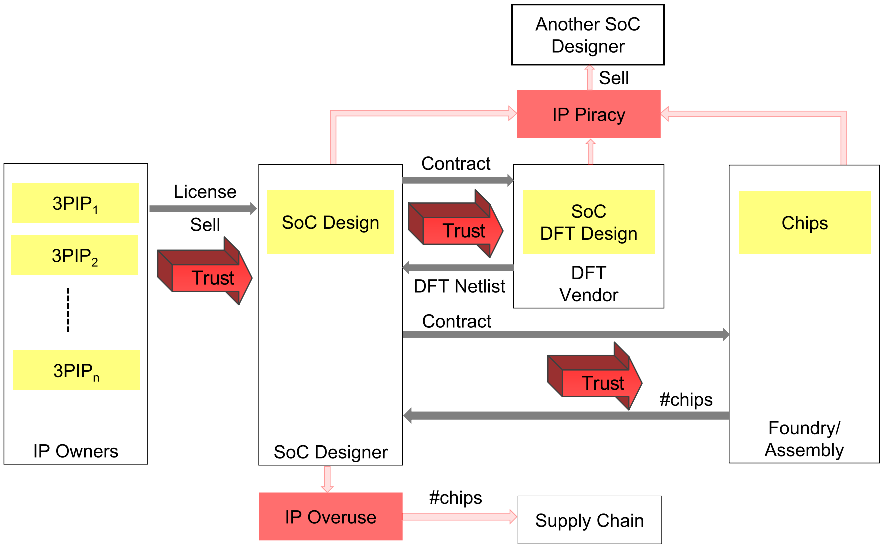


FIGURE 6.8

Lack of trust between 3PIP vendors and SoC designers, SoC designers and DFT vendor, and SoC designers and foundries in modern design/fabrication flow.

The SoC designer can also be potential victim of IP piracy. When the SoC design is outsourced to a third party vendor for synthesis or DFT insertion, that vendor has access to the entire design. As an example, a rogue DFT vendor working on the netlist version of the SoC can sell parts of the SoC design as firm IP to other SoC designers. Similarly, untrusted foundries may sell illegal copies of the GDSII files that they receive from SoC designers for fabrication.

Figure 6.8 shows the lack of trust between 3PIP vendors and SoC designers, SoC designers and DFT vendors, and SoC designers and foundries in modern design/fabrication flow; and how the lack of trust has led to IP overuse and piracy.

Note that the current semiconductor IP market is valued at $3.306 billion, and is estimated to reach $6.45 billion by 2022 [27] with the emergence of IoT devices. Thus, IP owners have a clear economic incentive to protect their products and, therefore, IP overuse and IP piracy pose a significant threat to them.

6.5.3 OVERPRODUCTION OF INTEGRATED CIRCUITS

Untrusted foundries and assemblies can produce more than the number of chips they are contracted to manufacture [28,29]. As no R&D cost are incurred for these chips, they as a result will receive larger profits by selling these chips under the SoC designer’s name. In addition, they can overbuild chips practically at no cost by reporting a lower yield (that is, a higher percentage of defect-free chips to the total number of chips) to the SoC designer or IP owner.

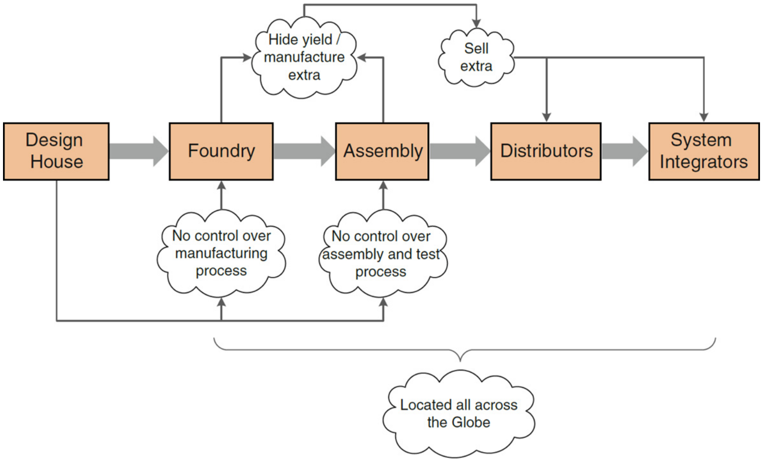

Hide yield / manufacture extra
Sell extra
Design House
Foundry
Assembly
Distributors
System Integrators
No control over manufacturing process
No control over assembly and test process
Located all across the Globe

FIGURE 6.9

Overproduction by untrusted foundry/assembly due to lack of control over fabrication and assembly of integrated circuits.

This process of manufacturing and selling outside the agreement with the design house (that is, the components' intellectual property (IP) owner) is known as "overproduction". This issue occurs because the design houses cannot monitor the fabrication and assembly process, nor obtain the actual yield information (shown in Fig. 6.9). A well understood concern with overproduction is the inevitable loss in profit for the design houses. Design companies usually invest a large amount of time and effort in the research and development (R&D) of their products. When an untrusted foundry or assembly overproduces and sells these components, the design house loses possible revenue that could have been gained from selling those components. A bigger concern with overproduced components is that of reliability. Overproduced components may simply end up in the market with minimal or no testing for reliability and functionality. These components may find their way back into the supply chain for many critical applications such as military equipment and consumer products, which raises concern for safety and reliability. Further, since these components bear the same name of the design houses, failure of these components would then tarnish the reputation of the original component manufacturer.

6.5.4 SHIPPING OUT-OF-SPEC/DEFECTIVE PARTS

A part is considered defective if it produces an incorrect response in post-manufacturing tests. As discussed in Section 6.2.2 and 6.2.3, an SoC goes through wafer test, package test, and final functional test to find if the chips are functioning according to the target specification, as indicated by Fig. 6.2. The chips rejected from these test processes should be destroyed (if they are nonfunctional), downgraded (if they are found not to satisfy the specification), or otherwise be properly disposed of. However, if they are sold on the open market instead, either knowingly by an untrusted entity or by a third-party who has stolen them, there will be an inevitable increase in their risk of failure.

6.5.5 REVERSE ENGINEERING OF INTEGRATED CIRCUITS

Reverse engineering (RE) [21,22] is the process of examining an original component in order to fully understand its structure and functionality. It can be achieved by extracting the physical interconnection information layer-by-layer destructively or non-destructively, followed by image processing analysis to reconstruct the complete structure for a component [21,30]. The prime motivation for reverse engineering a component is to make an existing copy of it, often by the competitors of the OCM. An entity involved in reverse engineering often possesses expensive and sophisticated instruments. Scanning electron microscope (SEM) or transmission electron microscope (TEM) are commonly used to take images of each layer of a component after delayering. An automated software can be used to stitch the images together to form a complete structure. For example, ICWorks Extractor from Chipworks Inc. (Ottawa, Canada) has the capability to form a 3D structure by combining all the images from the internal layers of a chip [21]. Reverse engineering can also occur by unauthorized knowledge transfer from a person with access to the part's design, which causes loss of profit for the OCM.

6.5.6 FORGED DOCUMENTATION

The documentation shipped with any component contains information regarding its specifications, testing, certificates of conformance (CoC), and statement of work (SoW). By modifying or forging these documents, a component can be misrepresented and sold, even if it is nonconforming or defective. It is often difficult to verify the authenticity of such documents, because the archived information for older designs and older parts may not be available at the OCM anymore. Legitimate documentation can also be copied and associated with parts from a lot that do not correspond with the legitimate documentation. The incentive for counterfeiters and risks associated with parts linked with forged documentation are similar to those discussed above for remarking.

6.5.7 REMARKING OF INTEGRATED CIRCUITS

Electronic components contain markings on their packages to uniquely identify them and their functionality. The marking contains information, such as part identifying number (PIN), lot identification code or date code, device manufacturer's identification, country of manufacture, electrostatic discharge (ESD) sensitivity identifier, certification mark, and so forth.

Clearly, components' markings are very important. They identify component's origin and, most importantly, determine how the component should be handled and used. For example, a space-grade component can withstand conditions (such as a wide range of temperatures and radiation levels) that would cause instant failure for a commercial-grade component. Factors such as the component manufacturer and grade also determine how much the component is worth. The price of space and military-grade components can be significantly higher than commercial grade components. For example, a BAE radiation-hardened processor, such as the RAD750, could cost in the range of tens of thousands of dollars compared to a commercial processor, which could be in the range of a few hundred dollars [32]. These space-grade processors are used in satellites, rovers, and space shuttles, and are designed to withstand a wide range of temperatures and radiation levels typically found in space. Herein lies the incentive for remarking a component (that is, changing its original markings). A counterfeiter can drive up a component's price on the open market by changing its markings to that of a higher grade or better manufacturer. However, such remarked components will not be able to withstand the

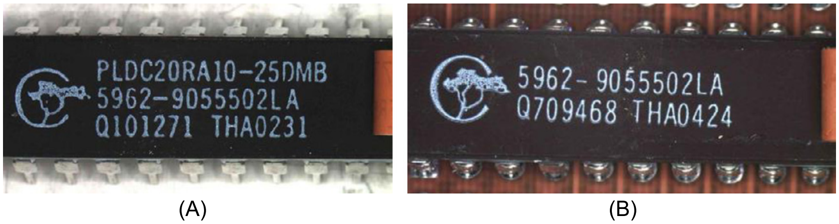


FIGURE 6.10

(A) Remarked chip, (B) original chip.

harsh conditions of their more durable, higher-grade counterparts. This can create substantial issues if such components end up in critical systems. A notable example of this is the P-8A Poseidon Aircraft incident, which was brought to light during a hearing held by the US Senate Committee on Armed Services in 2011 [33]. It was found that the ice-detection module aboard the P-8A Poseidon aircrafts, which transports anti-submarine and anti-surface warfare missiles, was found with counterfeit FPGA units. The ice-detection module is a critical component, which warns a pilot of ice that has developed on the surface of the aircraft. In this case, it was found that the FPGA units controlling the module were used and falsely remarked as being produced by Xilinx. On further analysis of the supply chain, the components were actually traced back to a manufacturer in Shenzhen, China.

It is fairly easy to remark a component that is indistinguishable from the original markings to the naked eye. A component is first prepared for remarking by either chemically or physically removing the original marking, and then blacktopping (resurfacing) the surface to hide any physical marks or imperfections that have been left from the marking-removal process. False markings are then printed either by laser marking or ink marking onto the components to appear as though produced by the OCMs. Figure 6.10 shows a remarked, and the original chip [31]. The original chip has two lines of marking. Notice that the remarking quality is good enough to make it look almost similar to the original chip.

6.5.8 RECYCLING OF INTEGRATED CIRCUITS

The term "recycled" refers to an electronic component that is reclaimed or recovered from a system, and is then modified to be misrepresented as a new component of an OCM. Recycled parts may exhibit lower performance and have a shorter lifetime due to aging, because of their prior usage. Further, the reclaiming process (removal under a very high temperature, aggressive physical removal from boards, washing, sanding, repackaging, and so on) could damage the part(s), introduce latent defects that pass initial testing but are prone to failure in later stages in the field, or make them completely nonfunctional, due to exposure to extreme conditions in an uncontrolled environment. Such parts will, of course, be unreliable and render the systems that unknowingly incorporate them equally unreliable.

The United States Senate Committee on Armed Services held a hearing regarding an investigation of counterfeit electronic parts in the defense supply chain, and the investigation revealed that e-waste from discarded electronic components are being used for these recycled counterfeit parts [34,35]. In the United States, only 25% of electronic waste was properly recycled in 2009 [36]. These figures might be comparable, or even worse, for many other developing or developed countries. This huge resource of

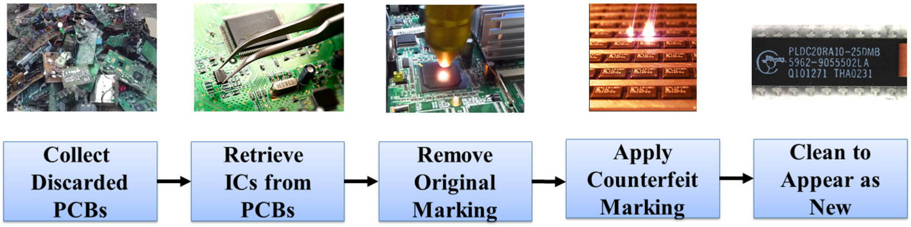


FIGURE 6.11

A typical IC recycling process.

e-waste allows counterfeiters to pile up an extremely large supply of components. These components are then recycled from the stockpile of e-waste using a crude process. A typical recycling process is as follows:

1. The recycler collects discarded printed circuit boards (PCBs) from which used components (such as digital ICs, analog ICs, capacitors, and resistors) can be harvested.
2. The PCBs are heated over an oven flame. When the soldering material starts to melt, the recycler smashes the PCB over a bucket to detach and collect the components.
3. The original marking of the components are removed by microblasting, where blasting agents are bombarded on a component's surface. Compressed air is generally used to accelerate the blasting particles. Some popular blasting agents include aluminum oxide powder, sodium bicarbonate powder, and glass bead. The choice of blasting agent depends on the components package type, such as dual in-line package (DIP) and plastic leaded chip carrier (PLCC).
4. A new coating material is applied to the component by using blacktopping and resurfacing.
5. New markings, same as the original grade-level marking, containing identification data, such as PIN number, date/lot code, manufacturer logo, and country of manufacture, are then printed either by ink printing or laser printing on the new blacktopped surface.
6. The component leads, balls, and/or columns are reworked (cleaning and straightening of leads, replating leads with new materials, forming new solder balls, and so forth) to make them appear new.

Figure 6.11 shows a recycling process documented by NASA [37]. Clearly, the recycling process impacts the reliability of recycled components, as they are subjected to harsh handling practices and impacts, such as the following:

1. The components are not protected against electrostatic discharge (ESD) and electrical overstress (EOS).
2. Moisture sensitive components are not properly baked and dry-packed.
3. The components may be damaged due to (a) high recycling temperature, (b) mechanical shock due to smashing and other handling, (c) humidity levels from cleaning with water and storage in damp conditions, and (d) other mechanical and environmental stress resulting from the recycling process.

In effect, the recycled components are degraded even further by such processes. This only exacerbates the prior effects of aging due to usage of the component in a system.

6.6 POTENTIAL COUNTERMEASURES

This section presents a brief discussion on the countermeasures that have been proposed to address the hardware supply-chain issues. Some of these techniques are based on academic research, and some are adopted by the industry. Further, this section presents the challenges associated with these techniques.

6.6.1 HARDWARE TROJAN DETECTION AND PREVENTION

Several Trojan detection and prevention approaches have been developed over the years. The readers are referred to Chapter 5 of this book for more details.

6.6.2 SECURITY RULE CHECK

To identify security vulnerabilities, unintentionally introduced by design mistake or by CAD tools, the Design Security Rule Check (DSeRC) concept was developed in [15,64]. This framework is intended to be integrated in the conventional chip design flow to analyze vulnerabilities of a design and assess its security at various stages of the design process, including the register transfer level (RTL), gate-level netlist, design-for-test (DFT) insertion, and physical design. The DSeRC framework reads the design files, constraints, and user input data, and check for vulnerabilities at all levels of abstraction (RTL, gate level, and physical layout level). Each of the vulnerabilities is tied with a set of rules and metrics, so that each design's security can be quantitatively measured. To successfully implement this framework, one needs wide-ranging access, such as to information-flow security verification, signal leakage analysis, and access control, all during the chip design flow [3,11,65]. The readers are referred to Chapter 13 of this book for more details on DSeRC framework.

6.6.3 IP ENCRYPTION

In order to protect confidentiality of IPs, and provide a common markup syntax for IP design that is interoperable across different electronic design and automation (EDA) tools and hardware flows, the IEEE SA-Standards Board developed the P1735 standard [26]. This standard has been adopted by EDA and semiconductor companies and IP vendors. The P1735 standard provides recommended practices for using encryption in order to ensure confidentiality of IP. To support interoperability and broad adoption, it also specifies a common markup format to represent an encrypted IP. The markup format uses standard-specific variables, or pragmas, to identify and encapsulate different portions of the protected IP. It also uses these pragmas to conduct functions, such as specifying the encryption and digest algorithms.

The standard also provides mechanisms to support rights management and licensing. Together these regulatory guides enable IP authors to assert fine-grained access control. With the rights management functionality, an IP author can assert which output signals are accessible to the IP user when the EDA tool simulates the IP. The licensing functionality allows access to authorized users only, for example, companies that have paid for the rights to use the IP.

The basic workflow of the standard is shown in Fig. 6.12. The standard mandates AES–CBC (but allows for other blockciphers) and RSA ($\geq$ 2048) for symmetric and asymmetric encryption, respectively. For AES it recommends a keysize of 128 or 256. Note that while the tool may perform

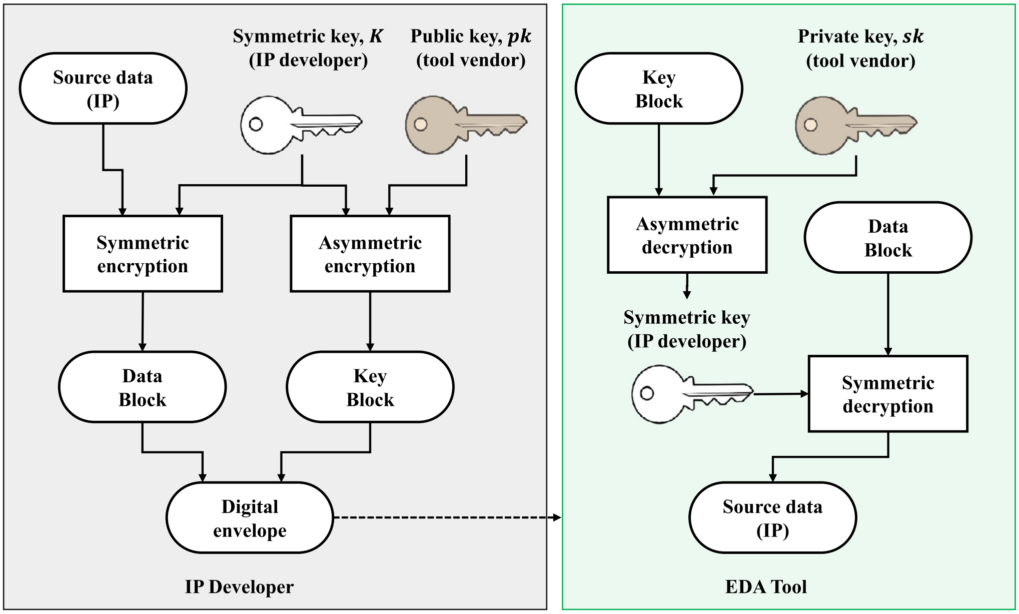


FIGURE 6.12

Workflow of the P1735 standard.

simulation, synthesis, and other processes on the IP, it never reveals the IP in its plaintext format to the IP user [26].

The current standard has unfortunately some cryptographic mistakes that have been exploited to recover the entire underlying plaintext of the encrypted IP without the knowledge of the key. The authors in [2] provide recommendation to address the limitations of the standard. Even if the limitations of IEEE-P1735 standards are addressed, the IP encryption scheme alone cannot address supply-chain issues like overproduction.

6.6.4 LOGIC OBFUSCATION

Another possible approach for preventing IP piracy and IC overproduction is through logic obfuscation. This technique places additional gates (defined as key gates) in a design to functionally lock a design, which can only be unlocked by applying the correct key [29,66,67]. For example, in Fig. 6.13B, a XOR key gate is placed to functionally lock the design shown in Fig. 6.13A. Depending on the chip unlock key $CUK[i]$ value, the D or $\overline{D}$ will appear at the output of the key gate. The correct value of the $CUK[i]$ generates the correct value of D, which is only known to the designer, who has original netlist. Ideally logic obfuscation has the potential to provide protection against both IP piracy and IC overproduction. However, different attacks, that is, SAT attack [68], key sensitization attack [69],

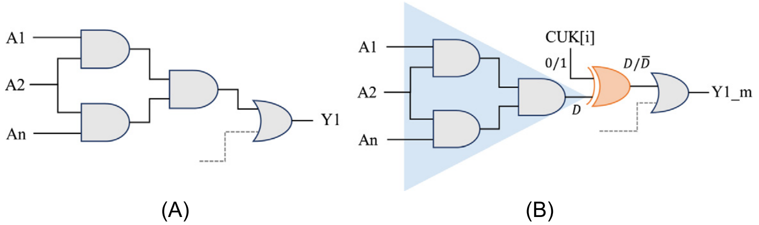


FIGURE 6.13

(A) Original netlist; (B) Obfuscated netlist with key gates shown in orange (dark gray in print version). CUK[i] represents the ith bit of the key.

removal attack have been proposed to break the logic obfuscation. These attacks utilize the locked netlist and the input-output response on the unlocked chip to extract the key.

6.6.5 HARDWARE WATERMARKING

Watermarking can be utilized to validate the authorship of an IP. Watermarking techniques uniquely identify an IP by creating a unique fingerprint in it [70–73]. As the watermarking technique is passive, one cannot use it to prevent IP overuse, IP piracy, and IC overproduction. Rather, it can only be used to verify proof of IP use.

6.6.6 IC METERING

The metering approaches have been designed to prevent IC overproduction by attempting to giving an SoC designer control over the number of ICs manufactured. These approaches can be either passive or active. Passive approaches uniquely identify each IC, and register the ICs using challenge-response pairs. Later, suspect ICs taken from the market are checked for proper registration [70,74–77]. For passive metering techniques, one major limitation is that they cannot actively prevent overproduction. For example, the SoC designers have to count on the foundries/assemblies to send them all defect-free chips, and trust them blindly on yield information. An untrusted foundry/assembly can hide actual yield information, and practically build huge amount of defect-free chips.

Active metering approaches lock each IC until it is unlocked by the SoC designer [1,24,29,78,79]. For example, Secure Split-Test (SST) [24,79] has been proposed to secure the manufacturing and testing process of SoC, and give control back to the SoC designers to prevent counterfeit, defective and/or out-of-spec SoC from entering supply chain. In SST, each chip is locked during the testing process. The SoC designer is the only entity who can interpret the locked test results and can unlock the passing chips. In this way, SST can prevent overproduction, and also prevent chips from reaching the supply chain. SST also establishes unique key for every chip, which drastically improves security against supply chain attacks. Guin et al. [1] presented an active metering approach named FORTIS, which combines the concept of IP encryption, logic obfuscation, and SST to ensure trust among all entities in the hardware supply chain. The FORTIS technique can effectively address supply-chain issues, including IP piracy, IC overproduction, out-of-spec ICs, remarked, and cloned ICs.

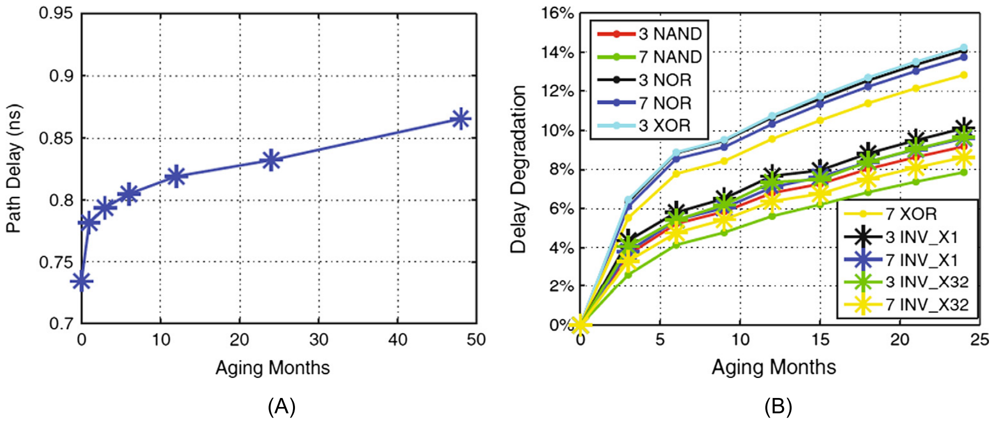


FIGURE 6.14

Path delay degradation due to aging. (A) Delay of an arbitrary path and (B) Delay degradation of different gate chains.

6.6.7 ECID AND PUF-BASED AUTHENTICATION

ECID and PUF-based authentication approaches have been proposed to identify remarked and cloned ICs. The main idea here is to tag ICs with unique IDs, and track them throughout the supply chain. The electronic-chip-ID-based (ECID-based) approaches rely on writing the unique ID into a nonprogrammable memory, such as One-Time-Programmable [OTP] and ROM. This requires post-fabrication external programming, such as laser fuses [80] or electrical fuses (eFuses) [81]. The eFuse is gaining popularity over the laser fuse because of its small area and scalability [81].

Alongside ECID, silicon physically unclonable functions (PUFs) have received much attention as a new approach for IC identification and authentication [82,83]. Silicon PUFs exploit inherent physical variations (process variations) that exist in modern integrated circuits. These variations are uncontrollable and unpredictable, making PUFs suitable for IC identification and authentication [28,84]. The variations can help generate a unique signature for each IC in a challenge-response form, which allows later identification of genuine ICs.

6.6.8 PATH-DELAY FINGERPRINTING

Path delay fingerprinting [85] was proposed to screen recycled ICs without adding extra hardware in the design. Since these recycled ICs have been used in the field, the performance of such ICs must have been degraded due to the impact of aging. Due to negative/positive bias temperature instability (NBTI/PBTI) and hot carrier injection (HCI), the path delays in recycled ICs will become larger. The larger path delays indicate higher probability of an IC being used for a long period of time in the field. Figure 6.14 shows the path delay degradation due to aging. The path was aged for 4 years, using simulation, with NBTI and HCI effects at room temperature. One can observe from Fig. 6.14A that the degradation of the path used for 1 year is around 10%, whereas if the circuit is used for 4 years, the degradation is about 17%, indicating that most aging occurred at the early usage phase of the circuit.

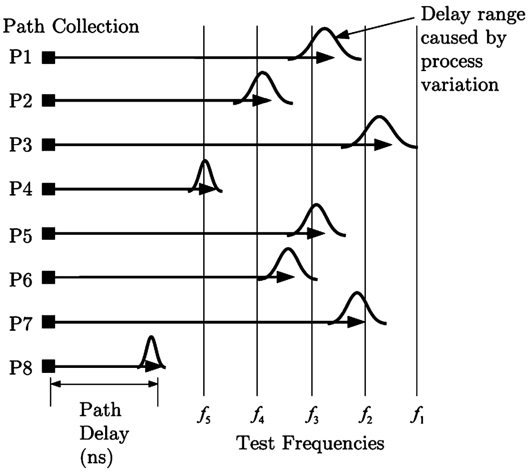


FIGURE 6.15

Clock sweeping.

Figure 6.14B presents the delay degradation of different chains, consisting of INVX1, INVX32, AND, NOR, and XOR gates, after 2 years of aging. It can be seen that different chains age at slightly different rates, which depends on the structure of the gates. The XOR gate chain has the highest aging rate, which helps to select the paths for fingerprinting.

In the path-delay fingerprinting approach, statistical data analysis is used to classify recycled (aging causes the delay variation) and new ICs (process variation causes the delay variation). Since the path delay information is measured during the manufacturing test process, no extra hardware circuitry is required for this technique.

6.6.9 CLOCK SWEEPING

The Clock sweeping technique was introduced in [86] for identifying recycled ICs. This technique applies patterns to a path multiple times with different frequencies to find a frequency at which the path cannot propagate its signal. By observing the frequencies at which the path can and cannot propagate its signal, one can measure the delay of the path with some degree of precision. The path-delay information can be used to create unique binary identifiers to differentiate between recycled and new ICs. The Clock sweeping technique has the following advantages: First, this technique can be applied to ICs already in the supply chain, including legacy designs. Second, it uses data that can be obtained through use of existing pattern sets and testing hardware capabilities. Finally, no additional hardware is necessary, as there is no area, power, or timing overhead to the technique.

Figure 6.15 shows a visual example of clock sweeping being performed on several paths. Assume that paths P1 through P8 are paths in the circuit, which end with a capturing flip-flop, and have some delay in nanoseconds. Each of the eight paths can be swept (tested) at the frequencies f_1 through f_5. All paths will be able to propagate their signal at f_1, as this is the rated frequency of the IC design. However, at f_2, the path P3 will usually fail to propagate its signal. At frequency f_3, path P3 will always

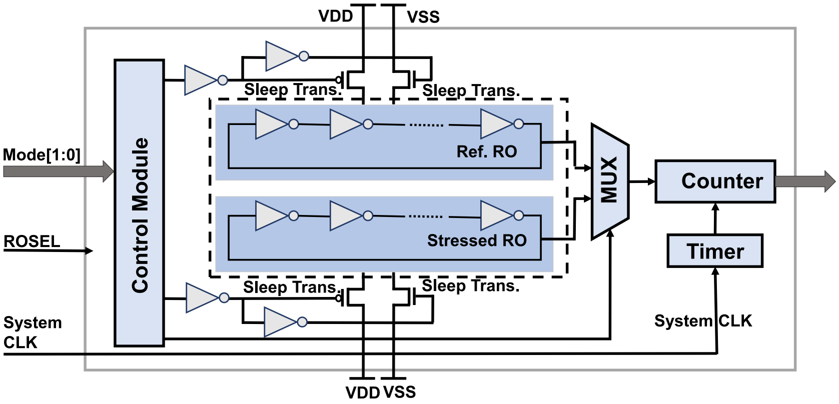


FIGURE 6.16

CDIR sensor.

fail to propagate its signal. Path P8 will succeed in propagating its signal at all five clock frequencies in this example, because it is too short to test with clock sweeping. All of the paths have some number of frequencies they will pass at, some they may fail at, and some they are guaranteed to fail at. Process variations change which frequency each path will fail at between different ICs.

6.6.10 COMBATING DIE AND IC-RECYCLING (CDIR) STRUCTURES

CDIR structures utilize IC aging phenomenon to authenticate if ICs are counterfeit or not. RO-Based CDIR sensor has been proposed in [87,88], where two ring oscillators (ROs) are embedded within the chip. The first RO is called the reference RO and is designed to age at a slow rate. The second RO is referred to as the stressed RO, and it is designed to age at a much faster rate than the reference RO. As the IC is used in the field, the stressed RO's rapid aging reduces its oscillation frequency, whereas the reference RO's oscillation frequency remains largely static over the chip's lifetime. Thus, a large disparity between the two ROs' frequencies implies that the chip has been used. To overcome global and local process variations, the two ROs are placed physically very close together so that the process and environmental variations between them are negligible.

Figure 6.16 shows the structure of this simple RO-CDIR, which is composed of a control module, a reference RO, a stressed RO, a MUX, a timer, and a counter. The counter measures the cycle count of the two ROs during a time period controlled by the timer. The system clock is used in the timer to minimize the measurement period variations due to circuit aging. The MUX selects which RO is going to be measured, and is controlled by the ROSEL signal. The inverters in the ROs can only be replaced by gates that can construct a RO, such as NAND and NOR. It will not change the effectiveness of the RO-CDIR significantly, according to prior analysis in [87]. In 90 nm technology, a 16-bit counter can operate at a frequency of up to 1 GHz, which means that an inverter-based RO must be composed of at least 21 stages [87]. The readers are referred to Chapter 12 of this book for more details on CDIR sensor.

6.6.11 ELECTRICAL TESTS

Electrical tests are efficient and nondestructive ways of detecting counterfeit ICs. The majority of defects due to counterfeiting can be detected by electrical tests. In addition, die- and bondwire-related defects may also be detected by these tests. The major advantage of introducing electrical tests in to a test plan is that they can identify cloned, out-of-spec/defective, and overproduced components along with the recycled and remarked components, as most of the electrical defects may be present in those components.

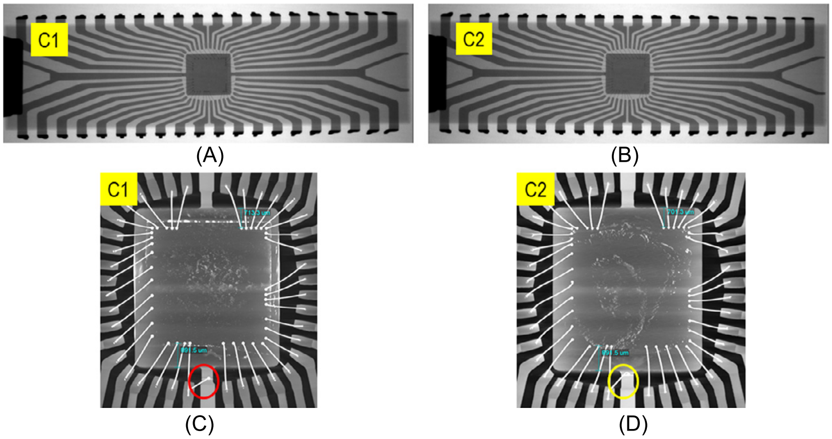


FIGURE 6.17

Counterfeit defects detected by X-ray imaging. (A) and (B) show the X-ray images of 2D lead frame view of counterfeit chips C1 and C2, respectively. (C) and (D) show the X-ray images of 3D die view of counterfeit chips C1 and C2, respectively. The red (medium gray in print version) and yellow (light gray in print version) circles represent that the bond wire connection is different between C1 and C2.

6.6.12 PHYSICAL INSPECTION

Physical inspection is the first set of tests conducted, to identify for possible evidence of counterfeiting. As part of the physical inspection procedure, the ICs are thoroughly inspected using imaging techniques of the exterior and interior. The exterior part of the package and leads of the component are analyzed using exterior tests. For example, the physical dimensions of the components are measured either by handheld or automated test equipment. Any abnormal deviation of measurement from the specification sheet indicates that the component may be counterfeit.

The chemical composition of the component is verified using material analysis. Defects such as wrong materials, contamination, oxidation of leads, and packages can be detected. There are several tests that can perform material analysis, for example XRF, EDS, and FTIR. The internal structures, such as die and bondwires of the components, may be inspected by delid/decapsulation or X-ray imaging. Figure 6.17 shows the counterfeit defects detected by X-ray imaging. There are three mainstream methods commercially available for decapsulation: chemical-, mechanical-, or laser-based solutions. Chemical decapsulation involves etching away the package with an acid solution.

Mechanical decapsulation involves grinding the part until the die is exposed. Once the part has been decapsulated and the required structures exposed, the interior tests need to be performed. These may include observation of the presence of an authentic die, gross cracks on the die, delamination, any damage on the die, die marking, missing or broken bond wires, reworked bonds, and bond-pull strength.

6.7 EXERCISES

6.7.1 TRUE/FALSE QUESTIONS

1. DFT structures are inserted only by the SoC designers.
2. Apart from fabrication, Foundries also perform tests to find defective ICs.
3. All hardware Trojans have a trigger.
4. The foundry is a trusted entity in the SoC design flow.
5. All security vulnerabilities are intentionally introduced.
6. DFT does not create security issues.
7. SoC developers are potential victims of IP piracy.
8. Out-of-spec. parts have reliability issues.

6.7.2 LONG-ANSWER TYPE QUESTIONS

1. Describe the motives of the semiconductor industry to shift to a horizontal business model.
2. How does horizontal business model reduce cost and time-to-market?
3. Why are most companies becoming fabless?
4. Describe the different types of IPs that can be procured from third-party vendors.
5. What types of tests are performed on a fabricated chip by the Foundry and the Assembly?
6. Who are the potential adversaries to implant a hardware Trojan? Provide a brief description on each of them. In your opinion, which one is most difficult to defend?
7. How can CAD tools introduce vulnerabilities? Explain with examples.
8. Why cannot test and debug structures simply be removed when they are creating unintentional security vulnerabilities?
9. Describe the taxonomy of different types of counterfeits.
10. Explain how IP overuse and IP piracy may take place. Provide respective examples.
11. Why do overproduced ICs cost less than their original counterpart?
12. Why does using out-of-spec chips pose a threat?
13. Why are Trojans inserted by the foundry difficult to detect?
14. What are the fundamental limitations of using IP encryption alone to address counterfeit issues?

6.7.3 MATHEMATICAL PROBLEMS

Let us consider the following data for a semiconductor design company “X”:

- Actual yield, $Y = 0.9$
- Die area $A = 1.5\text{ cm} \times 1.5\text{ cm}$

- Dies are processed on an $R = 200$ mm diameter wafer at \$1500/wafer
- Each mask cost \$100,000 (Total 10 masks)
- 5-man development team at \$200,000 per designer
- CAD tool costs are \$1,000,000
- Market for this part is 2,000,000 units

1. If the company wants to make 10% profit on each chip, what would be the cost of the chip in the market, calculate the cost of each chip for the semiconductor design company X?
2. Calculate the number of out-of-spec (due to yield) chips per wafer. Assume that a rogue Assembly want to sell these out-of-spec chips in the marker. How much profit will he/she make per each wafer?
3. Assume that the foundry reported a lower yield of $Y = 0.8$ instead of the actual yield ($Y = 0.9$). How would it affect the cost of each chip for the semiconductor design company X? How much profit will the foundry make per each wafer by reporting a lower yield?
4. Assume that the semiconductor design company X have placed a watermark in metal 5 layer to proof the ownership of its chip. The rogue foundry wants to eliminate the given watermark, and sell the chip as its own. To do this, the foundry recurs a cost of \$10,000 for reverse engineering and redoing metal 5 layout. Also, the foundry needs to produce a new mask for metal 5. Compare the production cost of each chip of the foundry and company X. Assume that the volume is the same for both.
5. Assume that the semiconductor design company X introduces an active metering technique to address the counterfeit issue. The active metering technique introduces 10% area overhead. Calculate the new cost of each chip with countermeasure. If the company wants to make 10% profit on each chip, what would be the new cost of the chip in the market?

REFERENCES

[1] U. Guin, Q. Shi, D. Forte, M.M. Tehranipoor, FORTIS: a comprehensive solution for establishing forward trust for protecting IPs and ICs, ACM Transactions on Design Automation of Electronic Systems (TODAES) 21 (2016) 63.

[2] A. Chhotaray, A. Nahiyan, T. Shrimpton, D. Forte, M. Tehranipoor, Standardizing bad cryptographic practice, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS), 2017, ACM, pp. 1533–1546.

[3] A. Nahiyan, M. Sadi, R. Vittal, G. Contreras, D. Forte, M. Tehranipoor, Hardware Trojan detection through information flow security verification, in: International Test Conference (DAC), 2017, IEEE, pp. 1–6.

[4] K. Xiao, D. Forte, Y. Jin, R. Karri, S. Bhunia, M. Tehranipoor, Hardware Trojans: lessons learned after one decade of research, ACM Transactions on Design Automation of Electronic Systems 22 (2016) 6.

[5] M. Tehranipoor, F. Koushanfar, A survey of hardware Trojan taxonomy and detection, IEEE Design & Test of Computers 27 (2010).

[6] J. Markoff, Old trick threatens the newest weapons, The New York Times (2009), https://www.nytimes.com/2009/10/27/science/27trojan.html.

[7] A. Nahiyan, M. Tehranipoor, Code coverage analysis for IP trust verification, in: Hardware IP Security and Trust, Springer, 2017, pp. 53–72.

[8] M. Tehranipoor, H. Salmani, X. Zhang, Integrated Circuit Authentication: Hardware Trojans and Counterfeit Detection, Springer Science & Business Media, 2013.

[9] C. Dunbar, G. Qu, Designing trusted embedded systems from finite state machines, ACM Transactions on Embedded Computing Systems (TECS) 13 (2014) 153.

[10] D.B. Roy, S. Bhasin, S. Guilley, J.-L. Danger, D. Mukhopadhyay, From theory to practice of private circuit: a cautionary note, in: Computer Design (ICCD), 2015 33rd IEEE International Conference on, IEEE, pp. 296–303.

[11] A. Nahiyan, K. Xiao, K. Yang, Y. Jin, D. Forte, M. Tehranipoor, AVFSM: a framework for identifying and mitigating vulnerabilities in FSMfs, in: Design Automation Conference (DAC), 2016 53nd ACM/EDAC/IEEE, IEEE, pp. 1–6.

[12] T. Huffmire, B. Brotherton, G. Wang, T. Sherwood, R. Kastner, T. Levin, T. Nguyen, C. Irvine, Moats and drawbridges: an isolation primitive for reconfigurable hardware based systems, in: Security and Privacy, 2007. SP'07. IEEE Symposium on, IEEE, pp. 281–295.

[13] A. Bogdanov, L.R. Knudsen, G. Leander, C. Paar, A. Poschmann, M.J. Robshaw, Y. Seurin, C. Vikkelsoe, Present: An Ultra-Lightweight Block Cipher, in: CHES, vol. 4727, Springer, 2007, pp. 450–466.

[14] OpenCores, http://opencores.org, Accessed August 2018.

[15] K. Xiao, A. Nahiyan, M. Tehranipoor, Security rule checking in IC design, Computer 49 (2016) 54–61.

[16] J. Lee, M. Tehranipoor, C. Patel, J. Plusquellic, Securing scan design using lock and key technique, in: Defect and Fault Tolerance in VLSI Systems, 2005. DFT 2005. 20th IEEE International Symposium on, IEEE, pp. 51–62.

[17] B. Yuce, N.F. Ghalaty, P. Schaumont, TVVF: estimating the vulnerability of hardware cryptosystems against timing violation attacks, in: Hardware Oriented Security and Trust (HOST), 2015 IEEE International Symposium on, IEEE, pp. 72–77.

[18] S. Morioka, A. Satoh, An optimized S-Box circuit architecture for low power AES design, in: International Workshop on Cryptographic Hardware and Embedded Systems, Springer, 2002, pp. 172–186.

[19] J. Boyar, R. Peralta, A small depth-16 circuit for the AES S-box, in: IFIP International Information Security Conference, Springer, 2012, pp. 287–298.

[20] J. Da Rolt, A. Das, G. Di Natale, M.-L. Flottes, B. Rouzeyre, I. Verbauwhede, Test versus security: past and present, IEEE Transactions on Emerging topics in Computing 2 (2014) 50–62.

[21] R. Torrance, D. James, The state-of-the-art in IC reverse engineering, in: CHES, vol. 5747, Springer, 2009, pp. 363–381.

[22] I. McLoughlin, Secure embedded systems: the threat of reverse engineering, in: Parallel and Distributed Systems, 2008. ICPADS'08. 14th IEEE International Conference on, IEEE, pp. 729–736.

[23] F. Koushanfar, G. Qu, Hardware metering, in: Proceedings of the 38th Annual Design Automation Conference, ACM, pp. 490–493.

[24] G.K. Contreras, M.T. Rahman, M. Tehranipoor, Secure split-test for preventing IC piracy by untrusted foundry and assembly, in: Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2013 IEEE International Symposium on, IEEE, pp. 196–203.

[25] U. Guin, D. DiMase, M. Tehranipoor, A comprehensive framework for counterfeit defect coverage analysis and detection assessment, Journal of Electronic Testing 30 (2014) 25–40.

[26] IEEE, 1735–2014 – IEEE recommended practice for encryption and management of electronic design intellectual property (IP), 2014.

[27] Markets Research, Global Semiconductor IP Market – Global forecast to 2022, Technical Report, https://www.marketsandmarkets.com/PressReleases/semiconductor-ip.asp. (Accessed August 2018), [Online].

[28] Y. Alkabani, F. Koushanfar, Active hardware metering for intellectual property protection and security, in: USENIX Security Symposium, pp. 291–306.

[29] R.S. Chakraborty, S. Bhunia, HARPOON: an obfuscation-based SoC design methodology for hardware protection, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 28 (2009) 1493–1502.

[30] R.J. Abella, J.M. Daschbach, R.J. McNichols, Reverse engineering industrial applications, Computers & Industrial Engineering 26 (1994) 381–385.

[31] S.C.I. Tester, Sentry counterfeit IC detector is your very own electronic sentry, guarding the entrance to your production facility from the attack of counterfeit components, https://www.abielectronics.co.uk/News/News8.php. (Accessed August 2018), [Online].

[32] J. Rhea, BAE systems moves into third generation rad-hard processors, Military & Aerospace Electronics 13 (2002).

[33] Senate Hearing 112–340, The committee's investigation into counterfeit electronic parts in the department of defense supply chain, https://www.hsdl.org/?view&did=725638. (Accessed August 2018), [Online].

[34] United States Senate Armed Services Committee, Inquiry Into Counterfeit Electronic Parts in the Department of Defense Supply Chain, https://www.hsdl.org/?view&did=709240. (Accessed August 2018), [Online].

[35] United States Senate Armed Services Committee, Suspect counterfeit electronic parts can be found on internet purchasing platforms, https://www.hsdl.org/?view&did=703697. (Accessed August 2018), [Online].

[36] United States Environmental Protection Agency, Electronic waste management in the United States through 2009, https://nepis.epa.gov/Exe/ZyPURL.cgi?Dockey=P100BKKL.TXT. (Accessed August 2018), [Online].

[37] B. Hughitt, Counterfeit electronic parts, NEPP Electron. Technol. Work, NASA Headquarters, Office of Safety and Mission Assurance, 2010.

[38] H. Salmani, M. Tehranipoor, Analyzing circuit vulnerability to hardware Trojan insertion at the behavioral level, in: Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2013 IEEE International Symposium on, IEEE, pp. 190–195.

[39] H. Salmani, M. Tehranipoor, R. Karri, On design vulnerability analysis and trust benchmarks development, in: Computer Design (ICCD), 2013 IEEE 31st International Conference on, IEEE, pp. 471–474.

[40] A. Waksman, M. Suozzo, S. Sethumadhavan, FANCI: identification of stealthy malicious logic using Boolean functional analysis, in: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, ACM, pp. 697–708.

[41] J. Zhang, F. Yuan, L. Wei, Y. Liu, Q. Xu, VeriTrust: verification for hardware trust, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 34 (2015) 1148–1161.

[42] X. Zhang, M. Tehranipoor, Case study: detecting hardware Trojans in third-party digital IP cores, in: Hardware-Oriented Security and Trust (HOST), 2011 IEEE International Symposium on, IEEE, pp. 67–70.

[43] J. Rajendran, V. Vedula, R. Karri, Detecting malicious modifications of data in third-party intellectual property cores, in: Proceedings of the 52nd Annual Design Automation Conference, ACM, p. 112.

[44] J. Rajendran, A.M. Dhandayuthapany, V. Vedula, R. Karri, Formal security verification of third party intellectual property cores for information leakage, in: VLSI Design and 2016 15th International Conference on Embedded Systems (VLSID), 2016 29th International Conference on, IEEE, pp. 547–552.

[45] Y. Jin, B. Yang, Y. Makris, Cycle-accurate information assurance by proof-carrying based signal sensitivity tracing, in: Hardware-Oriented Security and Trust (HOST), 2013 IEEE International Symposium on, IEEE, pp. 99–106.

[46] W. Hu, B. Mao, J. Oberg, R. Kastner, Detecting hardware Trojans with gate-level information-flow tracking, Computer 49 (2016) 44–52.

[47] J.J. Rajendran, O. Sinanoglu, R. Karri, Building trustworthy systems using untrusted components: a high-level synthesis approach, IEEE Transactions on Very Large Scale Integration (VLSI) Systems 24 (2016) 2946–2959.

[48] M. Hicks, M. Finnicum, S.T. King, M.M. Martin, J.M. Smith, Overcoming an untrusted computing base: detecting and removing malicious hardware automatically, in: Security and Privacy (SP), 2010 IEEE Symposium on, IEEE, pp. 159–172.

[49] C. Sturton, M. Hicks, D. Wagner, S.T. King, Defeating UCI: building stealthy and malicious hardware, in: Security and Privacy (SP), 2011 IEEE Symposium on, IEEE, pp. 64–77.

[50] J. Zhang, F. Yuan, Q. Xu, DeTrust: defeating hardware trust verification with stealthy implicitly-triggered hardware Trojans, in: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, ACM, pp. 153–166.

[51] C. Bao, D. Forte, A. Srivastava, On application of one-class SVM to reverse engineering-based hardware Trojan detection, in: Quality Electronic Design (ISQED), 2014 15th International Symposium on, IEEE, pp. 47–54.

[52] S. Bhunia, M.S. Hsiao, M. Banga, S. Narasimhan, Hardware Trojan attacks: threat analysis and countermeasures, Proceedings of the IEEE 102 (2014) 1229–1247.

[53] M. Banga, M.S. Hsiao, A novel sustained vector technique for the detection of hardware Trojans, in: VLSI Design, 2009 22nd International Conference on, IEEE, pp. 327–332.

[54] R.S. Chakraborty, S. Bhunia, Security against hardware Trojan through a novel application of design obfuscation, in: Proceedings of the 2009 International Conference on Computer-Aided Design, ACM, pp. 113–116.

[55] X. Wang, M. Tehranipoor, J. Plusquellic, Detecting malicious inclusions in secure hardware: challenges and solutions, in: Hardware-Oriented Security and Trust, 2008. HOST 2008. IEEE International Workshop on, IEEE, pp. 15–19.

[56] Y. Jin, Y. Makris, Hardware Trojan detection using path delay fingerprint, in: Hardware-Oriented Security and Trust, 2008. HOST 2008, IEEE International Workshop on, IEEE, pp. 51–57.

[57] K. Xiao, X. Zhang, M. Tehranipoor, A clock sweeping technique for detecting hardware Trojans impacting circuits delay, IEEE Design & Test 30 (2013) 26–34.

[58] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, B. Sunar, Trojan detection using fingerprinting, in: Security and Privacy, 2007. SP'07. IEEE Symposium on, IEEE, pp. 296–310.

[59] J. Aarestad, D. Acharyya, R. Rad, J. Plusquellic, Detecting Trojans through leakage current analysis using multiple supply pads, IEEE Transactions on Information Forensics and Security 5 (2010) 893–904.

[60] D. Forte, C. Bao, A. Srivastava, Temperature tracking: an innovative run-time approach for hardware Trojan detection, in: Computer-Aided Design (ICCAD), 2013 IEEE/ACM International Conference on, IEEE, pp. 532–539.

[61] F. Stellari, P. Song, A.J. Weger, J. Culp, A. Herbert, D. Pfeiffer, Verification of untrusted chips using trusted layout and emission measurements, in: Hardware-Oriented Security and Trust (HOST), 2014 IEEE International Symposium on, IEEE, pp. 19–24.

[62] B. Zhou, R. Adato, M. Zangeneh, T. Yang, A. Uyar, B. Goldberg, S. Unlu, A. Joshi, Detecting hardware Trojans using backside optical imaging of embedded watermarks, in: Design Automation Conference (DAC), 2015 52nd ACM/EDAC/IEEE, IEEE, pp. 1–6.

[63] K. Xiao, M. Tehranipoor, BISA: built-in self-authentication for preventing hardware Trojan insertion, in: Hardware-Oriented Security and Trust (HOST), 2013 IEEE International Symposium on, IEEE, pp. 45–50.

[64] A. Nahiyan, K. Xiao, D. Forte, M. Tehranipoor, Security rule check, in: Hardware IP Security and Trust, Springer, 2017, pp. 17–36.

[65] G.K. Contreras, A. Nahiyan, S. Bhunia, D. Forte, M. Tehranipoor, Security vulnerability analysis of design-for-test exploits for asset protection in SoCs, in: Design Automation Conference (ASP-DAC), 2017 22nd Asia and South Pacific, IEEE, pp. 617–622.

[66] X. Zhuang, T. Zhang, H.-H.S. Lee, S. Pande, Hardware assisted control flow obfuscation for embedded processors, in: Proceedings of the 2004 International Conference on Compilers, Architecture, and Synthesis for Embedded Systems, ACM, pp. 292–302.

[67] J.A. Roy, F. Koushanfar, I.L. Markov, Ending piracy of integrated circuits, Computer 43 (2010) 30–38.

[68] P. Subramanyan, S. Ray, S. Malik, Evaluating the security of logic encryption algorithms, in: Hardware Oriented Security and Trust (HOST), 2015 IEEE International Symposium on, IEEE, pp. 137–143.

[69] M. Yasin, J.J. Rajendran, O. Sinanoglu, R. Karri, On improving the security of logic locking, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 35 (2016) 1411–1424.

[70] F. Koushanfar, G. Qu, M. Potkonjak, Intellectual property metering, in: Information Hiding, Springer, 2001, pp. 81–95.

[71] E. Castillo, U. Meyer-Baese, A. García, L. Parrilla, A. Lloris, IPP@HDL: efficient intellectual property protection scheme for IP cores, IEEE Transactions on Very Large Scale Integration (VLSI) Systems 15 (2007) 578–591.

[72] J. Huang, J. Lach, IC activation and user authentication for security-sensitive systems, in: Hardware-Oriented Security and Trust, 2008. HOST 2008. IEEE International Workshop on, IEEE, pp. 76–80.

[73] D. Kirovski, Y.-Y. Hwang, M. Potkonjak, J. Cong, Protecting combinational logic synthesis solutions, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 25 (2006) 2687–2696.

[74] K. Lofstrom, W.R. Daasch, D. Taylor, IC identification circuit using device mismatch, in: Solid-State Circuits Conference, 2000. Digest of Technical Papers. ISSCC. 2000 IEEE International, IEEE, pp. 372–373.

[75] J.W. Lee, D. Lim, B. Gassend, G.E. Suh, M. Van Dijk, S. Devadas, A technique to build a secret key in integrated circuits for identification and authentication applications, in: VLSI Circuits, 2004. Digest of Technical Papers. 2004 Symposium on, IEEE, pp. 176–179.

[76] S.S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijen, P. Tuyls, The butterfly PUF protecting IP on every FPGA, in: Hardware-Oriented Security and Trust, 2008. HOST 2008. IEEE International Workshop on, IEEE, pp. 67–70.

[77] G.E. Suh, S. Devadas, Physical unclonable functions for device authentication and secret key generation, in: Proceedings of the 44th Annual Design Automation Conference, ACM, pp. 9–14.

[78] Y. Alkabani, F. Koushanfar, M. Potkonjak, Remote activation of ICs for piracy prevention and digital right management, in: Proceedings of the 2007 IEEE/ACM International Conference on Computer-Aided Design, IEEE Press, pp. 674–677.

[79] M.T. Rahman, D. Forte, Q. Shi, G.K. Contreras, M. Tehranipoor, CSST: preventing distribution of unlicensed and rejected ICs by untrusted foundry and assembly, in: Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2014 IEEE International Symposium on, IEEE, pp. 46–51.

[80] K. Arndt, C. Narayan, A. Brintzinger, W. Guthrie, D. Lachtrupp, J. Mauger, D. Glimmer, S. Lawn, B. Dinkel, A. Mitwalsky, Reliability of laser activated metal fuses in drams, in: Electronics Manufacturing Technology Symposium, 1999. Twenty-Fourth IEEE/CPMT, IEEE, pp. 389–394.

[81] N. Robson, J. Safran, C. Kothandaraman, A. Cestero, X. Chen, R. Rajeevakumar, A. Leslie, D. Moy, T. Kirihata, S. Iyer, Electrically programmable fuse (EFUSE): from memory redundancy to autonomic chips, in: Custom Integrated Circuits Conference, 2007. CICC'07, IEEE, IEEE, pp. 799–804.

[82] R. Pappu, B. Recht, J. Taylor, N. Gershenfeld, Physical one-way functions, Science 297 (2002) 2026–2030.

[83] L. Bolotnyy, G. Robins, Physically unclonable function-based security and privacy in RFID systems, in: Pervasive Computing and Communications, 2007. PerCom'07. Fifth Annual IEEE International Conference on, IEEE, pp. 211–220.

[84] X. Wang, M. Tehranipoor, Novel physical unclonable function with process and environmental variations, in: Proceedings of the Conference on Design, Automation and Test in Europe, European Design and Automation Association, pp. 1065–1070.

[85] X. Zhang, K. Xiao, M. Tehranipoor, Path-delay fingerprinting for identification of recovered ICs, in: Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2012 IEEE International Symposium on, IEEE, pp. 13–18.

[86] N. Tuzzio, K. Xiao, X. Zhang, M. Tehranipoor, A zero-overhead IC identification technique using clock sweeping and path delay analysis, in: Proceedings of the Great Lakes Symposium on VLSI, ACM, pp. 95–98.

[87] X. Zhang, N. Tuzzio, M. Tehranipoor, Identification of recovered ICs using fingerprints from a light-weight on-chip sensor, in: Proceedings of the 49th Annual Design Automation Conference, ACM, pp. 703–708.

[88] U. Guin, X. Zhang, D. Forte, M. Tehranipoor, Low-cost on-chip structures for combating die and IC recycling, in: Proceedings of the 51st Annual Design Automation Conference, ACM, pp. 1–6.