Side-Channel based Watermarks for IP Protection

Georg T. Becker, Markus Kasper, Amir Moradi, and Christof Paar

Horst Görtz Institute for IT Security, Ruhr University Bochum, Germany

Abstract. Copyright violations are an increasing problem for hardware designers. Illegal copies of IP cores can cost manufactures millions of dollars. As one possible solution to this problem, digital watermarking for integrated circuits has been proposed in the past. We propose a new watermarking mechanism that is based on side-channels and that can easily and reliably be detected. The idea is to embed a unique signal into a side-channel of the device that serves as a watermark, similar to a side-channel based hardware Trojan. This enables the owner of the watermark to check ICs for their code using the established side-channel. But detecting the illegal use of code in a hardware design is only the first step. With watermarking the owner can also proof towards a third party (e.g a judge) that the code was illegally reused.

1 Introduction

Copyright violations are an increasing problem for hardware designers. Illegal copies of ASICs can cost manufactures millions of dollars. Furthermore, nowadays the reuse of IP cores becomes popular as the complexity of ASICs grows. It is much cheaper and faster to buy parts of the design from another party then to develop the entire design on your own. However, selling IP cores to other companies always comes with the risk of abuse. The company could illegally use the IP core in more applications than agreed on or could even resell this IP core to other companies. As one possible solution to this problem, digital watermarking for integrated circuits has been proposed in the past. Different watermarking schemes for different layers of the development process were introduced such as constrain-based watermarking [2][4] or FSM-based watermarking [5]. However, it can be quite difficult to detect these watermarks on the hardware level, especially if the watermark-protected part of the design is embedded in a bigger design and has been altered. Many of the proposed watermarking techniques in the literature do not even support detection on the hardware level, but only in higher design levels. A short overview of different watermarking techniques for IP protection can be found in [1].
We propose a new watermarking mechanism that is based on side-channels and that can easily and reliably be detected. The idea is to embed a unique signal into a side-channel of the device that serves as a watermark, similar to the hardware Trojans introduced in [3]. This enables the owner of the watermark to check ICs for their code using the established side-channel. But detecting the illegal use of code in a hardware design is only the first step. It is also important for the owner to be able to proof towards a third party (e.g a judge) that the code was illegally reused.

2 Scenario

We define two goals of our watermarking scheme:

1. Detectability: The owner can detect whether or not his code is used in an IC.

2. Non-repudiation: The owner can prove towards a third party that his code was used in an IC.

In our scenario, an attacker is a person who wants to illegally reuse parts of someone else's design and wants to stay undetected. A successful attack on this watermarking scheme would therefore mean to remove the watermark from the design. If the watermarking scheme is also used to provide a proof of ownership (goal 2), an attacker could also compromise the system if he can "hijack" the watermark, i.e., the attacker can successfully convince others that the watermark is his own watermark. We define two different attack scenarios:

1. Removing attack: The attacker removes the watermark from his IC
2. Impersonation attack: The attacker tries to detect a watermark in a foreign design and claims that this watermark is his own.

3 Watermark design

The main idea of the design of our watermark is similar to the side-channel based hardware Trojan introduced in [3]. The difference is that instead of using the side-channel to leak out secret information, a watermarking-signal that identifies the owner of the watermark is leaked out. We examine two approaches how to embed the watermarks into the side-channel:

1. Spread spectrum based watermark
2. Input-modulated watermark

In the spread spectrum based watermark a pseudo random number generator is used to generate a watermarking-sequence that is leaked out over a leakage circuit. If the power consumption is used, this leakage circuit could be realized using big capacitances, toggling logic or pseudo-NMOS gates. This watermarking-signal is well below the noise floor of the used side-channel. The watermark can be revealed by correlating the correct watermarking-sequence with the side-channel. This is the same method used in CDMA spread spectrum communication systems. In spread spectrum systems the transmitted signal can be recovered even if the power of the transmitted signal (in our case the watermarking-signal) is very small and well below the noise floor. This makes the watermarking-signal very robust.

The second approach uses the concept of an input-modulated hardware Trojan from [3] to build an input-modulated watermark. The main difference between the Trojan and the watermark is that the Trojan is designed to leak out secret information while the watermark leaks out a unique watermarking-sequence. The watermarking-logic consists of a combination function and a leakage circuit. The combination function uses some known input bits to compute one output bit. This output bit is again leaked out using a leakage circuit. The idea of this trojan is that we have a data-dependent power consumption that also depends on the used combination function. The owner of the watermark knows this combination function and can use this knowledge to perform a side-channel analysis. If the watermark is embedded in the device, this side-channel analysis will be successful, while it will not be successful if the watermark is not embedded.

The main difference between the spread spectrum based watermark and the input modulated watermark can be summarized as followed: For the spread spectrum watermark one (or few) measurement with a lot of sample points is used to detect the watermark. While in the input-modulated watermark a lot of measurements with different inputs but only one (or few) sample point each are used to detect the watermark.

4 Attacks

There are three basic approaches how to remove a side-channel based watermark:

1. Remove or destroy the logic that implements the watermark
2. Raise the noise of the side-channel
3. Transmit an inverse watermarking-sequence

To remove or destroy the logic that implements the watermark, this watermarking-logic needs to be located first. To locate the watermarking-logic, reverse engineering is needed. However, the watermark is designed to be very small, making it difficult to be detected in large designs. For example, watermarks with only 100-1000 gates can be realized, while ASICs can have millions of gates. Therefore, reverse engineering the chip to locate the watermarking-logic requires a lot of effort. The goal must be to make this effort at least as high, as the effort for engineering the protected design. In this case, stealing a design would be uneconomical.
The second attack approach, to raise the noise of the side-channel, is in most cases unfeasible in practice. Increasing the noise only reduces the signal to noise ratio of the watermark. The smaller the signal to noise ratio of the watermark is the more data needs to be measured to be able to detect the watermark. However, the noise of a device can only be increased to a certain degree, as otherwise the power consumption of the device becomes too big. On the other hand, a verifier can take a lot of measurements so that even a very small signal to noise ratio is sufficient to detect the watermark. Therefore, in practice the noise cannot be raised high enough to make detection impossible.
In the third attack an attacker transmits an inverse watermarking-signal. This inverse signal counterbalances the original watermarking-signal and results in constant power consumption for both signals, making it impossible to detect the watermarking-signal. But this attack is quite difficult in practice. First of all it is not easy to exactly synchronize the inverse signal with the original signal. Furthermore, the watermarking-signal is secret. The watermarking-signal is hidden well below the noise floor. Without the knowledge of the details of the used pseudo random number generator or the used combination function an attacker will not be able to transmit such an inverse watermarking-signal. Whether or not the watermarking-signal really stays secret strongly depends on the exact design of the watermark.

5 Non-repudiation

It is possible to extend the watermarking scheme to also provide non-repudiation. To do this a watermarking standard needs to be defined first. This standard defines how a hash value can be mapped on a watermark. For example, in the spread spectrum based watermark the standard could define that a linear feedback shift register is used to generate the watermark and how the initial state of the LFSR can be derived from a hash value. If someone wants to secure his design with a watermark, he first generates the hash value of his identity concatenated to a nonce. He then uses this hash value to generate the watermark in accordance to the standard. In this way an attacker cannot illegally claim that the used watermark is his own, as the attacker does not have an identity and a corresponding nonce that has the needed hash value used to create the watermark. (Finding an identity id' and nonce n' so that H($identity||nonce$)=H($id'||n'$) would mean to break the secondary preimage resistance of the hash function) To create a non-repudiational input-based watermark works in the same way. The only difference is that not a pseudo random number generator is derived from the hash value but a combination function.

The only disadvantage in the approach to derive the watermark from a hash value is that once the owner reveals the details of his watermark to someone, this person might be able to remove the watermark from future designs as an inverse watermarking-signal can be computed with this information. (But an attacker cannot remove the watermark from already produced ICs)

6 Experimental results

We implemented a spread-spectrum based watermark on the Side-channel Attack Standard Evaluation-Board (SASEBO) together with an 1st order DPA resistant AES implementation which represented the IP core we wanted to watermark. In our implementation we used an 32-bit LFSR as the PRNG and a leaking circuit which makes 256 transitions on temporary registers when the PRNG output is 1. In the first test we took measurements while the AES implementation was idle. We could clearly detect the watermark with around 5,000 clock cycles. In the second test we took measurements while the AES implementation was constantly running. The watermark could still be easily detected with 500,000 clock cycles. In the third test we tried to remove the watermark by transmitting an inverse bitstream. We added another circuit similar to the watermark with inverted output and with the same leaking circuit in order to omit the existence of the watermark. However, it turned out that the watermark was still clearly visible. We have tested this configuration when the AES core was idle. Of course the correlation coefficient between predictions and measured values has been decreased, but it is still detectable. So although both, the watermarking circuit and the "defending" watermarking circuit, are equal, detecting the watermark is still possible because the power consumption of the two circuits are not exactly the same due to the fact that they are differently routed in the circuit. This experiments illustrates how difficult is in practice to remove a watermark by transmitting an inverse signal.
These tests show that a spread-spectrum based watermark is feasible and is very robust. Without reverse engineering it will be very difficult to remove the watermark.

7 Conclusion

The proposed watermarking scheme has the advantage to be very robust and that it can easily be detected even if a lot of changes were made to the design. Furthermore, the watermark does not alter the original hardware design so that the performance of the original design stays the same. The disadvantage of this watermarking scheme is that additional logic is used. However, in big designs the size of the watermarking-logic is negligible small.

References

1. A.T. Abdel-Hamid, S. Tahar, and E.M. Abulhamid. IP Watermarking Techniques: Survey and Comparison. In *Proc. IEEE 3rd International Workshop on System-on-Chip (IWSOC'03)*, 2003.
2. A. Kahng, D. Kirovski, S. Mantik, M. Potkonjak, and J. L. Wong. Copy Detection for Intellectual Property Protection of VLSI Design. In *Proc. IEEE/ACM International Conference on Computer-Aided Design*, 199.
3. L. Lin, M. Kasper, T. Gneysu, C. Paar, and W. Burleson. Trojan Side-Channels: Lightweight Hardware Trojans through Side-Channel Engineering. In *Proc. Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, 2009.
4. N. Narayan, R. D. Newbould, J. D. Carothers, J. J. Rodrguez, and W. Timothy Holman. IP Protection for VLSI Designs Via Watermarking of Routes. In *Proc. 14th Annual IEEE International ASIC/SOC Conference*, 2001.
5. I. Torunoglu and E. Charbon. Watermarking-Based Copyright Protection of Sequential Functions. *IEEE Journal of Solid-State Circuits Vol. 35*, pages 434–440, Feb 2000.