Side-Channel Based Watermarks for Integrated Circuits

Georg T. Becker*, Markus Kasper†, Amir Moradi† and Christof Paar*†
*University of Massachusetts Amherst, USA
†Horst Görtz Institute for IT Security, Ruhr University Bochum, Germany

***Abstract*—Intellectual property (IP) right violations are an increasing problem for hardware designers. Illegal copies of IP cores can cause multi-million dollar damages and are thus considered a serious threat. One possible solution to this problem can be digital watermarking schemes for integrated circuits. We propose a new watermarking technique that employs side-channels as building blocks and can easily and reliably be detected by methods adapted from side-channel analysis. The main idea is to embed a unique signal into a side-channel of the device that serves as a watermark. This enables circuit designers to check integrated circuits for unauthorized use of their watermarked cores. The watermark is hidden below the noise floor of the side channel and is thus hidden from third parties. Furthermore, the proposed schemes can be implemented with very few gates and are thus even harder to detect and to remove. The proposed watermarks can also be realized in a programmable fashion to leak a digital signature.**

I. Introduction

The hardware design process is an expensive and time consuming task. From an economical point of view the increasing complexity of applications often prevents designing an entire chip from scratch. Instead, parts of the design are reused from earlier developments or bought from other companies as so-called IP cores (intellectual property cores). These IP cores implement specialized functionality to be used as building blocks within integrated circuit designs. IP cores can be separated into soft and hard IP cores. A soft IP core consists of an implementation in a synthesizable register transfer language (RTL) such as Verilog or VHDL. It can be sold either directly in an RTL format or as a generic gate-level netlist. RTL IP cores allow the user to adapt the licensed design to his application specific requirements, while this is cumbersome with a plain netlist IP core. Nevertheless both, netlist cores as well as RTL cores, are not restricted to a specific technology and can be ported to any process or foundry. Hard IP cores on the other hand are physical designs that come as completely laid out function blocks that cannot be modified and are thus restricted to a specific technology. Analog circuits are usually sold as hard IP.

The IP core market has risen to a multi-billion dollar business, making it a valuable target for frauds and piracy. The threats of cloned products, IP theft, and copyright infringement necessitate semiconductor designers and manufacturers to implement countermeasures into their products. Several possible protection methods against illegal IP usage have been proposed in the past. One solution to prevent IP core theft is to deliver encrypted IP cores only. These IP cores can then only be decrypted by the tools which synthesize the design and their plain sources will therefore never be seen by the customer [4]. However, this approach is logistically very difficult and does not prevent customers who legally bought the IP core from illegally sharing or reusing it for multiple designs.

A promising solution proposed to counter the IP theft threat is the concept of watermarking for IP cores [6], [9], [10], [11]. Watermarking is a wide-spread concept already used in many other contexts, e.g. picture, audio, or video data. The idea is to embed information into a signal, that is very difficult to be removed. This way each copy of the signal will also include the embedded watermark information. Watermarks are most often used in copyright protection systems for digital media to deter unauthorized copying.

When adapting the watermarking to protect designs of digital systems the threat to counter is unauthorized usage of IP cores. The goal of the designer of a circuit is therefore to be able to distinguish if his design is used in a given Integrated Circuit (IC). As the IC that needs to be tested is in most cases only available as a completely assembled and packaged chip, the goal of watermarks should be to keep the detectability even after manufacturing. In many watermarking schemes for IP cores, the watermark is implemented to protect only the high level representation of a chip design, e.g., the digital VHDL or Verilog representation. This kind of watermark cannot be detected in synthesized products anymore.

When implementing a watermark-protected design, detecting unauthorized use is the most interesting motivation. However, once having detected illegal usage it is also interesting to be able to prove this finding to a third party. This goal is called proof-of-ownership and is important to perform legal actions against the discovered theft.

We summarize the goals of IP watermarking schemes:

1) **Detectability:** Given an IC, the owner of an IP core can examine whether or not his IP core is used in the IC.
2) **Proof-of-ownership:** Given an IC, the owner of an IP core can prove to a third party that his IP core is used in the IC.

Consequently, attacking a watermarking scheme means to either violate the detectability or proof-of-ownership goal. Violating the detectability goal is to remove the embedded information of the watermark from the IP core or to render it useless. Furthermore, in a successful attack the removal of the watermark must not destroy the functionality of the IP core. Violating the detectability goal obviously also implies breaking the proof-of-ownership property. If a designer can find valid watermarks with his signature in foreign designs

This work was supported in part by NSF Grant CCF-0916854.

978-1-4244-7812-5/10/$26.00 ©2010 IEEE

the proof-of-ownership goal is violated as well. Unless special precautions are taken an attacker can in this case claim being the owner of the watermark himself.

A short overview of different watermarking techniques for IP protection can be found in [2]. Concepts for watermarks have been proposed for many different levels of the hardware design process. One of the most popular schemes suited for IP cores are the constraint-based watermarks introduced in [5], [9]. In these schemes hardware designs are tagged by defining additional design constraints which do not affect the functionality of the IP core. One major drawback of constraint-based watermarks is that the watermark can only be discovered at the same level of abstraction they were inserted, i.e., these watermarks cannot feasibly be detected in produced chips [2]. As in most cases a verifier will not have access to the high abstraction levels of a suspicious integrated circuit, this type of watermarking is impracticable for many scenarios.

The idea to use a side-channel, e.g., power consumption, to embed a watermark into an IP core has been introduced in [12]. During the reset phase of an FPGA the proposed watermark modulates the power consumption side-channel to transmit a signature by means of On-Off Keying (OOK) and Binary Phase Shift Keying (BPSK). The advantage of embedding a watermark in the characteristics of the power consumption of a circuit is that it can easily be detected even post-manufacturing, although being embedded at a high level of abstraction, e.g., as a code in a hardware description language like VHDL or Verilog.

In this contribution we propose a new watermarking technique which also employs the power side-channel of an embedded device to implement a hidden tag. This tag can also be implemented in a high-level description and allows easy and reliable detection even post manufacturing. The used methods to tag the side-channel leakage of the device originate from the concept of Trojan side-channels introduced in [8] and allow for extremely small watermark designs. We propose two different methods to implement a side-channel based watermark: a spread-spectrum based watermark and a scheme we call input-modulated watermark. While our watermarks are well suited to tag hard IP and netlist cores, they should not be applied to embed watermarks into RTL IP cores since identification and removal of the watermarking circuit would be an easy and straightforward task. While this approach is closely related to the method presented in [12], our schemes are superior to earlier published results, as they allow for much smaller watermarks. Our watermarks can also be adapted to ASICs and they allow hiding the watermark signal below the noise floor of the power side-channel. This is mainly due to our decoding mechanism: It is based on correlation and is thus very robust and requires a much smaller signal to noise ratio (SNR) than the earlier works. This property also avoids destroying the watermark signal by increasing the noise level and allows reliable operation of the watermark not only when the protected circuit is in its idle state but also when it is under heavy workload. Furthermore, the hidden nature of the watermark makes jamming the watermark signal by generating inverse watermark signals much more difficult. In [7] a very similar design to embed a side-channel watermark was proposed that is based on a thermal side-channel instead of power. While the thermal side-channel has the advantage that temperature measurements can be performed easier, it has the big disadvantage of being very slow and needing a lot of energy compared to the power side-channel. This makes it very difficult to hide the watermark, especially as a potential attacker can try to reveal the watermark using the more precise power side-channel. Thereby the attacker can gain an advantage over to the verifier that uses the heat side-channel. Overall, a power watermark is easier to hide and probably more robust than a heat based watermark, especially against transmitting an inverse watermark (see III-C).

The remainder of the paper is structured as follows. In the next sections we introduce the technical details of our watermarking schemes. At the end of each section we provide a short overview of the results from first practical implementations. We then present a scheme to extend the watermarks to achieve full proof-of-ownership support. In Section III we discuss three attack scenarios, namely reverse-engineering, raising of noise and implementing an inverse watermark. We finally conclude the paper with a short summary of the achieved results.

II. Watermark design

The main idea of the design of our watermark is similar to the side-channel based hardware Trojan introduced at CHES 2009 [8]. In [8], an artificial side-channel is used to leak out secret information. In our watermarking design we also insert an artificial side-channel into the IP core. The difference between the side-channel based hardware Trojan and the side-channel based watermark is that instead of leaking out secret information, the side-channel is engineered to contain a watermarking signal. In this paper we examine two different approaches to embed a watermarking signal into a side-channel:

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

The main difference between these two approaches can be summarized as followed: For the spread spectrum watermark a single measurement with several sample points is used to detect the watermark. In contrast, the input-modulated watermark is detected by evaluating several measurements acquired at an instance of time and with different input values.

A. Spread spectrum based watermark

In the spread spectrum based watermark a pseudo random number generator (PRNG) is used to generate a watermarking-sequence that is leaked out by means of a low power binary amplitude modulation of the power consumption. The watermark can be revealed by correlating the correct watermarking-sequence with the measured power traces. This is the same method as used in spread spectrum communication systems (also called CDMA - code division multiple access), where the transmission power of a signal is distributed over a wide bandwidth in a way that it can still be reliably recovered even if the signal is transmitted well below the noise floor. The same applies to our spread spectrum based watermark. The watermarking sequence can be leaked out well below the noise floor of the used side-channel and can still be reliably detected. This has two effects: First, the watermark is inherently very robust to noise and second, the watermark can be hidden below the noise floor of the power consumption and thus cannot be seen by an adversary. This hidden nature can be considered as a kind of physical encryption. The method of hiding information using spread spectrum has been used before in media watermarking schemes [3] and other applications such as military communication. For example, the military GPS signal is encrypted and hidden in basically the same way.

1) Embedding the watermark: The watermark consists of two parts: a PRNG and a leakage circuit. The PRNG needs to produce a pseudo-random bitstream, which needs to be unpredictable to allow hiding of the signal. Long linear feedback shift registers (LFSR) with enough state bits to prevent brute forcing the initialization vector (IV) are well suited for this task. Nevertheless, the fact that LFSRs can easily be recovered from their known output bitstreams has to be taken into account when designing the transmission power of the watermark. An attacker must not be able to recover the transmitted bitstream from the generated leakage. This assumption holds as long as the transmitted bits are properly hidden in the noise. A way to avoid this requirement is to replace the LFSR by a secure stream cipher using a secret key. However, although stream ciphers can be implemented in hardware very efficiently, they increase the complexity of the watermark design compared to simpler PRNGs.

The circuit implementing the watermark and the PRNG should be as small as possible. This reduces the cost of the watermark while allowing to hide the design in the surrounding circuit to defend reverse-engineering attacks. Thus the design has to be balanced with respect to implementations size and PRNG strength.

The second part of our watermark, the leakage circuit, maps the PRNG output to a physical power consumption. It generates additional leakage when its input is "one" and does not generate any additional leakage when its input is "zero". In an ASIC design, the leakage circuit can be implemented for example using big capacitances, toggling logic or pseudo-NMOS gates. In an FPGA implementation, circular shift registers can be used that are clocked according to the PRNG output. Note that the amount of generated leakage is part of the design space and can be engineered to take any desired signal-to-noise ratio (SNR). Although we focus on the power consumption side-channel throughout the paper, other side-channels such as EM leakage could be employed as well.

2) Detecting the watermark: The embedded watermark can be detected with similar techniques as they are used in a differential power analysis. The verifier measures a single power trace containing several clock cycles. He then compresses this power trace to a power vector with one value per clock cycle, e.g., by averaging all measurement points of each clock cycle. The verifier then simulates the bit stream generated by the PRNG using his knowledge about the implementation details. In the last detection step the verifier correlates the simulated sequence to the compressed power vector. In case he does not know the starting point of the PRNG, i.e., the correct position to align the simulated sequence to the power vector, he has to slide one (the power vector) over the other (the simulated sequence) and repeat the detection for each possible alignment. If the watermark is embedded in the examined IC, the correlation coefficient should show a prominent peak at the position of correct alignment. If the correlation coefficient does not show any significant peak, then the watermark is not embedded in the design. Using statistics to detect the watermark allows transmission with very low SNR and provides robustness to noise. The length of the simulated sequence used in the correlation step can be increased for an even more reliable detection of the watermark.

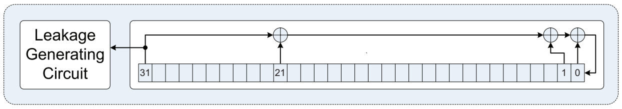
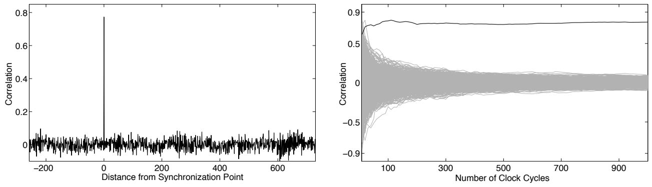


Fig. 1. Diagram of the implemented spread-spectrum based watermark

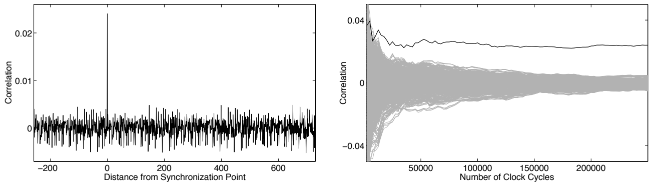
3) Experimental results: To practically evaluate our proposed watermarks we used a Side-channel Attack Standard Evaluation-Board (SASEBO) [1] that is equipped with an xc2vp7 Virtex-II Pro FPGA. The power consumption leakage was measured using a LeCroy WP715Zi 1.5GHz oscilloscope at a sampling rate of 250MS/s.

We implemented a 1st order DPA resistant AES implementation and tagged it with the spread spectrum based watermark introduced above. This setup serves as our proof of concept implementation to experimentally verify our proposed watermarking scheme. As shown by Fig. 1 we used a 32-bit LFSR as the PRNG and connected it to a leakage circuit. The leakage circuit itself was designed from 16 look-up tables (LUTs) each configured as 16-bit circular shift registers filled with alternating ones and zeros. These registers were clocked in all clock cycles where the output of the PRNG is "one". In the first experiment we measured a long power trace covering 1000 clock cycles while the AES core was idle. The power trace was then compressed by averaging over each clock cycle and then correlated to the corresponding simulated PRNG sequence. Calculating the correlation for a window of possible alignment positions leads to the vector of correlation coefficients shown in Fig. 2(a). According to Fig. 2(b), using the leakage of around 100 clock cycles would be enough to detect the existence of the watermark in this case. In the second experiment we took a longer trace (in comparison to the first one) while the AES implementation was constantly running and processing different random inputs. This time the measured power trace covered 250 000 clock cycles, and the results of the watermark detection are shown in Fig. 3. Clearly adapting the number of clock cycles used for correlation can overcome the existence of the either intentional or instinctive noise and makes the detection of the watermark feasible. An implementer of the watermark has thus two handles to design the watermark detection properties: He can use longer sequences during detection or he can increase the amount of the generated leakage. The later one has to be used carefully to ensure that the generated leakage is still hidden below the noise floor.



(a) (b)

Fig. 2. Analysis of the spread spectrum based watermark while the AES core was idle and waiting for the next plaintext (a) using the leakage of 1000 clock cycles, (b) over the number of clock cycles



(a) (b)

Fig. 3. Analysis of the spread spectrum based watermark while the AES core was constantly encrypting (a) using the leakage of 250 000 clock cycles, (b) over the number of clock cycles

B. Input-modulated watermark

The second approach to implement a watermark we propose in this paper uses the concept of an input-modulated hardware Trojan as introduced at CHES 2009 [8]. We call this proposal an input-modulated watermark. The idea of an input-modulated hardware Trojan is to add additional logic to the IC that results in a power consumption which relies on the added logic, known input bits and some secret bits. This power consumption can then be exploited using a differential power analysis to reveal the secret bits. The main difference between the input-modulated hardware Trojan and our watermark is that the Trojan is designed to leak out secret information while the watermark is not supposed to leak out any unknown information but only its presence.

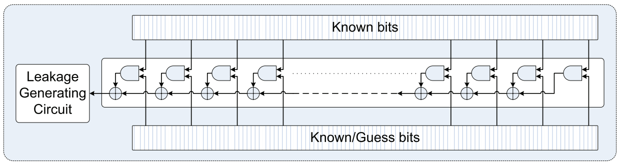


Fig. 4. Diagram of an input-modulated watermark that consists of a combinational function and a leakage circuit

1) Embedding an input-modulated watermark: As shown in Fig. 4, the watermarking-logic consists of two parts, a combinational function and a leakage circuit. The combinational function uses some known input bits to compute one output bit. This output bit is then transmitted by means of the leakage circuit. The idea of this watermark is that we have an artificial data-dependent power consumption that is engineered by introducing the combinational function. The owner of the watermark knows the implemented function and which bits were used as inputs and can use this knowledge to perform a differential power analysis. If the watermark is embedded in the IC, then this differential power analysis will be successful, while it should not be successful if no watermark or a watermark with a different combinational function or different input bits is used.

To be able to implement this type of watermark some bits of the IP core need to be known by the verifier and these bits need to vary for different measurements. They do not necessarily need to be direct inputs or outputs of the IP core but can also be determined by internal states as long as the verifier is able to determine these bits for a given measurement. By using internal states as “known bits” a systematic analysis with chosen inputs can be prevented and the number of possible values used as inputs to the combination function increases. A very easy and straightforward combinational function that was used in the proof of concept implementation of the Trojan side-channel paper is to pairwise combine the input bits with an and conjunction and then compute the $exclusive$-or sum of the outputs of all and operations.

Nevertheless, in practice a more complex function should be used to increase the difficulty of reverse engineering the combinational function given its input and output behavior.

2) Experimental results: As mentioned before, the input-modulated watermark is based on the same idea as the input-modulated side-channel hardware Trojan. In [8] practical results for a hardware Trojan for an FPGA implementation of an AES key schedule were presented where the input to the combinational function was 8 bits of the plaintext and 8 bits of the round key. The input-modulated watermark may use the same circuit and changes only the inputs of the combinational function. We thus omitted to present the experiments and refer the interested reader to the experimental results already given in [8]. For our input-modulated watermark, we would use known values for all 16 input bits of our combinational function and then perform a similar correlation power analysis. If the watermark is not embedded in the IC, then there should not be any significant correlation between the power traces and the expected output of the combinational function.

C. Proof-of-ownership

The watermarks as proposed so far only allow to distinguish whether the watermark is embedded or not, i.e., they only provide a single bit of information. To prove to another party that an IP core was used in a design detecting the watermark is not sufficient, as the ownership of the watermark remains unproven. To provide proof-of-ownership the watermarking scheme needs to be expanded to bind a watermark to a unique identity. This can be achieved by means of digital signatures. In a first step the company generates the hash value of some design ID (e.g., the part number). Then, the private key of the company is used to sign this hash value. The signature can then be transmitted by the watermarks. To allow the proposed designs of the watermarks to transmit information we again employ the initial concept of Trojan side channels. That is, for the spread-spectrum watermark, storing the information to be used for transmission in a circular shift register and XORing the MSB or LSB with the output of the PRNG to generate a specific watermarking sequence. However, an additional shift register of the size of the signature is needed for this solution, which increases the size of the watermark. Another solution to bind the digital signature to the watermark is to use parts or whole of the signature as the initial value of the PRNG. In this case the area overhead of the additional shift register can be omitted. An input-modulated watermark transmitting a signature can be designed by storing the signature in internal registers and subsequently feeding it bytewise as inputs to the combinational function in the same way the input-modulated Trojan side-channel encodes the bytes of the secret key.

The method proposed here prevents attackers from illegally claiming ownership of the used watermark, since the signature can only be generated by the owner of the private key. Note that the modifications discussed in this section also allow to protect the circuit against piracy. For this the signature transmitted by the watermark is stored in a programmable part of the device and will be programmed independently from manufacturing. As the signature is programmed post-manufacturing, it can be generated over a device specific serial number in addition to the part number, so that each device contains a unique watermark. This way only devices programmed by the circuit designer include the correct signature and mere cloning of the semiconductor is not sufficient to build indistinguishable copies of a design. The watermarking schemes previously proposed in scientific literature cannot protect against cloning, as either the watermark signal is visible to everyone [12] or the watermark cannot be programmed after manufacturing [6], [9], [10], [11]. In summary, a semiconductor device should include two watermarks: One that is programmed after manufacturing to protect against piracy and cloning and one that is implemented fully in hardware to detect IP theft.

III. Attacks

In this section we discuss three intuitive approaches to remove a side-channel based watermark: Remove or destroy the circuit implementing the watermark, increase the noise on the side-channel, or transmit an inverse watermarking signal. The later two attacks both aim at reducing the available SNR for detection of the watermark.

A. Reverse-engineering attack

Obviously, if the circuit implementing the side-channel watermark is destroyed or removed from the IP core, both design goals detectability and proof-of-ownership are violated. To be able to destroy or remove the watermark an attacker first needs to identify the corresponding part of the circuit. Note that the attacker does not know whether a watermark is applied to protect a circuit and what kind of watermark is implemented. Therefore, similar to Trojan hardware, watermarks have to be implemented very small and subtle to evade identification during reverse engineering attacks. Furthermore, they should be implemented in a way to be interwoven with the surrounding functional circuit. This can further increase the difficulty to identify the watermark while impeding removal of the circuit without destroying the functionality of the surrounding circuit. Especially the input-modulated watermark can be very small and could be as small as around hundred gates.

The complexity of the reverse engineering attack in practice depends on the design level at which the attacker has access to the protected circuit. If the attacker has access only to the IP core at the post manufacturing level, detecting the watermark will be much more difficult than detecting it in a netlist or even in RTL sources of the design. In this case the attacker would need to first reverse engineer the hardware design to higher abstraction levels for a feasible analysis of its functionality.

Reverse engineering an entire design is usually a very difficult, complex, and thus expensive task. For watermarking purposes the goal of the watermark can be considered achieved if the efforts required to illegally use an unlicensed IP core are equivalent to the efforts necessary to reverse engineer a circuit to the level of fully understanding the design.

B. Raising the noise

How easy a verifier is able to detect a watermark depends on the signal-to-noise ratio (SNR) of the watermarking signal. The lower the SNR, the more (or the longer) measurements are needed to detect the watermark. Increasing the noise of the side-channel will thus reduce the SNR and therefore impedes detection. However, adding additional noise sources results in an increase of power consumption and is thus limited by practical constraints such as battery lifetimes.

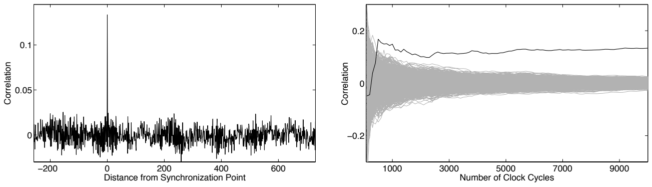
Both watermarks are very robust to this kind of attacks. To detect a spread-spectrum watermark a verifier can lower the effect of the noise by averaging over multiple measurements or by increasing the number of clock cycles covered by the measured trace. For an input-modulated watermark the amount of acquired power traces used during detection is also only limited by the time required to perform the additional measurements. Since the size of the leakage circuit is a design choice of the designer, the SNR and hence the robustness to noise is part of the design space.

C. Transmission of an inverse watermark signal

The idea of our third attack scenario is to hide the signal of the watermark by adding another leakage source that generates leakage in all clock cycles where the watermark itself does not generate any leakage. The idea of this attack is that this inverse signal counterbalances the original watermarking signal and results in a constant power consumption for both signals. We call this introduction of an inverse watermark signal. In theory this makes the detection of the watermarking signal impossible. We will now show that this attack can not be put into practice.

In contrast to the side-channel watermarks proposed in [12] in our design the watermarking signal is hidden well below the noise floor and unknown to an attacker. Without the knowledge of the details of the used PRNG and its initial vector (for a spread spectrum based watermark) or the used combinational function and input bits (for an input-modulated watermark) an attacker can not even compose the inverse signal. Additionally the attacker would need to know the design of the leakage generating circuit to achieve the correct amplitude for the inverse signal. We now consider that the attacker has full access to the details of the used watermark (which is very unlikely) and is able to implement an inverse copy of the watermarking circuit. This can still not avoid detection of the watermarking signal. Process variations during manufacturing as well as slight changes in internal capacitances will result in subtle differences in the power consumptions of the watermark and the inverse watermark. Also very small delays of the clock signal result in inaccurate alignments of the two signals. This makes detection of the proposed watermarking signals possible in practice even with the presence of an inverted watermarking signal.

To experimentally demonstrate this we implemented the inverse-signal transmission attack in our FPGA implementation of the spread spectrum based watermark. We added an exact copy of the watermarking circuit to our design that consists of the same leakage circuit and the same PRNG with the only exception that the output of the PRNG is inverted. The results of this attack can be seen in Fig. 5.



(a) (b)

Fig. 5. Detection of a spread spectrum based watermark that was counter-balanced by an inverse watermarking signal. We performed the same analysis as described in II-A2 while the AES core was idle (a) using the leakage of 10 000 clock cycles, (b) over the number of clock cycles.

We have tested this configuration when the AES core was idle. It turned out that the watermark was still clearly visible with as few as 10 000 clock cycles. The correlation coefficient decreased and thus the attack led to a 10 fold increase of the required number of covered clock cycles. However, although both, the watermarking circuit and the inverse one, use equal building blocks of the FPGA, detecting the watermark is still possible because the power consumption of the two circuits are not exactly inverse due to the different routing of both parts. In practice this attack is equivalent to reducing the SNR of the watermark and cannot prevent detection of the watermark due to the arguments given in the previous section.

IV. Conclusion

In this paper we introduced new watermarking techniques for integrated circuits which employ side-channels as building blocks. The advantage of the introduced watermarks is that they can easily and reliably be detected in physical implementations by means of a side-channel analysis. Our proposed watermarks do not alter the functionality of the original IP core, and their overhead is very small and can be neglected in bigger designs. We gave first experimental results that demonstrate the feasibility of our approach and show its robustness against typical attacks. Our discussion explained that removal of the watermark signal is very difficult due to its hidden nature and the stochastic nature of the applied detection methods. The fact that our watermarks are hidden is also interesting from a system perspective as it allows to embed these watermarks completely unnoticed. If an attacker can not tell whether there is an embedded watermark in a device, it is an even more challenging task to remove it. The proposed scheme additionally allows to embed multiple independent watermarks within a single circuit. Our schemes can be used to counter both IP theft and piracy. It can be implemented in a programmable way to transmit a company specific digital signature information.

References

[1] Side-channel attack standard evaluation board (sasebo). Further information are available via http://www.rcis.aist.go.jp/special/SASEBO/index-en.html.

[2] A. T. Abdel-Hamid, S. Tahar, and E. M. Aboulhamid. IP Watermarking Techniques: Survey and Comparison. In *System-on-Chip for Real-Time Applications - IWSOC 2003*, page 60. IEEE Computer Society, 2003.

[3] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon. Secure Spread Spectrum Watermarking for Multimedia. *IEEE Transactions on Image Processing*, 6(12):1673–1687, 1997.

[4] J. Guajardo, T. Güneysu, S. S. Kumar, and C. Paar. Secure IP-Block Distribution for Hardware Devices. In *Hardware-Oriented Security and Trust - HOST 2009*, pages 82–89. IEEE Computer Society, 2009.

[5] A. B. Kahng, D. Kirovski, S. Mantik, M. Potkonjak, and J. L. Wong. Copy Detection for Intellectual Property Protection of VLSI Designs. In *International Conference on Computer-Aided Design - ICCAD 99*, pages 600–605. IEEE, 1999.

[6] A. B. Kahng, J. Lach, W. H. Mangione-Smith, S. Mantik, I. L. Markov, M. Potkonjak, P. Tucker, H. Wang, and G. Wolfe. Watermarking Techniques for Intellectual Property Protection. In *Design Automation Conference - DAC 98*, pages 776–781. ACM, 1998.

[7] T. Kean, D. McLaren, and C. Marsh. Verifying the authenticity of chip designs with the designtag system. In *Hardware-Oriented Security and Trust - HOST 2008*, pages 59–64. IEEE Computer Society, 2008.

[8] L. Lin, M. Kasper, T. Güneysu, C. Paar, and W. Burleson. Trojan Side-Channels: Lightweight Hardware Trojans through Side-Channel Engineering. In *Cryptographic Hardware and Embedded Systems - CHES 2009*, volume 5747 of *LNCS*, pages 382–395. Springer, 2009.

[9] N. Narayan, R. D. Newbould, J. D. Carothers, J. J. Rodriguez, and W. Timothy Holman. IP Protection for VLSI Designs Via Watermarking of Routes. In *International ASIC/SOC Conference*, pages 406–410. IEEE, 2001.

[10] A. L. Oliveira. Techniques for the Creation of Digital Watermarks in Sequentialcircuit Designs. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 20(9):1101–1117, 2001.

[11] I. Torunoglu and E. Charbon. Watermarking-Based Copyright Protection of Sequential Functions. *IEEE Journal of Solid-State Circuits*, 35(3):434–440, 2000.

[12] D. Ziener and J. Teich. Power Signature Watermarking of IP Cores for FPGAs. *Signal Processing Systems*, 51(1):123–136, 2008.