

Hardware Trojan Detection Solutions and Design-for-Trust Challenges



Mohammad Tehranipoor, Hassan Salmani, Xuehui Zhang, and Xiaoxiao Wang
University of Connecticut

Ramesh Karri, Jeyavijayan Rajendran, and Kurt Rosenfeld
Polytechnic Institute of New York University

Globalization of the semiconductor industry and evolving fabrication processes have made integrated circuits increasingly vulnerable to Trojans. Researchers must expand efforts to verify trust in intellectual property cores and ICs.

Vulnerabilities in the current integrated circuit (IC) development process have raised serious concerns about possible threats from hardware Trojans to military, financial, transportation, and other critical systems.¹⁻⁴ An adversary can introduce a Trojan through an IC that will disable or destroy a system at some specific future time. Alternatively, an attacker can design a wire or some other IC component to survive the testing phase but fail before the expected lifetime. A hardware Trojan can also covertly cause a system to leak confidential information or secret keys.

Trojans can be implemented as hardware modifications to application-specific integrated circuits (ASICs), commercial off-the-shelf (COTS) parts, microprocessors, microcontrollers, network processors, or digital signal processors (DSPs), or as firmware modifications—for example, to field-programmable gate array (FPGA) bitstreams.

To ensure that an IC used by a client is authentic, either the developer must make the IC design and fabrication processes trustworthy or the client must verify the IC for trustworthiness. Because the former approach requires

a trusted design center and foundry, it is expensive and economically infeasible given current trends in the globalization of IC design and fabrication. On the other hand, verifying trustworthiness requires a postmanufacturing step to validate conformance of the fabricated IC to the original functional and performance specifications.

HARDWARE TROJANS

A system-on-chip (SoC) design can contain several layered and interconnecting functional components, including tens of intellectual property cores designed by vendors around the world. There are three basic types of IP cores:⁵

- *soft* IP cores are delivered as synthesizable register transfer level (RTL) hardware description language (HDL);
- *hard* IP cores are delivered as GDSII representations of a fully placed and routed core design; and
- *firm* IP cores are optimized in structure and topology for performance and area, possibly using a generic library (GL).

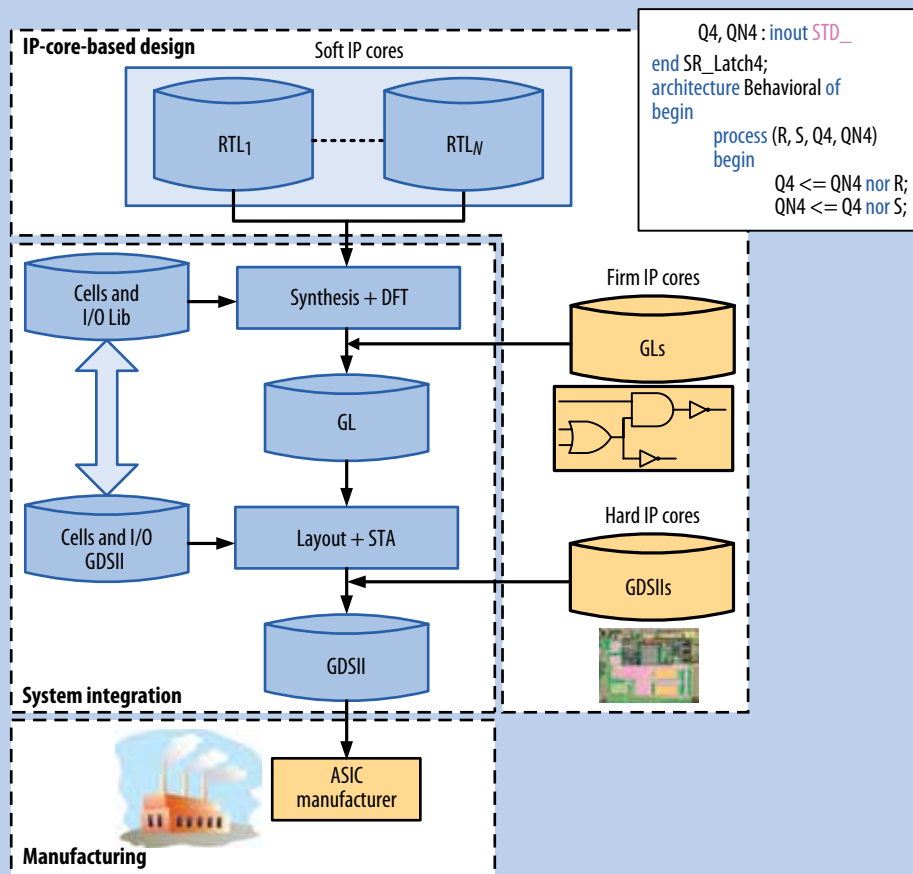


Figure 1. Modern system-on-chip design flow consists of IP-core-based design, system integration, and manufacturing.

Figure 1 shows a typical SoC design flow, which consists of IP-core-based design, system integration, and manufacturing.

Design specification, either by the design house or an outside IP core vendor, is generally the first step. Developers translate this specification into an RTL description in VHSIC HDL (VHDL) or Verilog HDL. Various RTL IP cores can be used at this stage of the design flow.

Developers synthesize the RTL description into a gate-level netlist based on the logic cells and I/Os of a target technology library, then they integrate gate-level IP cores from a vendor into this netlist. They add design-for-test (DFT) structures to improve the design's testability.

The next step is to translate the gate-level netlist into the physical layout based on cells and I/O geometries. It is possible to import IP cores from vendors in GDSII layout file format. After performing static timing analysis (STA) and power closure, developers generate the final layout in GDSII format and send it out for fabrication.

Trojans can be inserted in ICs at the RTL during design specification, at the gate level during DFT insertion, at the layout level during placement and routing, or during IC manufacturing. An attacker can also insert a Trojan through

IP cores provided by external vendors. It is thus necessary to ensure trust in all three parts of the SoC design flow.

Designers must verify the trustworthiness of IP cores as well as thoroughly test fabricated ICs to ensure that they perform as intended. In addition, because SoC design-flow activities can occur at different geographic sites, the lack of centralized control makes it extremely difficult to ensure their trustworthiness—design strategies should accordingly take trust into account.

Due to the lack of complete verification coverage for most IP cores, system integrators commonly perform additional verification and code coverage analysis. If an IP core is encrypted for piracy prevention, the vendor must provide decryption keys.

DETECTING TROJANS IN IP CORES

Ensuring trust in IP cores is extremely difficult, as there is no golden version against which to compare a given IP core during verification. In theory, an effective way to detect a Trojan in an IP core is to activate the Trojan and observe its effects, but the Trojan's type, size, and location are unknown and its activation condition is most likely a rare event.

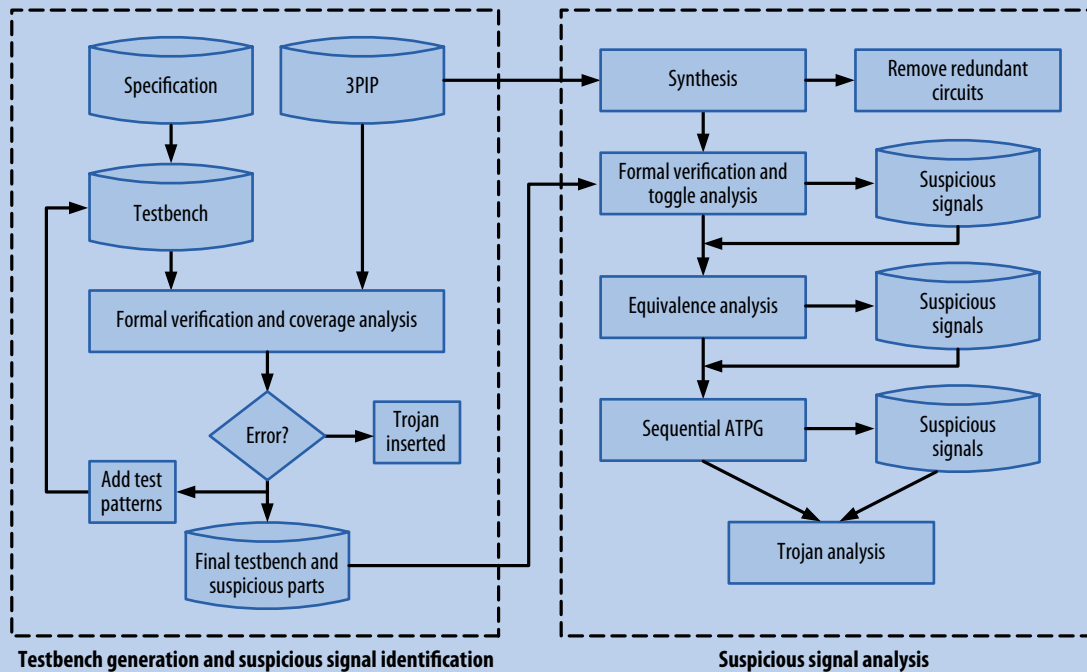


Figure 2. Detecting Trojans in an IP core requires identifying suspicious signals and components. This process has two phases: testbench generation and suspicious signal identification, followed by suspicious signal analysis.

Detecting Trojans in an IP core requires identifying suspicious signals and components.⁶ As Figure 2 shows, this process has two phases: testbench generation and suspicious signal identification, followed by suspicious signal analysis. The flow in the figure focuses on ensuring trust in soft IP cores, as they are the most dominant cores in the market today due primarily to the flexibility they offer SoC designers.

Usually, two specifications are available for each IP core. The specification from the third-party IP (3PIP) core vendor describes the core's function and cannot be trusted. However, the system integrator's requirements are trustworthy and thus can be verified.

Designers use formal verification⁶ to verify IP functionality and apply code coverage analysis to identify suspicious signals and components. Signals and components not activated during verification are suspected Trojans. Code coverage analysis includes line, statement, finite state machine (FSM), and path coverage at the RTL.

Because redundant circuits in the IP core remain at a fixed-logic value and thus cannot be activated by input patterns, designers tentatively remove these circuits from the list of suspicious signals and components. Designers revisit these circuits during the Trojan activation step, as the Trojan circuit can be redundant as well.

Sequential automatic test pattern generation (ATPG) methods generate special patterns to change signal values

during simulation. If the IP core functions perfectly with these patterns, the activated suspicious signals should be part of the original circuit; otherwise, they must be part of the Trojan.

Fault equivalence analysis reduces the number of suspicious signals,^{6,7} but activating the remaining signals is difficult. To activate these signals and further analyze their impact on circuit function, designers add new gate-level circuit structures that increase their controllability.

The last step is to determine whether the suspicious signals are actually part of a Trojan or the circuit. This step is quite fast because the suspicious signal list is considerably smaller.

Once SoC designers verify an RTL IP core's trustworthiness, they incorporate it in the regular design flow as shown in Figure 1.

DETECTING TROJANS IN FABRICATED CIRCUITS

Researchers have developed several methods to verify that fabricated ICs are free of Trojans.⁸⁻¹⁶ Side-channel techniques analyze power, timing, and other signals. Trojans typically alter the circuit design by degrading performance, changing power characteristics, and introducing reliability problems. Other techniques attempt to fully activate Trojans in an IC by targeting nodes in the circuit multiple times.⁸

Power-based signal analysis

Transient power in ICs can be used to detect Trojans. Most Trojans must be connected to the circuit's power supply lines to operate, and any transition in a Trojan will draw current from the power distribution network, where it can be measured externally. Such transitions, however, are submerged in the noise of other circuit transitions.¹⁰

To detect Trojans using power-based signal analysis, developers first identify a golden (Trojan-free) IC by conducting a battery of tests on a large number of chips. The golden IC's power signature is then obtained by applying random functional or deterministic input patterns. The measured power includes power consumed by the circuit; measurement noise, which developers can remove by making repeated measurements; and random process variations, which developers cannot remove.⁹ Any additional measured power is assumed to be consumed by a Trojan. After obtaining the reference signature, developers apply the same input patterns to the IC under authentication (IUA). If the IUA's power signature is different from the reference signature, the IUA could contain a Trojan.

Power-based signal analysis presents two major challenges. First, because of process variations in transistor parameters, no two ICs are alike, thus the power measured for the same input pattern set will be different. Second, random patterns cannot reliably generate transitions in hardware Trojans; a skillful attacker can make Trojans resilient against such patterns.

Timing-based signal analysis

Because a Trojan gate adds additional load to circuit paths, it can impact a circuit's timing characteristics.¹³⁻¹⁵ Even when the impact is small, sophisticated path-delay testing methods might be able to capture it, especially if the Trojan impacts a critical path.

Detecting Trojans using timing-based signal analysis also presents several challenges. First, differentiating Trojans from process variations is difficult because both can equally impact path delay. Second, it is extremely hard to detect a Trojan inserted on short paths in the circuit, as high frequencies are required to test these paths. Applying patterns at higher than functional frequency impacts circuit environmental noises (such as power supply noise), making detection inaccurate. Third, timing-based methods, like power-based methods, assume the existence of a golden IC. This assumption is not valid if a Trojan is inserted in all ICs.

Trojan activation methods

Trojan activation methods can accelerate the Trojan detection process and have, in some cases, been combined with power analysis during implementation. If a portion of the Trojan circuitry is activated, it will consume more dynamic power and thus make it easier to differentiate the

waveform of a Trojan-inserted circuit from that of a Trojan-free circuit. Trojan activation schemes can be categorized as either region-free or region-aware.

Region-free Trojan activation. Randomization-based methods attempt to systematically activate Trojans, regardless of where in the IC they might be located. For instance, Susmit and Sumit Jha constructed a unique probabilistic signature of a circuit based on its inputs.^{17,18} They then applied input patterns to IUAs and compared their outputs with those of the original circuit. Any difference between the outputs indicated the likely existence of a Trojan. This method assumes a high confidence level that the original design and fabricated IC are the same.

Side-channel signal analysis and Trojan activation are problematic due to rare activating nets in the circuit, process variations, and measurement noise.

Region-aware Trojan activation. Alternatively, developers can use a two-stage test-generation technique to magnify the difference between an IUA's power waveform and that of the golden IC.¹⁹⁻²¹ The first stage involves partitioning the IUA to identify potential Trojan regions. The goal is to increase activity within a Trojan circuit while simultaneously minimizing activity for the rest of the IUA. Circuit flip-flops are classified into regions according to structural connectivity, and vector sequences are generated to identify regions that exhibit increased relative activity. The second stage involves applying new vector patterns that focus on the identified regions to magnify the disparity between the original circuit and possible Trojan-inserted circuits.

DESIGNING FOR HARDWARE TRUST

Side-channel signal analysis and Trojan activation are problematic due to rare activating nets in the circuit, process variations, and measurement noise. To improve these methods' effectiveness, SoC designers must develop design-for-trust strategies.

Improving circuit net controllability

The stealthy nature of hardware Trojans suggests that they are most likely connected to circuit nets with low controllability or observability. To avoid detection using structural or functional patterns, attackers ensure that Trojans are activated only by rare conditions such as an uncommon circuit state or certain temperatures or noise.

ATPG methods for detecting defects operate on a Trojan-free circuit's netlist and thus cannot ensure trust in an IC. Instead, developers must generate test patterns to detect

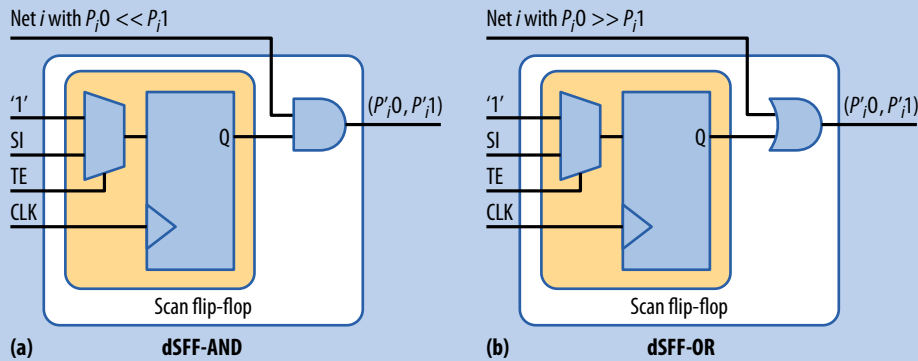


Figure 3. Dummy flip-flop structures: (a) $P_i,1 \gg P_i,0$ and (b) $P_i,0 \gg P_i,1$.

the impact of any Trojans on design characteristics beyond process and environmental variations.

A Trojan can have $q > 1$ trigger inputs that can be nets with very low transition probabilities. When the transition probability of net i is very low, either $P_i,0 \gg P_i,1$ or $P_i,1 \gg P_i,0$. The probability of generating a specific trigger vector is

$$P_{\text{trigger vector}} = \prod_{i=1}^q P_i,$$

where $P_i = \begin{cases} P_i,0 & \text{for trigger input net } i \text{ to be } 0. \\ P_i,1 & \text{for trigger input net } i \text{ to be } 1. \end{cases}$

$P_{\text{trigger vector}}$ is very low if P_i is low. By increasing the transition probability of nets with a low transition rate, it is possible to eliminate hard-to-activate sites in an IC design.

Circuit transitions are mainly caused by primary inputs and scan flip-flops. In large IC designs, the ability of primary inputs to cause switching is restricted to the first levels of circuits. However, a designer can access circuits' internal cells using a scan architecture and add a test mode to a circuit such that in this mode all scan flip-flops functionally form one or more shift registers.

To remove hard-to-activate sites in an IC, the designer can insert dummy scan flip-flops to increase the transition probability of circuit nets up to a specific threshold P_{th} .²² This in turn can expose the presence of Trojans by increasing the ratio of Trojan to circuit power consumption. For example, in Figure 3 the probabilities of '1' and '0' at the output of scan flip-flop and primary inputs are maximum and equal to 1/2. Supplying internal nets having equal '1' and '0' probabilities can increase their respective transition probabilities.

Improving localized switching

A major challenge when using power-based signal analysis to detect hardware Trojans is the large number of circuit transitions that mask transitions generated in Trojans, submerging Trojan-induced transient power into overall circuit power. Minimizing circuit switching

increases the Trojan-to-circuit activity ratio, significantly increasing the probability of detecting smaller Trojans that have a small or negligible impact on circuit power.

The total power consumption of a circuit under test is highly correlated with the total number of transitions in the scan cells during scan-based pattern application. During scan insertion, scan cells are grouped into scan chains based on different criteria. Scan chains usually scatter across the layout, and the entire design is subjected to transition using each chain. Reordering scan cells based on their final physical location in the layout can localize switching activity to one region while limiting switching activity in other regions.²³

One proposed algorithm²³ locates scan cells and restitches scan chains. First, it extracts the cells' placement information. The algorithm then removes connections between scan cells. Next, it connects cells to each other based on their location and the number of regions (N). Finally, the algorithm updates the netlist with restitched cells for potential routing.

Table 1 shows the effectiveness of scan-cell reordering in limiting switching activity in a designated target region for two ISCAS-89 sequential benchmark circuits: s38417, with 1,564 flip-flops and 4,933 gates; and s35932, with 1,728 flip-flops and 3,926 gates. For both benchmarks, we grouped scan cells into $N = 4$ scan chains using layout-aware scan-cell reordering. We ran the simulation four times and applied 132 patterns to the circuits. The patterns randomly applied '0' or '1' to the scan chain covering the target region (region 4) and '0' to all other scan chains. To increase randomness, we kept scan-enable input active (test-per-clock mode). In all four runs, switching activity was mostly limited to region 4 for both benchmarks.

Improving Trojan detection at the RTL

Developers can add ring oscillators to an IC design to detect changes that might be caused by a hardware Trojan. These special hardware structures consist of an

Table 1. Percentage of switching activity in s38417 and s35932 benchmark circuits.

Region	S38417				S35932			
	Run 1 (percent)	Run 2 (percent)	Run 3 (percent)	Run 4 (percent)	Run 1 (percent)	Run 2 (percent)	Run 3 (percent)	Run 4 (percent)
1	15.7	15.8	15.0	15.0	11.6	11.4	11.9	11.0
2	07.1	07.2	07.2	07.0	08.3	07.5	08.2	06.8
3	09.7	09.3	09.6	09.5	10.3	09.9	10.4	09.0
4 (target)	67.5	67.7	69.0	68.3	69.7	71.0	69.3	73.0

odd number of back-to-back inverters connected in a ring. A ring oscillator's frequency depends on the number of inverters and the transient characteristics of the inverters and wires. If other gates in the ring load the inverter, the oscillator's frequency changes.

The idea is to embed the circuit with ring oscillators so that any subsequent modifications to the design will change the oscillators' frequency. Figure 4, for example, shows a ring oscillator embedded in a two-bit carry-save adder. Test vectors are applied to primary inputs A1, B1, A2, B2, and C0 to activate the ring. When the detection signal is set to high, the frequency is measured on the ring oscillator output node. The ring oscillator's frequency varies if an extra gate has been inserted or the existing gate's functionality has been modified.

When the IC design is mapped to an FPGA, the embedded ring oscillators can measure the delay of paths other than those in which the oscillators are inserted. In an FPGA, the configurable logic blocks' basic elements are programmed as inverters and connected through switchboxes and wires. A ring oscillator's frequency thus

depends on the number of basic logic elements it uses, the number of switchboxes through which the wires pass, and the transient characteristics of the basic logic elements, switchboxes, and wires. The target FPGA's architecture determines the number of switchboxes, while the configuration generated by computer-aided design (CAD) tools determines the routing delays through the switchboxes and wires. A Trojan in the design could change the location of the ring oscillator's components and thereby its frequency.

Dynamically reconfiguring the ring oscillator chains extends this basic approach. The designer can create various ring oscillator chains with different frequencies and, at trust validation time, use one or more of these configurations to obtain the corresponding reference frequencies. Enlarging the number of possible ring oscillator configurations makes it difficult for an attacker to hard-code all the reference frequencies and to recognize a ring oscillator configuration on the fly so as to present the corresponding hard-coded reference frequency without additional logic.

For an ASIC design, designers can implement ring-

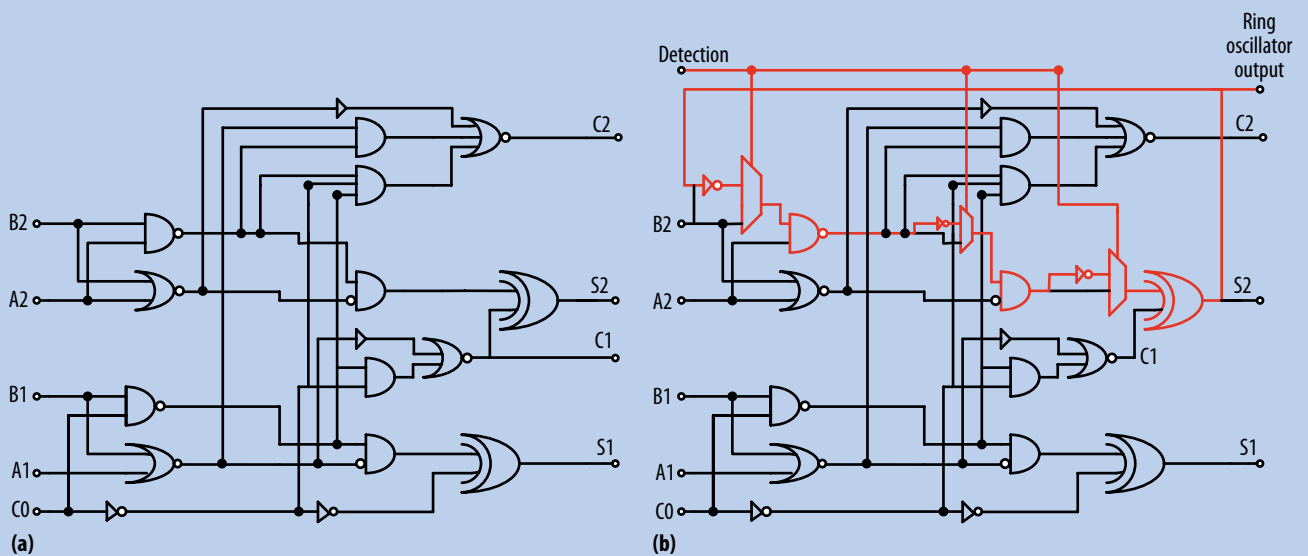


Figure 4. Two-bit carry-save adder (a) before and (b) after embedding a ring oscillator (indicated by dotted lines).

oscillator-based Trojan detection using two similar steps. Prefabrication, the designer embeds the ring oscillators and estimates the frequency using circuit simulations. The reference and surrounding frequencies can be chosen to accommodate process variation effects. Postfabrication, during IC trust validation, designers configure the ring oscillators, measure their frequencies, and check them against the reference ranges.

The main limitation of ring-oscillator-based Trojan detection is that temperature and process variations can affect the ring oscillator's frequency.

CHALLENGES

Researchers must tackle several major challenges to achieve high levels of trust in IP cores and ICs.

Hardware trust benchmarks

Researchers carry out their work on hardware trust in an uncoordinated manner in labs around the world using an assortment of homegrown reference designs and simulation

Developing meaningful trust benchmarks is necessary to compare the effectiveness of various Trojan detection approaches.

environments. Because reported results use ad hoc figures that apply to some designs, platforms, technologies, and Trojans but not to others, they are not universally accepted.

The CAD and VLSI research communities have successfully designed and implemented a common set of standardized benchmarks to demonstrate the effectiveness of various techniques. These include the ISCAS-85 combinational circuit benchmarks and ISCAS-89 sequential circuit benchmarks for VLSI logic synthesis (www.cbl.ncsu.edu/benchmarks/Benchmarks-upto-1996.html), the ITC-02 benchmarks for SoC testing (<http://itc02socbenchm.pratt.duke.edu>), and the IWLS benchmarks for electronic system-level synthesis (www.iwls.org).

Developing meaningful trust benchmarks is necessary to compare the effectiveness of various Trojan detection approaches such as side-channel signal analysis, functional test, structural test, partial activation pattern generation, and full activation pattern generation.⁸ These benchmarks are also necessary to prioritize efforts at developing defenses. An industry-accepted benchmark set can bridge the differences in current implementation platforms and styles so that Trojan detection metrics have an objective, well-defined meaning.

Researchers at the University of Connecticut, the Polytechnic Institute of New York University, Rice University, and the University of California, Los Angeles, have initiated

the Trust-Hub hardware trust benchmarking effort (www.trust-hub.org).

Experimental platforms

Researchers have validated most Trojan detection techniques either on an FPGA platform or using simulations. While easy to prototype and verify, however, FPGAs cannot capture all the circuit-level characteristics of a modern IC—for example, vulnerability to power-supply noise—and are not sensitive to Trojan-induced transient current. Test chips infected with Trojans are needed to verify detection techniques and to study the effects of process variations and measurement noise.

Trojan detection metrics

Metrics are needed to evaluate the effectiveness of different Trojan detection methods. For example, researchers need to quantify the ability of both power-based and timing-based signal analysis techniques to detect a range of Trojans in the presence of process variations. Metrics are also needed to evaluate Trojan activation methods.

Golden IC

Most Trojan detection methodologies assume the existence of a golden IC, which is obtained by arbitrarily selecting a chip from a large batch of fabricated ICs and thoroughly testing it. This procedure assumes that Trojans are inserted into random ICs, but to do so, an attacker must use a different set of masks for selected chips, making such an effort unattractive. It is more viable for an attacker to insert a stealthy Trojan into every fabricated IC that passes manufacturing tests and trust validations, obviating the need for additional expensive masks. This raises the challenge of detecting Trojans in ICs without relying on a golden IC.

Design-for-trust strategies

Current design methodologies provide multiple opportunities to insert Trojans that can go undetected. It is important to incorporate new design-for-trust strategies that prevent attackers from inserting Trojans into a design as well as effectively detect Trojans in fabricated circuits—in other words, ICs must be designed such that undetected changes are nearly impossible. We have proposed three such design-for-trust strategies for ICs and soft IP cores, and we encourage other researchers to contribute appropriate methods for other design levels.

Trust in COTS and legacy components


COTS components are commonly used in ICs. These components are usually designed and fabricated offshore and thus cannot be trusted. The challenge is to develop testing methodologies that consider COTS components' specifications and functionality without having access

to their internal structure. The internal details of components no longer supplied by the original equipment manufacturer also might not be available; the original specification might be unavailable as well. Building replacements thus is an ad hoc process. Developing techniques to validate trust in COTS and legacy components remains a difficult problem.

Hardware assurance expertise

Hardware has become a vulnerable link in the chain of trust in computing systems and must be reinforced. Even a cursory look at information assurance education reveals that hardware security is not being taught, and most research activity is focused on software assurance. Tomorrow's systems will combine hardware and software to provide security and trustworthiness, and engineers must understand the design principles and techniques that relate to both.

Lack of sufficient sponsors is also inhibiting research in hardware trust. The National Science Foundation and other government agencies, as well as the semiconductor industry, must provide more funding to facilitate R&D efforts. Only with such resources can experts address the security and trust challenges of future design and fabrication processes.

The problem of hardware security has gained significant attention during the past several years. The assumption that hardware is trustworthy and that security efforts need only focus on networks and software is no longer valid given globalization of integrated circuits and systems design and fabrication. Until researchers develop novel techniques to secure hardware, any application potentially can be considered untrusted while in the field. 

Acknowledgment

Mohammad Tehranipoor's work is supported in part by National Science Foundation grants CNS-1059390 and CNS-0844995. The work of Ramesh Karri and his group is supported in part by NSF grants ECCS-0621856, CNS-0619741, CNS-0831349, and CNS-0958510; AFRL grant FA8750-09-1-0146; and a gift from Cisco Systems.

References

1. DARPA TRUST in Integrated Circuits program, www.darpa.mil/Our_Work/MTO/Programs/Trusted_Integrated_Circuits_%28TRUST%29.aspx.
2. J.I. Lieberman, "National Security Aspects of the Global Migration of the U.S. Semiconductor Industry," June 2003, white paper, US Senate Armed Services Committee; www.ece.unm.edu/~jimp/HOST/govt_reports/liberman_semiconductor.pdf.
3. S. Adee, "The Hunt for the Kill Switch," *IEEE Spectrum*, May 2008, pp. 34-39.
4. "Innovation at Risk: Intellectual Property Challenges and Opportunities," white paper, Semiconductor Equipment and Materials Int'l, 2008.
5. VSI Alliance, *VSI Alliance Architecture Document: Version 1.0*, 1997.
6. X. Zhang and M. Tehranipoor, "A Complementary Approach for Targeting Trojans in Third-Party IP Cores," tech. report CADT-02-15-2010, Dept. Electrical and Computer Eng., Univ. Connecticut, 2010.
7. M.L. Bushnell and V.D. Agrawal, *Essentials of Electronic Testing for Digital Memory & Mixed-Signal VLSI Circuits*, Springer, 2000.
8. M. Tehranipoor and F. Koushanfar, "A Survey of Hardware Trojan Taxonomy and Detection," *IEEE Design & Test of Computers*, Jan. 2010, pp. 10-25.
9. D. Agrawal et al., "Trojan Detection Using IC Fingerprinting," *Proc. 2007 IEEE Symp. Security and Privacy (SP 07)*, IEEE CS Press, 2007, pp. 296-310.
10. X. Wang et al., "Hardware Trojan Detection and Isolation Using Current Integration and Localized Current Analysis," *Proc. 2008 IEEE Int'l Symp. Defect and Tolerance of VLSI Systems (DFT 08)*, IEEE CS Press, 2008, pp. 87-95.
11. Y. Alkabani and F. Koushanfar, "Consistency-Based Characterization for IC Trojan Detection," *Proc. 2009 Int'l Conf. Computer-Aided Design (ICCAD 09)*, ACM Press, 2009, pp. 123-127.
12. M. Potkonjak et al., "Hardware Trojan Horse Detection Using Gate-Level Characterization," *Proc. 46th Ann. Design Automation Conf. (DAC 09)*, ACM Press, 2009, pp. 688-693.
13. J. Li and J. Lach, "At-Speed Delay Characterization for IC Authentication and Trojan Horse Detection," *Proc. 2008 IEEE Int'l Workshop Hardware-Oriented Security and Trust (HOST 08)*, IEEE CS Press, 2008, pp. 8-14.
14. D. Rai and J. Lach, "Performance of Delay-Based Trojan Detection Techniques under Parameter Variations," *Proc. 2009 IEEE Int'l Workshop Hardware-Oriented Security and Trust (HOST 09)*, IEEE CS Press, 2009, pp. 58-65.
15. Y. Jin and Y. Makris, "Hardware Trojan Detection Using Path Delay Fingerprint," *Proc. 2008 IEEE Int'l Workshop Hardware-Oriented Security and Trust (HOST 08)*, IEEE CS Press, 2008, pp. 51-57.
16. X. Wang, M. Tehranipoor, and J. Plusquellic, "Detecting Malicious Inclusions in Secure Hardware: Challenges and Solutions," *Proc. 2008 Int'l Workshop Hardware-Oriented Security and Trust (HOST 08)*, IEEE CS Press, 2008, pp. 15-19.
17. S. Jha and S.K. Jha, "Randomization-Based Probabilistic Approach to Detect Trojan Circuits," *Proc. 2008 11th IEEE Symp. High Assurance Systems Eng. (HASE 08)*, IEEE CS Press, 2008, pp. 117-124.
18. F. Wolff et al., "Towards Trojan-Free Trusted ICs: Problem Analysis and Detection Scheme," *Proc. Conf. Design, Automation and Test in Europe (DATE 08)*, ACM Press, 2008, pp. 1362-1365.
19. M. Banga et al., "Guided Test Generation for Isolation and Detection of Embedded Trojans in ICs," *Proc. 18th ACM Great Lakes Symp. VLSI (GLSVLSI 08)*, ACM Press, 2008, pp. 363-366.
20. M. Banga and M.S. Hsiao, "A Region-Based Approach for the Identification of Hardware Trojans," *Proc. 2008 IEEE Int'l*

Workshop Hardware-Oriented Security and Trust (HOST 08), IEEE CS Press, 2008, pp. 40-47.

21. M. Banga and M.S. Hsiao, "A Novel Sustained Vector Technique for the Detection of Hardware Trojans," *Proc. 22nd Int'l Conf. VLSI Design (VLSID 09)*, IEEE CS Press, 2009, pp. 327-332.
22. H. Salmani, M. Tehranipoor, and J. Plusquellic, "New Design Strategy for Improving Hardware Trojan Detection and Reducing Trojan Activation Time," *Proc. 2009 IEEE Int'l Workshop Hardware-Oriented Security and Trust (HOST 09)*, IEEE CS Press, 2009, pp. 66-73.
23. H. Salmani and M. Tehranipoor, "A Layout-Aware Approach for Improving Localized Switching to Detect Hardware Trojans in Integrated Circuits," *Proc. 2010 IEEE Int'l Workshop Information Forensics and Security (WIFS 10)*, IEEE Press, 2010; doi: 10.1109/WIFS.2010.5711438.

Mohammad Tehranipoor is an associate professor in the Department of Electrical and Computer Engineering (ECE) at the University of Connecticut's School of Engineering. His research interests include reliable nanoscale systems design, secure IP/IC design, hardware security and trust, and design for testability. Tehranipoor received a PhD in electrical engineering from the University of Texas at Dallas. He is a senior member of IEEE and a member of the ACM. Contact him at tehrani@engr.uconn.edu.

Hassan Salmani is a PhD student in the ECE Department at the University of Connecticut's School of Engineering. His research interests include hardware security and trust, and low-power test. Salmani received an MS in computer engineering from Sharif University of Technology, Tehran, Iran. He is a member of IEEE. Contact him at salmani_h@engr.uconn.edu.

Xuehui Zhang is a PhD student in the ECE Department at the University of Connecticut's School of Engineering. Her research interests include hardware Trojan detection in integrated circuits and IP cores, and on-chip sensor design for reliability and temperature analysis. Zhang received an MS in computer sci-

ence and engineering from Beihang University, Beijing, China. She is a member of IEEE. Contact her at xuz09001@engr.uconn.edu.

Xiaoxiao Wang is currently a design and test engineer at Freescale Semiconductor. Her research interests include reliability analysis, on-chip structure design, and hardware security and trust. Wang received a PhD in electrical and computer engineering from the University of Connecticut. She is a member of IEEE. Contact her at xwang@engr.uconn.edu.

Ramesh Karri is a professor in the ECE Department at Polytechnic Institute of New York University. His research interests include trusted hardware design, side-channel attacks and side-channel-resistant architectures, the interaction between security and reliability, and nanoscale architectures. Karri received a PhD in computer science from the University of California, San Diego. He is a member of the IEEE Computer Society and an associate editor of IEEE Transactions on Information Forensics and Security and the ACM Journal on Emerging Technologies in Computing. Contact him at rkarri@duke.poly.edu.

Jeyavijayan Rajendran is a PhD student in the ECE Department at Polytechnic Institute of New York University. His research interests include nanoscale architectures and trusted hardware design. Rajendran received a BE in electronics and communication engineering from Anna University, Chennai, India. He is a student member of IEEE. Contact him at jrajen01@students.poly.edu.

Kurt Rosenfeld is an engineer at Google and a PhD student in the Computer Science Department at Polytechnic Institute of New York University. His research interests include hardware security and information security. Rosenfeld received an MS in electrical engineering from City College of New York. He is a student member of the ACM. Contact him at kurt@isis.poly.edu.



Selected CS articles and columns are available for free at <http://ComputingNow.computer.org>.