Hardware Security of CE Devices

By Anirban Sengupta

In my previous columns, I have covered various topics, including the evolution of the intellectual property core industry, digital integrated circuit (IC) design flow, high-level perspectives on intellectual property (IP) cores, and the transient fault resiliency of IP cores. Though one of my previous columns focused on providing a generic cognizance about the protection of reusable IP cores, it did not delve deep into the threat models and defense mechanisms against hardware trojans and IP piracy. This column discusses hardware security of consumer electronics (CE) devices, focusing primarily on threat models and defense mechanisms against two major attacks: hardware trojans and IP piracy.

VULNERABILITIES FOR THE GLOBAL SEMICONDUCTOR SUPPLY CHAIN

At the core of any CE device is hardware or a processor in the form of system-on-chip (SoC)/ICs [9], where the hardware optimization is usually performed through evolutionary techniques such as particle swarm optimization, genetic algorithms, and so forth [10], [11]. The global semiconductor supply chain for SoC/IC design is highly susceptible to hardware attacks, such as trojans and IP piracy. However, globalization in the semiconductor supply chain is inevitable due to requirements of maximizing design productivity and minimizing design cycle time. Further, keeping the entire design and manufacturing process of IC design in house increases the nonrecurring cost significantly, especially for mass-production commercial developments. Hardware attacks must be taken very seriously so that we are able to trust the operation of IC or continue using commercial off-the-shelf electronics systems and ICs irrespective of the presence of hardware trojans or other vulnerabilities [1], [2].

THREAT MODELS AND DEFENSE AGAINST TROJANS

Figure 1 (adopted from [3] and [12]) shows a typical SoC/IC design process indicating trustworthy and untrustworthy steps. A closer look at the design cycle reveals that trojan vulnerabilities are present in the third-party IP design house and foundry. (Note that while, in Figure 1, the foundry is marked trustworthy and only the third-party IP vendor is considered untrustworthy, the foundry or manufacturer may also be untrustworthy.) The in-house design integration phase is considered trustworthy, as it does not involve third-party elements.

A hardware trojan is an intentional, malicious modification of a digital system by a rogue element present in the third-party IP design house and/or the foundry. It is a back-door entry where an adversary leverages an existing foothold to attack the system. Hardware trojans can be inserted in any stage of the IC design process, which is considered untrustworthy. Untrustworthy sources are typically third-party design houses that deliver third-party IP cores (3PIPs) as schematic/VHDL/Verilog files and the foundry responsible for manufacturing. In fact, a hardware trojan may even be inserted secretly during the postmanufacturing stage. Therefore, maintaining tight control of the IC design process is critical for secured operation; however, doing so is extremely difficult due to involvement of third-party electronic design automation tools, 3PIP cores, and outsourcing to contract design houses [4].

CHALLENGES IN TROJAN SECURITY

1) *Trojan insertion*: The ease with which hardware trojans penetrate the system and remain dormant there until triggered is the main concern for security stakeholders. Trojans may be inserted absolutely anywhere in the design and at any stage of the IC design process. The current trend of isolating design from manufacturing mandates outsourcing and extreme reliance on third-party design tools and IP cores. With so many untrustworthy elements involved, there is a colossal opportunity for an adversary to insert a trojan in hardware.
2) *Trojan characteristics*: Trojans may be of multiple types that are often not comprehendible. However, the trojans are usually classified in terms of the triggering technique and payload, as well as the level of abstraction at which they are inserted. The trigger states or signals and payload of trojans are very difficult to find in the

Digital Object Identifier 10.1109/MCE.2016.2614552
Date of publication: 14 December 2016

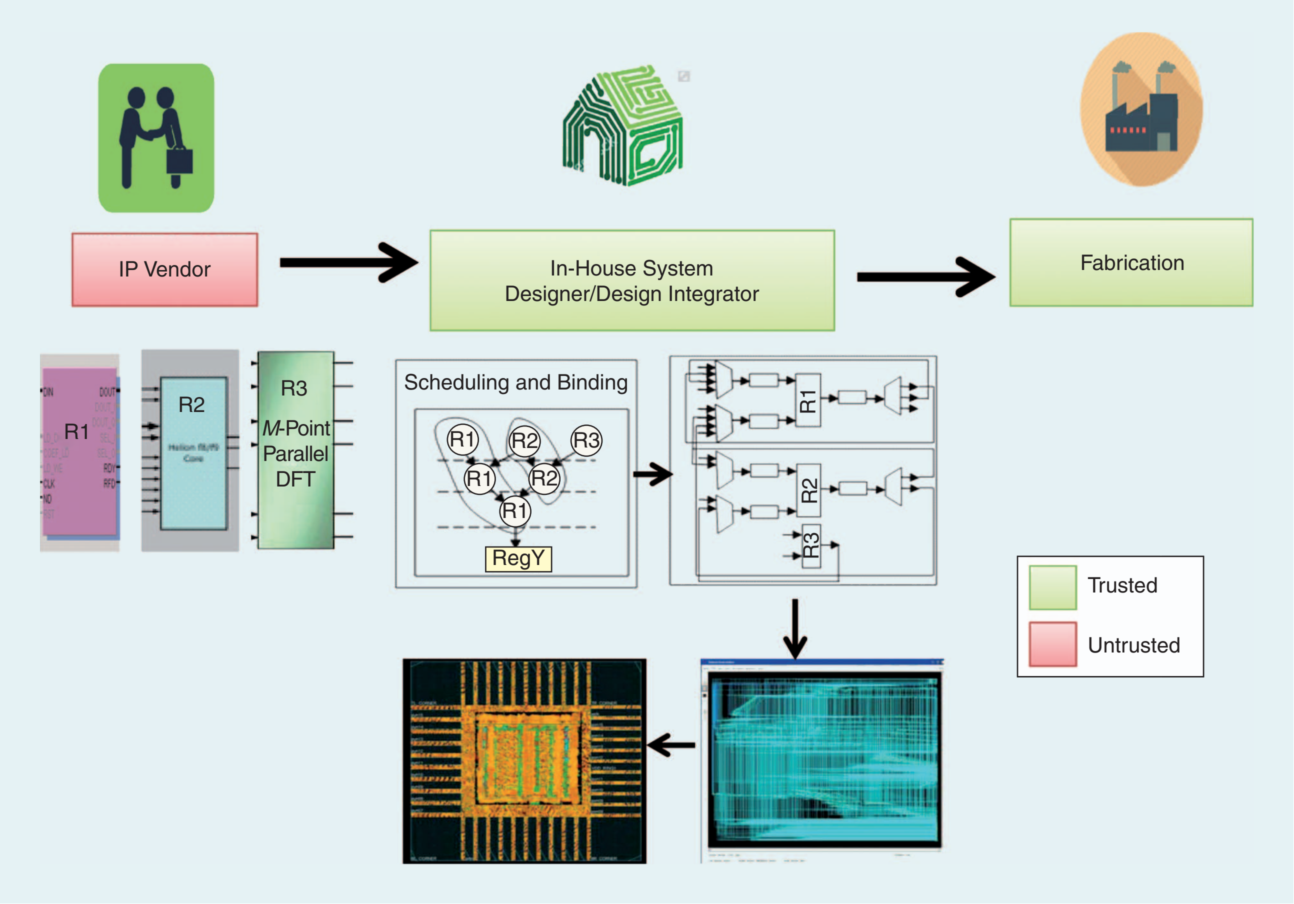

FIGURE 1. A flowchart that details the typical SoC/IC design process.

design. Trojans typically get triggered under rare conditions, which are difficult to detect; the trojan trigger signal may also be hidden in some other signal, thus deceiving the verification team.

Furthermore, some trojans may only affect a few output values, while others continue to produce correct output. Without a detailed examination, the trojan can easily escape. Thus, regular validation, such as register transfer level (RTL) functional verification, through specific test vectors is needed; however, it does not guarantee trojan detection.

Finally, some of the 3PIP cores delivered are considered preverified functional blocks and may also be large in size; thus, they do not typically undergo RTL validation. Typical payloads include change in output computational value, leaking confidential information, and degradation in performance [1]–[4].

DEFENSES AGAINST TROJANS

Let us consider the two major hardware trojan attack scenarios by an adversary in 3PIP and foundry, as shown in Figure 2. The first scenario depicts a 3PIP vendor (untrustworthy) as the attacker (marked by red star) and the SoC integrator and foundry as trustworthy elements. In this case, the shield against the trojan can be provided by the SoC integrator through incorporation of some security constraints during the SoC design process. More explicitly, this could be hardware allocation from distinct vendors to operations of the control data flow graph during high-level synthesis [3], [4]. Furthermore, the SoC integrator may employ code coverage analysis to identify suspicious signals (such as trojan triggering) through the use of specific automatic test pattern generation test vectors [5]. In the second case, the foundry is considered untrustworthy; an attacker may insert a trojan during manufacturing or postmanufacturing. Functional or structural tests such as path delay measurement, thermal profiling or a combination of these tests are used to detect trojans [6].

	3PIP Vendor	SoC Integrator	Foundry
Scenario 1	★	✚	✚
Scenario 2	✚	✚	★

FIGURE 2. A typical trojan attack scenario in the IC development cycle [6]. (Note: the red star indicates the attacker and the green "plus" symbol indicates the shielding party/protector.)

THREAT MODELS AND DEFENSE AGAINST IP PIRACY

Another type of threat to the underlying hardware of a CE device is IP piracy/IC overbuilding. An attacker with access to IP may tamper with it to steal or claim ownership. In the multimillion-dollar IC market, such false claim of ownership

	3PIP Vendor	SoC Integrator/Buyer	Foundry	Security
Scenario 1	Watermark	Attacker	—	Vendor Ownership
Scenario 2	Watermark	—	Attacker	Vendor Ownership
Scenario 3	Attacker	Fingerprint	—	Buyer Ownership

FIGURE 3. This matrix summarizes three attack scenarios related to IP piracy.

may result in catastrophic consequences for the vendor or user. As IP refers to the creation of intellect for which the owner receives legal rights, its ownership must be protected. Patents, copyrights, and trademarks are usually enough to secure rightful ownership; however, it is not enough for reusable soft IP cores, as these are delivered as RTL codes.

Securing reusable IP cores can be of two types: 1) protection from the vendor's perspective and 2) protection from the user's perspective. The first protection aims to secure the vendor (creator) against false claim of ownership by embedding a vendor signature (called a *watermark*) inside the design. The signature to be chosen is a subject of detailed discussion in the next subsection. The second protection aims to secure the buyer (or user) against being framed by a dishonest IP vendor by embedding a user signature (called a *fingerprint*). This is possible when an IP vendor sells extra copies (piracy) of the IP and blames the user. The buyer's legal protection must also be protected symmetrically [7].

SECURITY CHALLENGES AGAINST IP PIRACY

1) *Embedding a watermark:* A watermark is embedded in the design of an IP; however, inserting a watermark is not a trivial matter as it has to satisfy several properties—often conflicting in nature. An attacker may tamper with the vendor signature to reveal the watermark and falsely claim ownership. As a result, embedding a robust watermark is a mandatory criterion. Additional requisites include 1) the watermark must not be detectable through a conventional scan; 2) the watermark must incur minimal design overhead; 3) the watermark must be easily detectable and verifiable by an authentic entity, who has complete knowledge of the encoding rules; 4) the watermark must not affect or degrade the original functionality of the design; 5) the watermark must be able to withstand attacks such as IP piracy; 6) the watermark must be immune to nominal tampering; 7) the watermark embedding cost must be as low as possible; and 8) the complexity of embedding the watermark must be low [8].
2) *Embedding a fingerprint*: A fingerprint is embedded in the design of an IP; similar to the watermark, inserting a user fingerprint also has to satisfy numerous properties. An adversary may tamper with the user signature to reveal the fingerprint and claim illegal ownership of the IP. Further, an IP vendor may sell extra copies (piracy) of the IP and accuse the genuine user.

Thus, embedding a strong user fingerprint on the top of the vendor watermark is mandatory, as are the following requisites: 1) the fingerprint must not be detectable through a conventional scan; 2) the fingerprint must incur low design overhead; 3) adding the layer of the fingerprint must not distort the vendor watermark; 4) the fingerprint must be fault tolerant like watermark, in that minor tampering should still allow the user to prove ownership; 5) the fingerprint should incur low embedding costs; 6) the fingerprint must be unique to each user/buyer; 7) the embedding complexity should be minimal; and 8) fingerprint detection should not be complex for an entity with full knowledge of the encoding rules.

DEFENSES AGAINST IP PIRACY/ FALSE CLAIM OF OWNERSHIP

Let's consider the following three scenarios of attack: 1) SoC integrator/user, 2) foundry, and 3) 3PIP vendor (Figure 3). In the first case, the SoC integrator is untrustworthy and may be a threat to the vendor IP ownership. In such cases, a robust vendor watermark satisfying the properties described earlier can act as a final line of defense against such threats. A watermark may be embedded at any design abstraction level, such as at the behavioral level (at one of the high-level synthesis steps), the RT level, the field-programmable gate array level, or the layout level. Embedding a watermark at a higher abstraction level provides flexibility to choose an optimal design candidate solution (through design space exploration), which minimizes the design overhead.

Furthermore, RTL designs generated through high-level synthesis yield a watermarked embedded design that retains its security property in lower-level designs generated after logic-level synthesis. In the second case, an attack can be launched at the foundry by pirating a 3PIP after extracting from layout of the design. In such cases, the vendor watermark can protect his design from these threats. A watermark embedded at any design stage before layout is potentially capable of protecting the vendor ownership. In the third case, when the vendor is the attacker, he can sell extra copies of the IP and blame the user. The buyer's legal ownership in such a scenario can be protected by embedding a unique buyer fingerprint on the top of the vendor watermark. This provides symmetrical protection for both the parties against ownership threats. As discussed, similar to a watermark, a fingerprint may also be inserted at any stage of the design process to act as a final line of defense against untrustworthy vendor threats [6], [8].

ANALYSIS AND INFERENCE

It is evident from the discussions so far that both hardware trojan and IP piracy pose serious security threats to the hardware of CE devices. This emerges as a deep-rooted concern owing to globalization involved in the semiconductor design supply chain and multivendor IP integration practices. Trojans can also be inserted in any stage of the IC design flow and, until triggered (which usually happens under rare conditions), are not detectable. Various defense mechanisms are deployed to handle this threat; however, defense applied early in the design cycle (in higher layers of abstraction) is expected to provide security with lower overhead.

On the other hand, susceptibility to IP piracy is also huge, as multiple third-party elements are involved. Though strong defense mechanisms exist, solutions like watermarking and fingerprinting applied early in the design cycle protect against such threats at a lower embedding and implementation cost.

ABOUT THE AUTHOR

Anirban Sengupta (asengupt@iiti.ac.in) earned a Ph.D. degree and an M.S. degree in electrical and computer engineering from Ryerson University, Toronto, Canada, and he is a registered professional engineer of Ontario. He also holds a B.Tech. degree from West Bengal University of Technology, India. He is currently an assistant professor of computer science and engineering at the Indian Institute of Technology, Indore, where he directs the research lab on architectural synthesis on adaptive computation.

REFERENCES

[1] M. Beaumont, B. Hopkins, and T. Newby, "Hardware trojans—Prevention, detection, countermeasures (a literature review)," Dept. of Defense, Defense Sci. and Technology Org., Australia, DSTO-TN-1012, 2011.

[2] A. Sengupta, "Protection of IP-core designs for CE products," *IEEE Consum. Electron. Mag.*, vol. 5, pp. 83–89, Dec. 2015.

[3] A. Sengupta, S. Bhadauria, and S.P. Mohanty, "TL-HLS: Methodology for low-cost hardware trojan security aware scheduling with optimal loop unrolling factor during high level synthesis," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, to be published. doi: 10.1109/TCAD.2016.2597232.

[4] J. Rajendran, H. Zhang, and O. Sinanoglu, "High-level synthesis for security and trust," in *Proc. IEEE 19th Int. On-Line Testing Symposium*, Chania, Greece, 2013, pp. 1–6.

[5] S. Dupius, P.-S. Ba, M.-L. Flottes, G. Di Natale, and B. Rouzeyre, "New testing procedure for finding insertion sites of stealthy hardware trojans," in *Proc. Design, Automation, and Test in Europe Conf. and Exhibition*, 2015, pp. 776–781.

[6] M. Rostami, F. Koushanfar, J. Rajendran, and R. Karri, "Hardware security: Threat models and metrics," in *Proc. IEEE/ACM Int. Conf. on Computer-Aided Design*, 2013, pp. 819–823.

[7] J. Lach, W. H. Mangione-Smith, and M. Potkonjak, "Fingerprinting techniques for field-programmable gate array intellectual property protection," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 20, no. 10, pp. 1253–1261, 2001.

[8] A. Sengupta and S. Bhadauria, "Exploring low cost optimal watermark for reusable IP cores during high level synthesis," *IEEE Access*, vol. 4, pp. 2198–2215, May 2016.

[9] A. Sengupta, R. Sedaghat, Z. Zeng, "Multi objective efficient design space exploration and architectural synthesis of an application specific processor (ASP)," *Microprocess. Microsyst.*, vol. 35, no. 4, June 2011, pp. 392–404.

[10] A. Sengupta and V. K. Mishra, "Integrated particle swarm optimization (i-PSO): An adaptive design space exploration framework for power-performance tradeoff in architectural synthesis," in *Proc. IEEE Int. Symp. Quality Electronic Design*, Santa Clara, CA, March 2014, pp. 60–67.

[11] A. Sengupta, R. Sedaghat, and Z. Zeng, "Hardware efficient design of speed optimized power stringent application specific processor," in *Proc. IEEE Int. Conf. Microelectronics*, Morocco, pp. 167–170, Dec. 22, 2009.

[12] A. Sengupta, S. Bhadauria, and S. P. Mohanty, "Low cost security aware high level synthesis methodology," *IET Comput. Digital Techniques*, to be published. doi: 10.1049/iet-cdt.2016.0014.

The Art of Storage

(continued from page 129)

Micron shipped its first 3-D NAND products. Everspin announced a 256-MB MRAM chip and promised 1-GB chips by the end of the year. IBM adapted TLC to phase change memory (PCM). Samsung shipped 48-layer 3-D NAND products.

Solid-state memory is a critical element in the history and future of consumer electronic products. As shown in Figure 6, total flash memory revenue has grown over the years, with a few dips, exceeding US$35 billion by 2015. Future enterprise, as well as client applications, will drive even more flash memory to drive mobile, automotive, and other consumer applications.

> The differences between memory and storage are becoming blurred, with finer gradations in performance in nonvolatile memory possible.

In the future, nonvolatile solid-state memory devices, like STT MRAM, ReRAM, and PCMs, will replace volatile DRAM and SRAM. The differences between memory and storage are becoming blurred, with finer gradations in performance in nonvolatile memory possible. Nonvolatile solid-state memory will enable lower power and denser and higher endurance memories that will find applications in generations of consumer products, as well as enabling the enterprise applications running in data centers that power consumer applications and future connected IoT applications.

ABOUT THE AUTHOR

Tom Coughlin is a Senior Member of the IEEE.