What I know about...

IC Security

Charles Clayton

Last updated: November 22, 2016

I. Introduction

Modern IC fabrication is prohibitively expensive for designers to manufacture their own products. Thus these companies must employ a fabless business model and outsource fabrication. Full IC designs are shared with potentially untrustworthy foundries [1] which poses risks such as IC piracy, IC overproduction, inadequate testing, and hardware Trojans (HTs) [2].

The foundry is trusted to 1) only ship ICs that pass testing, 2) only fabricate as many ICs as requested by the designer, 3) not tamper with with the design, 4) not share the design.

Inadequately protected intellectual property can negatively impact innovation, and counterfeit production can severely damage consumer welfare in the the case of critical control systems.

II. Types of Counterfeit ICs

Remarked: These counterfeit types take recycled or lower-quality ICs and remark the packaging with forged information. *Overproduced:* Foundries with access to a design can fabricate extra ICs and sell them illegally. *Cloned:* This can be done by illegally accessing the design, for instance from the foundry, or by reverse-engineering the design.

Defective: Foundries can ship ICs that fail testing, posing a threat to the reliability of a system. *Tampered:* Inserting malicious components, in the die or package level. These can include time-bombs or information back-doors [3].

A. Trojans

Conventional tests cannot be used to detect trojans, since they focus on finding incorrect behaviour rather than detecting additional functionality. The destructive technique is to employ the same process as reverse-engineering, whereby the IC is demetalized and imaged in layer-by-layer scans. This is an expensive and time-consuming process, and only works for individual ICs.

A non-destructive approach is run-time monitoring, which detects abnormal behaviour during operation. Here the IC is designed with redundancy, and the behaviour of redundant networks are compared [4]. The results of the trusted network is used and the compromised network is ignored.

Alternatively we could use test-time detection. Logic-testing focuses on feeding the IC input test-vectors to attempt to activate any Trojans and observe their effects. Exhaustive testing is infeasible because of the magnitude of possible input combinations. However since Trojans are designed to only be triggered in rare circumstances, we can take advantage of this. MERO is a technique to generate test-vectors that trigger rare nodes to rare values multiple times. This increases the likelyhood of triggering Trojans activated by rare combinations.

Additionally there are side-channel analysis-based approaches, which test physical parameters to detect inserted Trojans. These include testing for extra gates by observing leakage power, testing for extra switching activity by observing dynamic power, and testing for extra extra path by observing capacitances. However natural fabrication variations can make a small Trojan invisible in the noise.

Trojan circuitry is 3-4 orders of magnitude smaller than can be detected using signal processing techniques [**?**].

Rather than detect Trojans, which is very challenging, it is more efficient to design ICs for prevention. Designs can be obfuscated to make Trojan insertion difficult. Circuits will be transformed into equivalent circuits, but harder for attackers to reverse-engineer and insert Trojans. Alternatively, ICs can be designed to make Trojan detection easier. Power and delay sensitivity could be increased, so inserted Trojans would be more obvious. Also, reducing the amount of rare events facilitates the ease with which Trojans can be activated in testing.

III. Detecting Counterfeit ICs

A. Physical Tests

Most physical inspections are time-consuming and sometimes destructive. Low-Power Visual Inspection (LPVI) is the basic test performed on all components. Leads can be examined for fake plating, residual material, or deformations from being desoldered. The packaging can be inspected for residual label markings from recycled ICs and heat-sink markings can show prior usage.

Other techniques for investigating the structure of an IC without damaging it include Scanning Acoustic Microscopy (SAM) and Scanning Electron Microscopy (SEM), which entail using ultrasound and electron beam respectively to detecting defects or anomalies in the internal structure of the IC [5].

B. Parametric Tests

Parametric tests are time-efficient, but environmental and process variations make conclusions challenging.

DC Parametric Tests

Contact test
Power consumption test
Output short current test
Output drive current test
Threshold test
Voltage bump test

Leakage test

AC Parametric Tests

Terminal impedance timing test Rise and fall, set-up, hold, release, propagation delay tests

Time sensitivity test

Access time test

Running time test

C. Functional Tests

These are the most efficient tests for detecting defects.

Memory read/write tests

MARCH tests

Exhaustive functional testing

IV. Potential Solutions

A. Chip Identification

Each IC is provided with a unique ID. This can either be done post-fabrication using one-time-programmable memory, or by a physically unclonable function (PUF). PUFs are hardware devices that are analogous to software hashes. Output is unpredictable even if provided with physical access. These systems can exploit random process variations, such as path timing delays, to provide unique signatures [6].

These unique signatures can be used generate on-chip identification for each IC to store in a database, which can later be used to authenticate genuine ICs versus counterfeit ICs [3] and identify the foundries selling the counterfeits.

B. Active Metering

With active metering, the designer can control the use of produced ICs by implementing locking mechanisms. These locks require a unique input key to enable the IC to function properly. For instance, the startup state of the IC could be in a non-functional state, or there could be encrypted combinational logic scattered throughout the design. Only the correct sequence of inputs can render the locking logic transparent or send the IC FSM to a functioning initial state [2]. This allows the designer to lock and unlock the IC remotely [7], and requires the foundry disclose the IDs of the chips they have manufactured and want to unlock [1].

These keys can be developed to correspond with the unique IDs generated on each chip. The foundry sends the ID of each manufactured IC to the designer, and the designer then provides the key for each IC corresponding to the IDs so only the agreed ICs are unlocked [3]. Unauthorized ICs can be produced by the foundry, but may not be unlocked to function.

However, the unlocking key must be provided to the foundry in order to conduct testing, so the foundry may still ship ICs that fail testing [1]. Additionally, the foundry may pretend yield was low during testing, that is claim ICs that passed testing actually did not in order to keep and sell defect-free ICs out of contract [7].

C. Connecticut Secure Split Test (CSST)

This method builds upon the original SST by reducing the communication overhead between the designer and the foundry. There needs to be only one back-and-forth session between the two [7]. CSST prevents defective, overproduced, or cloned ICs from entering the market [5].

ICs are manufactured in a locked state and tested in the foundry while locked. The test results can only be interpreted by the designer, determining what ICs pass and what fail. The designer can then provide the keys to unlock only the chips that pass testing.

To do this, each IC generates a truly random number (TRN) and stores it in one-time programmable memory [7]. This can use physical phenonmena such as clock jitter, temperature, path delays, or power supply noise. The number is encrypted with an RSA block using a public key from the designer, which prevents the foundry from understanding how an IC is determined to pass or fail testing [1].

In testing, the encrypted value is output to the foundry so the foundry does not know the original TRN. The foundry also gathers performance tests where the outputs have been perturbed with the TRN, and sends the designer this information.

The designer decrypts the TRN using a private key then computes the all the desired test outputs. These results are compared to that sent by the foundry, and the IC is determined to have passed/failed the tests. Then the designer sends the foundry the keys required to unlock the ICs that passed.

This method has two advantages, 1) only the ICs that pass testing are unlocked thus the foundry cannot send the others to market, and 2) even if the foundry does send these to market, the designer has obtained their IC number, thus if they found in the market, they can be identified as a failed IC from the foundry [7].

1) Hardware Required: The hardware required to implement this is a functional-locking block and a scan-locking block. The functional-locking block is to ensure that locked ICs cannot function. It is made up of a TRN generator and decryption logic, which are then XOR'd together with a non-critical paths on the circuit. The XOR gate is only transparent when the TRN and the decrypted number are the same.

The scan-locking block ensures that the functional-locking block cannot be bypassed. An attacker could apply input patterns to an unlocked IC and observe the outputs in an attempt to scan out functional results [1].

2) Weakness to Tampering Attack: Although both [1] and [7] dismiss a tampering attack as extremely challenging, the secure-split test could be compromised by an attacker tampering with the layout. If the output of the TRN generator was re-routed to go directly to the output of the RSA block, then the key would not need to be provided to unlock the IC. This is challenging for an attacker because they do not know the position of the TRN generator components and the XOR mask, and the logic can be further obfuscated, but for a technologically savvy attacker with access to the design files, it is possible [1].

[8] also showed it was possible to insert Trojans without added circuitry by using stealthy dopant-level changes. This

allowed them to predict the keys of the Intel Ivy Bridge TRN generator by modifying the dopant polarity of existing transistors in few flip flops to force them to stay high or low, reducing the randomness substantially. They also were able to do this with only the FEOL layers discussed in the following section.

D. Split Manufacturing

An entirely different approach to manufacturing ICs securely is split manufacturing. The designers outsource the production of the transistors and lower metal layers, called the front end of line (FEOL) layers, to a foundry. These are the layers that require advanced technology to fabricate. However, the upper metal layers, called the back end of the line (BEOL) layers, are not included in the design provided to the foundry. The BEOL layers can then be fabricated at a trusted in-house foundry, as they require less sophisticated technology [9].

Omitting the BEOL connections from the design disclosed to an untrusted foundry then prevents reverse-engineering and any IC counterfeiting [9]. This also prevents the insertion of any Trojans that must use the BEOL connections. [10] advises splitting fabrication after the M1 layer so all gate connections are hidden, which obfuscates the design intent and provides adequate. They also demonstrate that split fabricated ICs have negligible performance, power, or area overhead compared to ICs fabricated in one facility.

[9] shows that an attacker with access to the FEOL layers can exploit the placement methods used in typical floorplanning and routing tools. They developed a method to correctly reverse-engineer 96% of the missing BEOL layers. This allows attackers to potentially pirate the design or insert Trojans, showing that simple split manufacturing is not completely secure. However, they defined the FEOL layers as $\leq$M4, whereas split fabrication after M1 provides much better circuit obfuscation [11].

Even with split fabrication, an attacker with access to the designs of the FEOL layers could still perform reliability attacks, for example by accelerating the aging of specific gates causing operational failures [11].

E. Combating Die/IC Recovery (CDIR)

F. Lightweight On-chip Sensors

G. Path-delay Fingerprinting

H. Logic Obfuscation

General circuit obfuscation impossible to achieve [**?**].

I. Source Code Encryption

REFERENCES

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

[2] M. Tehranipoor, U. Guin, and D. Forte, *Counterfeit Integrated Circuits: Detection and Avoidance*. Springer International Publishing, 2015. [Online]. Available: https://books.google.ca/books?id=2yqkBgAAQBAJ

[3] U. Guin, D. DiMase, and M. Tehranipoor, "Counterfeit integrated circuits: detection, avoidance, and the challenges ahead," *Journal of Electronic Testing*, vol. 30, no. 1, pp. 9–23, 2014.

[4] M. Tehranipoor and C. Wang, *Introduction to Hardware Security and Trust*, ser. SpringerLink : Bücher. Springer, 2011. [Online]. Available: https://books.google.ca/books?id=bNiw9448FeIC

[5] U. Guin, K. Huang, D. DiMase, J. M. Carulli, M. Tehranipoor, and Y. Makris, "Counterfeit integrated circuits: a rising threat in the global semiconductor supply chain," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1207–1228, 2014.

[6] G. Edward Suh, C. ODonnell, I. Sachdev, and D. Srinivas, "PUF overview and key management," Universty Lecture, 2005. [Online]. Available: https://pages.cs.wisc.edu/ isca2005/slides/01-03.PPT

[7] M. T. Rahman, D. Forte, Q. Shi, G. K. Contreras, and M. Tehranipoor, "CSST: An efficient secure split-test for preventing ic piracy," in *Test Workshop (NATW), 2014 IEEE 23rd North Atlantic*. IEEE, 2014, pp. 43–47.

[8] G. T. Becker, F. Regazzoni, C. Paar, and W. P. Burleson, "Stealthy dopant-level hardware trojans," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2013, pp. 197–214.

[9] J. J. Rajendran, O. Sinanoglu, and R. Karri, "Is split manufacturing secure?" in *Proceedings of the Conference on Design, Automation and Test in Europe*. EDA Consortium, 2013, pp. 1259–1264.

[10] K. Vaidyanathan, B. P. Das, E. Sumbul, R. Liu, and L. Pileggi, "Building trusted ics using split fabrication," in *Hardware-Oriented Security and Trust (HOST), 2014 IEEE International Symposium on*. IEEE, 2014, pp. 1–6.

[11] K. Vaidyanathan, B. P. Das, and L. Pileggi, "Detecting reliability attacks during split fabrication using test-only beol stack," in *Proceedings of the 51st Annual Design Automation Conference*. ACM, 2014, pp. 1–6.