A Survey on IP Watermarking Techniques

Amr T. Abdel-Hamid[1], Sofiène Tahar[1],
and El Mostapha Aboulhamid[2]

[1] Electrical and Computer Engineering Department
Concordia University, Montreal, Canada
Email: {at_abdel, tahar}@ece.concordia.ca

[2] Dep. d'informatique et de recherche operationnelle
Université de Montréal, Montreal, Canada
Email: aboulham@iro.umontreal.ca

Abstract

Intellectual property (IP) block reuse is essential for facilitating the design process of system-on-a-chip. Sharing IP designs poses significant high security risks. Recently, digital watermarking emerged as a candidate solution for copyright protection of IP blocks. In this paper, we survey and classify different techniques used for watermarking IP designs. To this end, we defined several evaluation criteria, which can also be used as a benchmark for new IP watermarking developments. Furthermore, we established a comprehensive set of requirements for future IP watermarking techniques.

Keywords: Intellectual Property Protection, IP Watermarking, System-on-a-Chip, Digital Watermarking, Copyright Protection.

1 Introduction

Incremental changes to current design methodologies are inadequate for enabling full potential System-on-a-chip (SOC) implementation. The wide availability of reusable virtual components or intellectual property blocks (IPs) are most effective when it comes to reducing cost and development time of SOC designs. Sharing IP designs poses significant high security risks. Most of these IPs need time and effort to be designed and verified, yet they can be easily copied, or modified to cover the authorship proof. Creators and owners of IP designs want assurances that their content will not be illegally redistributed, and consumers want assurances that the content they buy is legitimate. *Watermarking* techniques were widely used throughout history, for copyright protection as well as data hiding. IP watermarking was introduced as a candidate to protect this sensitive copyright information.

IP blocks are delivered in three main flavors depending on price, applications, and contracts between companies. The Virtual Socket Interface (VSI) architecture document [44] describes such levels as:

Soft IP: are delivered in the form of synthesizable hardware design language (HDL) code. They have the advantage of being more flexible and the disadvantage of not being as predictable in terms of performance (i.e., timing, area, power). Soft IPs typically have increased intellectual property risks because RTL (register transfer level) source code is required by the integrator.

Firm IP: are optimized in structure and topology for performance and area through floor planning/placement, possibly using a generic technology library. Firm IPs offer a compromise between soft and hard. More flexible and portable than hard, yet more predictive of performance and area than soft. Firm IPs include a combination of synthesizable RTL, reference technology library, detailed floor-plan, and a full or partial netlist. Firm IPs do not include routing. Risks are equivalent to those of soft IPs if RTL is included and are less if it is not.

Hard IP: are optimized for power, size, or performance and mapped to a specific technology. Examples include netlists that are fully placed, and routed, or optimized custom physical layout. They have the advantage of being much more predictable, but consequently are less flexible and portable due to process dependencies. Hard IPs require, at a minimum, a high level behavioral model, a test list, full physical and timing models along with the final layout. The ability to protect hard IPs is much better because of copyright facilities and there is no requirement for an RTL code.

The VSI Alliance IP protection development working group [15] identifies three main approaches to secure IPs. First, a *deterrent* approach, where the owner uses legal means trying to stop attempts for illegal distribution, i.e., using patents, copyrights and trade secrets. Second, a *protection* approach, where the owner tries to prevent the unauthorized usage of the IP physically by license agreements and encryption. Protection techniques, mostly based on model encryption [41, 43], or distributed environment [9, 10], fall short in securing designs or track them in case they are stolen or reused without permission. For such reasons, a third *detection* approach was introduced, where the owner detects and traces both legal and illegal usages of the designs as in watermarking or fingerprinting. This tracking should be strong enough to be considered as evidence in front of a court if needed. The VSI alliance proposed the usage of the three approaches for proper protection of IP designs.

In this paper, we outline IP watermarking and fingerprinting techniques for copyright protection, surveying the current state-of-art of IP digital watermarking research. In order to evaluate the described techniques, we also defined several evaluation criteria. Finally, we highlight the main technical problems that must be solved before digital watermarking can be widely used. The rest of the paper is organized as follows: Section 2 describes briefly the preliminaries of digital watermarking that we found important to help with reading the rest of the paper. In Section 3 , we introduce the evaluation criteria developed, as well as different attack classes that a watermarking technique might face. Section 4 overviews the state-of-the-art of approaches used for IP watermarking. It describes the main advantages and disadvantages of each technique, trying to evaluate them using the criteria defined above. Finally, Section 5 extracts the main guidelines and conclusions for future IP watermarking research directions.

2 Digital Watermarking

Petitcolas *et al.* [30] defined the term *Steganography* as "*having a covert communication between two parties whose existence is unknown to a possible attacker*". Steganography is divided into three main application classes [30], (1) *information hiding*, which utilizes the secrecy and undetectability of steganography to transfer secret data, used mainly for espionage applications, (2) *content verification* applications (authentication), where a fragile watermark is introduced to secure the contents integrity; and (3) *intellectual property* protection applications, where the watermark is mainly used to convey the information about content ownership and intellectual property rights.

Copyright marking (widely known as *watermarking*), as opposed to steganography, has the additional requirement of robustness against possible attacks. Robust watermarking has the property of being infeasible to remove them or make them useless without destroying the object at the same time. This means that usually it has to be embedded in the most perceptually significant components of the object [30].

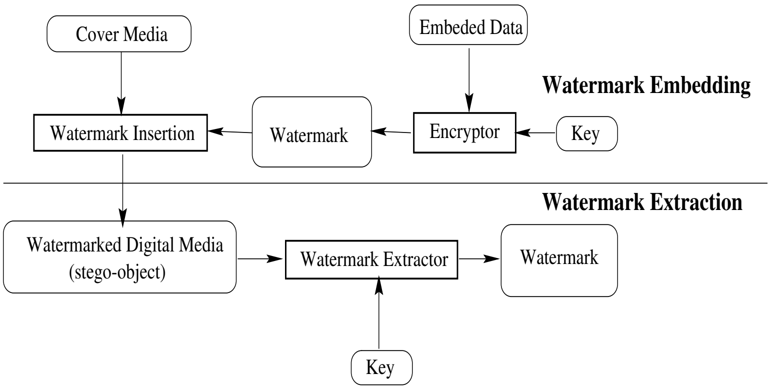

Figure 1: Digital Watermarking

Figure 1 describes a generic model of watermarking [4]. The process is divided into two parts: *watermarking embedding*, and *watermark extraction* (also known as *tagging* and *tracking*, respectively). In the embedding phase, the embedded data, which is the message that one wishes to send secretly, is usually hidden in another media referred to as a cover-text, or cover-image (in our case cover-code or cover-media). This produces the stego-text or other *stego-object*. A key (*stego-key*) is used to control the hiding process, thus restricting detection and/or recovery of the embedded data to parties who know it (or who know some derived key value). This stego-key can be either a public key or a private key depending on the scheme of the watermarking. In the extraction phase, the stego-object is used with the key to extract the watermark and identifies it.

Simmons [38] developed a mathematical representation for watermarking problems based on what he called the "prisoners' problem". Simmons described the watermark as a secret low bandwidth channel (called *subliminal channel*) between two prisoners, who are trying to communicate although they have been locked in a widely apart cells. The only means of communication between these prisoners was this secret channel passing through the wardens. Simmons [38] has defined such channel as a way for secret data transfer. Yet, attacking this subliminal channel might pose limit destruction to the system under investigation.

Subliminal channels provide a good secret communication channel, but, without a proper robustness criteria, they cannot be used in copyright protection. This difficulty led to the introduction, of the *supraliminal channel* by Cravar [8] as a better way for authorship protection. A supraliminal channel is defined as "*a low bandwidth channel that the intruder cannot afford to modify as it uses the most significant components of the object as a means of transition*". It becomes in practice impossible for the intruder to alter the message as he/she must either allow the message through or censor it [8]. The effect of this technique is to turn an *active warden*, who tries to remove the watermark with different attacks, to a *passive warden*, who cannot change or delete the watermark.

3 IP Watermarking Evaluation Criteria

Petitcolas [31] identified a set of measures for watermark evaluation. Although these measures were developed mainly for multimedia applications, we find some of them to be essential while evaluating any IP watermarking techniques. Based on these points and the specific needs of hardware and SOC design, we defined a set of requirements, which any IP watermarking approach should satisfy:

1. **Relying on the secrecy of the algorithm**. According to one of the oldest defined security rules, defined by Kerckhoffs [20] in 1883, any encryption or security technique should not rely on the secrecy of the algorithm, but to the mathematical complexity of such algorithm, "*The system must not require secrecy and can be stolen by the enemy without causing trouble*". The approach should not depend on the secrecy of neither the watermarking insertion nor extraction algorithms. The algorithm should instead depend on one of the system properties to protect the authorship data.

2. **Level of reliability**. This is a very important measure, which can be divided into two main aspects: (1) *robustness*, which measures the strength of the hidden mark against attacks, and the percentage of

undetected watermarked design that might appear; and (2) *false positive*, which occurs whenever the detector could find a mark in a non-watermarked design. Both measures are related to attack analysis and will be discussed in details in the next subsection.

3. **Affecting the design functionality**. Testing and verification of hardware systems is an extremely complicated task. In order to introduce a watermark to the system, the watermarking technique should be totally sound in the sense of its effect on the system behavior. Watermarking techniques should prove their soundness against such a criteria, preferably by proving it mathematically.

4. **Preventing intruder from re-embedding another watermark**. As a passive technique, one of the main challenges of watermarking schemes is the authenticity of the watermark. Scheme designers need to find techniques to protect their designs from intruders who may try to embed another watermark in the design at least to destroy watermark authenticity in front of a court.

5. **Embedding enough data to identify ownership**. The watermarking scheme should add enough data to identify the owner of the design. This data should be concrete enough to be considered as an evidence in front of a court. Nevertheless, the data size should be small enough to neither impose a high overhead on the design size nor to affect the design performance. The amount of data embedded is one of the measures used to differentiate between different IP watermarking techniques.

6. **Implementation overhead**. Watermarking a design is a complementary process to increase its competitiveness but affecting the design performance or having a high overhead in the insertion process would be considered a real drawback. For IP watermarking, we will consider the area, power and delay overheads compared to the original design without watermarking.

7. **Detection and tracking**. Watermark insertion is only half the process, tracking and detection is the second important aspect in any watermarking technique. Tracking and detecting the watermark or its traces after possible attacks is essential. This will be considered as one of the main aspects for judging watermarking techniques.

8. **Asymmetry**. Since Diffe and Helman [11] presented their public encryption scheme, public techniques have proven their strength especially in non-secure environments. Sharing IP designs poses the same threats as other secret data in the public domain. Third parties, such as brokers and sub-contractors, need to know the watermark key for tracking purposes. But these parties are not considered secure entities. Leakage and stealing IPs can still happen through in-house workers, who may know the watermarking key. Asymmetric watermarking is still considered a challenge in many media domains. Deleting watermarks and attacking it is mostly related to the knowledge of its presence and where it might be located.

3.1 Attacks Analysis

Digital watermarking attacks are categorized in four main classes [4]: *unauthorized removal*, *unauthorized embedding*, *unauthorized detection*, and *system attacks*. The same categorization applies for IP watermarking schemes. System attacks aim at attacking the concept of watermarking itself, such as attacking the cryptographic base of the watermarking, or removing the chip that checks the watermark physically in case of video media for instance. This kind of attacks cannot to be avoided by the watermarking schemes. The VSI Alliance IP protection scheme solves this by protecting the design through different transactions.

3.1.1 Masking and Removal Probabilities

Removal attacks [4] aim at the removal of the watermark information. This is tried without breaking the watermark, i.e., without searching for the key used in the embedding. Removal attacks are divided into either *elimination attacks* or *masking attacks*. The intruder tries to eliminate the watermark completely in the elimination attacks. As an example, the intruder tries to estimate the watermark and subtract it from the watermarked design. On the other hand, masking attacks do not aim at removing the watermark itself, but aim at distorting the watermark detector such that it will not be able to sense the availability of the watermark.

The robustness of a watermark is measured through benchmarks in multimedia domain, e.g., Stirmark [29] for images. Given the different nature of IP designs, benchmarking IP watermarking schemes is harder than that of multimedia applications. IP watermarking schemes rely on probabilities instead to prove the strength and robustness of their approaches. In the subsequent text, we will define and use a set of probabilities that will help us to measure the watermark robustness. The main probabilities will be considered mainly to measure masking, removal, and false-positives. Finally, because those measures are directly related to robustness, asymmetry or public-key operating mode will be addressed as well. We used the same set of probabilities to calculate public-robustness, under an extra assumption stating that the watermark-key is known beforehand.

We define the probability of masking (P_m) as "*the probability that any attack would change or delete enough information to cover the watermark without deteriorating the design under investigation*". This probability might change depending on the way of watermark detection as well as the usage of secret or public organizations of our scheme.

Deleting a part of the added signature might mask the watermark, yet, the watermark traces the exist in the system can be detected using other techniques and used in front of court. This means that sometimes, the intruder needs to delete most of the watermark without deteriorating the design under investigation. Thus, the removal probability of the watermark (P_r) can be defined as "*the probability that any attack would delete the whole signature without deteriorating the design under investigation*". Again this probability depends on the way of operation (symmetric or asymmetric), and will be discussed accordingly.

Asymmetric techniques in general are not considered as robust by the data hiding society [31]. In our case, the main measure of our approach capability to work in an asymmetric (or public-key) mode is P_r^a or P_m^a. Depending on such measure, the system cannot work in the public mode unless these probabilities are smaller than a certain value defined by the designer. It is worth to be noted that a second secret watermark can be added to the system in case the intruder could break the public one. This will add some extra overhead on the system, but will be rewarded by a higher level of security.

3.1.2 Probability of Coincidence

The authenticity of the watermark, or the probability to find the watermark by coincidence in a non-watermarked design (false positives), is measured by the *probability of coincidence*. This probability is considered as a measure for detecting the watermark in a design by accident in a non-watermark design.

In [42], the probability of coincidence (P_u) was defined as "*the odds that an unintended watermark is detected in a design*". It is also considered as a measure for ghost attacks (discussed below). This probability will be calculated for different approaches and used as a measure of watermark validity.

3.1.3 Embedding Attacks (Forging)

Embedding attacks aims at embedding another watermark in the design. This can be done by ghost searching, where the intruder tries to find a ghost watermark and consider it as his watermark. This is directly equal to the probability of finding the false positives discussed earlier. Since watermarking is a static technique, we think that such a problem can be solved by using a secure third party, e.g., a watermarking governing body. This governing body will be responsible for generating and distributing time-stamped authenticated signatures, as well as keeping a record for such signatures for the extraction phase. Figure 2 summarizes the above approach, where the secure third part will use the ownership information provided by the IP designer and encrypts it using any public/private-key encryption algorithm after time-stamping it. The encrypted information is then hashed, hence giving a short digest to decrease the watermark embedding overhead. This digest is computationally infeasible to find another message that hashes the same value. Such third entities is used in other types of media in a similar way to secure the introduced watermark.

4 IP Watermarking: State-of-the-art

Many IP watermarking or fingerprinting techniques can be found in the open literature. IP watermarking techniques can be classified into two main classes: (1) *dynamic watermarking*, where the watermark cannot be detected except by running the watermarked IP to detect the generated signal, such as digital signal processing (DSP) or finite state machine (FSM) watermarking; and (2) *static watermarking*, where the watermark is considered a property of the design, and can only be detected by different static techniques, such as route and placement watermarking.

In the next subsections, we are going to discuss the state-of-the-art of different hardware IP watermarking techniques showing the advantages and

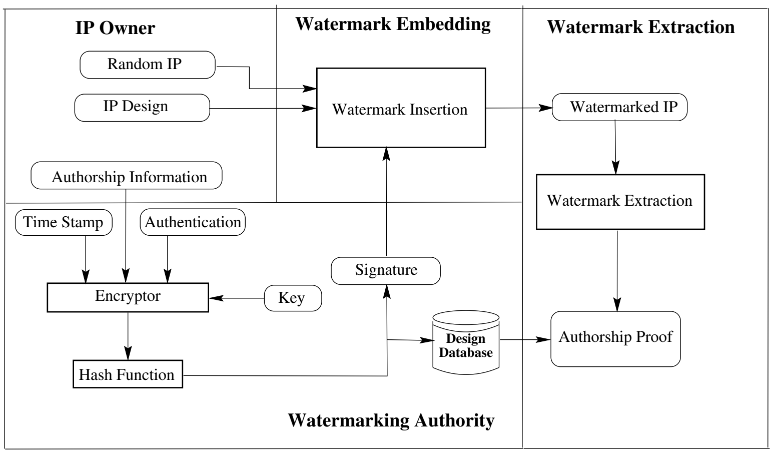

Figure 2: IP Watermarking using a Third Entity as a Governing Body

disadvantages of each technique according to the evaluation criteria discussed above.

4.1 Dynamic Watermarking Techniques

4.1.1 Test Sequence Watermarking

In [12], Fan *et al.*, proposed a technique for securing IPs using the random test sequences. After integrating the IP into the SOC, test signals have to be traceable. Using this fact, the authors combined this test sequence with the watermark generating circuit. This is done by integrating a watermark generating circuit in the on-chip test module, so that whenever the design gets into test mode, the watermark will be generated automatically. The authors [12] proposed different ways to integrate the watermarking circuit into the on-chip test circuit, bits generated from this sequence can be either embedded as an extra bit for each test sequence, or the whole watermark can be generated directly at the beginning or at the end. According to the watermark information, the IP provider is able to verify the ownership rights and does not need to examine the photomicrograph.

The approach is pretty novel, but examining the approach against the evaluation criteria discussed above shows:

1. The approach does not watermark the IP, but mainly the test circuit. The first attack that any intruder can think of is deleting the test circuit

and adding his/her own. This will not affect the performance of the design by any means. This means that in order to keep the watermark secure, we need to keep the algorithm's secrecy, i.e., Kerckhoffs' rule.

2. Form the above, it follows that the approach has real robustness flow, it cannot be used of course in an asymmetric mode. The authors did not develop enough evidence to make us think otherwise. The calculation of removal, and masking probabilities is not possible, because the design is not touched, and deleting the watermark would be simple and direct.

3. The approach, on the other hand, can be used effectively as a complementary technique to generate the watermark inputs for other dynamic techniques. This can be done especially because adding such data will have minimal overhead on the system, where any part of the testing circuits can be used.

4.1.2 Digital Signal Processing Watermarking

Digital signal processing (DSP) watermarking is introduced in [3] and [36] at the algorithmic level of the design flow. The main idea of both approaches introduced is based on the ability of designers to make minor changes in the decibel (db) requirements of filters, without affecting their operation.

In [3], the designer of a high level digital filter should encode one character (7 bits) as his/her hidden watermark data. Then the high level filter design is divided into seven partitions where each partition is used as a modulation signal of one of the bits. This means dividing the filter into seven parts and use each part as a carrier signal with little db change if the bit is one or no db change if the bit it zero.

In [36], the authors divided the problem into two parts. They have introduced the watermark to both algorithmic and architectural levels, in order to achieve more robustness. At the algorithmic level, they have introduced a similar approach as [3], where seven bits are added. Yet, at the architectural level, they have used a static approach in order to watermark the transpose of the finite impulse response (FIR) filter [36]. Their approach is based on using different structures of the filter building block according to the bits needed to be embedded.

Both approaches have a low embedding overhead, as well as a low design overhead. However, both are easily detectable as long as the DSP filter is not covered totaly in the design. Both approaches can be used at the DSP algorithmic level, which is a pretty high level of the design flow. The authors did not discuss the strength of their approach or any robustness criteria. Besdies, the approaches depend on a very low data rate, just one character (7 bits), which makes them really unpractical to be used in an industrial environment. With such low bit rate, the watermark is extremely sensitive to design fluctuations at the lower levels, like temperature or any other changes in the environment that might causes high false positive rates or missing the watermarked algorithm. Also, such a watermark is extremely sensitive to masking attacks as the smallest changes in the filter function would mask or even remove the watermark.

4.1.3 Watermarking Finite State Machines

At the behavioral level, Oliveira [28], and Torunoglu *et al.* [42] introduced two different techniques used in the watermarking of sequential parts of the design. Both algorithms are based on adding new input/output sequences at the finite state machine (FSM) representation of the design. The main advantage of both approaches is the ability to detect the presence of the watermark at all lower design levels.

FSM Watermarking Based on Unused Transitions: In [42], Torunoglu and Charbon introduced the first IP protection approach through FSM watermarking. The algorithm is mainly based on extracting the unused transitions in a state transition graph (STG) of the behavioral model. These unused transitions are inserted in the STG and associated with a new defined input/output sequence, which will act as the watermark.

The approach in [42] starts with building the FSM representation of the sequential design, then visiting every state and finding the unused state transitions (input/output pairs). In case the FSM is completely specified (CS-FSM), new input/output pairs are added to expand the FSM. The minimum number of transitions needed (n_{min}) is then calculated, and compared to the maximum number of free transitions (n_{max}) to satisfy the probability (P_u) that a non-watermarked design would carry this watermark by coincidence. If this probability cannot be satisfied, input/output pairs should be added to satisfy the watermark requirements.

The input/output sequence is calculated, such that the input sequence is random to the set of unused transition inputs. On the other hand, the output, which is the hidden information, is encrypted using a key (K). Extra transitions are added such that the output of the given input sequence generates the encrypted hidden data, i.e., one should have both the key and the input sequence to be able to read the watermark.

Figure 3 [42] shows an example of the watermarking process, where Figure 3 (a) shows the original design, Figure 3 (b) describes the watermarked design, and Figure 3 (c) shows another watermarked solution after augmenting the inputs to add more transitions.

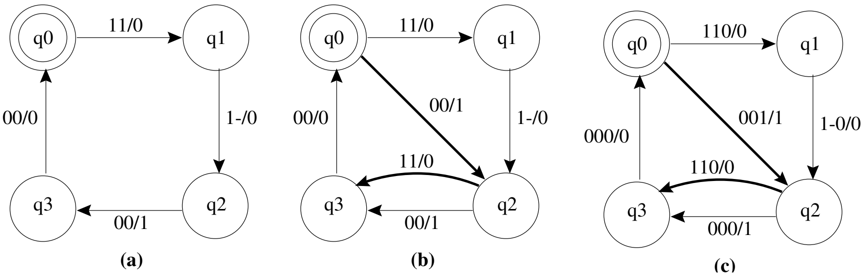

Figure 3: FSM Watermarking Utilizing Unused Transitions

The approach works at a high level of the design flow, which provides extra strength, and does not depend on the secrecy of the algorithm as a way of securing the design. The algorithm can be detected at mostly all lower design levels, sometimes even after design manufacturing. The authors, however, used the probability of coincidence as the only measure for robustness, which only covered the false-positives case. To evaluate the approach using the proposed evaluation criteria, we had used the values generated by the authors in [42] to calculate both the masking probability (P_m^u) and the removal probability (P_r^u) for the IWLS93 [25] benchmark set shown in Table 1. Considering all transitions have equal occurrence probabilities, the masking probability (P_m^u) can be calculated as "*the probability that any attack would delete at least one transition in order to cover the watermark without affecting any of the original design transitions*". Hence, P_m^u was calculated as follows:

$$P_m^u = \frac{n_{min}}{n + n_{min}}$$

Furthermore, the removal probability (P_r^u) is calculated as "*the probability that any attack would delete all added watermark transitions without affecting any of the original design transitions*". Hence, P_r^u can be calculated as follows:

$$P_r^u = \frac{1}{C_{n+n_{min}}^{n_{min}}}$$

where $C_{n+n_{min}}^{n_{min}}$ is the combination of n_{min} and $n + n_{min}$.

Table 1: IWLS93 Benchmark Results using Unused Transitions Algorithm

Circuit	♯I[O]	♯T[S]	$(I/O)_{wm}$	n_{min}	P_u	P_m^u	P_r^u	*Area%*
S27	4[1]	34[6]	1/3	9	1.4e-11	0.2	1.4e-9	143
BBARA	4[2]	60[10]	1/1	10	9.3e-10	0.14	2.2e-12	74
DK14	3[5]	56[7]	1/0	7	2.9e-11	.095	1.8e-9	24
EX1	9[19]	138[20]	0/0	4	1.3e-23	0.028	6.7e-8	3.2
EX1	9[19]	138[20]	0/0	2	3.6e-12	0.014	1.02e-4	0.6
STYR	9[10]	166[30]	1/0	4	9.1e-13	0.023	2.9e-8	22
SCF	27[56]	166[121]	0/0	2	1.9e-34	0.011	7.1e-5	0.2

The measures provided in Table 1 showed how vulnerable the algorithm is for masking attacks. Masking attacks do not delete the whole watermark, yet they cover the authorship information, which means that the direct detection method proposed by the authors is not reliable enough and exhaustive search or the Genome search [42] will be the main watermark extraction method. Besides, the removal probability (P_r^s), although higher, is not high enough in some cases to consider the system totaly secure, such as the case of SCF (Table 1). On the other hand, the overhead starts high as expected with small designs, yet it becomes negligible when designs get larger. The only problem in this is that the authors did not rely on a fixed length signature, which would raise questions about the amount of information embedded especially in the large designs, for instance embedding only 40 bits in STYR (Table 1).

Finally, finding the input sequence that satisfies P_u and is not considered with a high overhead on the STG is an NP-hard problem [28]. The authors of [42] proposed exhaustive search, or Monte-Carlo search [2] to get over this, yet solving this would propose a high overhead in the design phase. The algorithm is immune for FSM reduction techniques, as the variables used are usually part of other transitions, which makes it real hard to remove.

FSM Watermarking by Property Implanting: The other FSM watermarking approach available was proposed by Oliveira [28]. The author tried to manipulate implicitly the STG of the finite state machine to implant the watermark as a property in the new one.

To watermark a design as proposed in [28], the user should define an arbitrary long string that clearly describes his/her ownership rights. This data is considered as the watermark information. After encrypting this message using a public key, the user should then use a one-way hash function, such as MD5 [37], to obtain a compact signature of this arbitrarily long sentence. The arbitrary sequence is then broken to input sequence combinations. For example, if the design has 16 inputs, and the sequence is 128 bits, it defines a unique sequence of 8 input combinations.

The user then changes the STG in such a way that the sequence of states reached by this sequence of inputs exhibits a specific property, which is rare in non-modified STGs. This property is purely topological and does not depend on the specific encoding. If, later on, the watermark needs to be uncovered, the designer provides this input sequence and the property he/she defined.

In order to define the input sequence to change the STG properties, Oliveira [28] adds extra states and transitions in a systematic way to satisfy this property. The algorithm has a low overhead on the design flow, because it does not need to go through the FSM to find the unused transitions (an NP-hard problem as discussed above). In [28], it is even proposed to use a very strong way to build and implant the watermark without the need of building the FSM of the design, i.e., low building overhead.

The author in [28] used a 128 bit signature, which is large enough to identify different users. The approach depends on adding a counter that checks for the input sequence expected and reaches a certain value to indicate that the design has traversed the implanted watermark. This counter can be a real weak point when it comes to masking attacks, as deleting the counter or changing it means destroying the whole watermark. Also, the counter, and the way the property is added should be secret in order to insure proper security, Kerckhoffs' secrecy law. The author in [28] did not show how the probability of false positives are calculated, yet he mentioned that he calculated and found that only 2 designs from the whole IWLS93 test bench might have higher probability of false-positives larger than zero.

Furthermore, the extra states added can be removed sometimes using a state reduction approach, the author in [28] considered this to be hard to achieve in large designs, which is true because of the state-explosion problem that would arise. Also, he proposed to solve this problem by slightly changing the functionality of the STG. This is hard to be done mechanically as it is pretty complicated and might affect the design functionality.

4.2 Static Watermarking Techniques: Constraint-Based IP Watermarking

In [16, 17, 18, 19], Kahng *el al.* proposed a *constraint-based* IP watermarking approach, which is a generic algorithm that can be used at different levels of the design flow. The approach is based on the usage of available tools used mainly to solve NP-hard problems. The algorithm adds extra constraints to such solution, yielding to the new watermarked design. The approach is based on a generic optimizer and constraint-satisfaction (SAT) problems [17]. The watermarking tool proposed is mainly composed of the following parts (Figure 4):

1) An optimization problem, which is an NP-hard problem that needs constraints and heuristics to be solved.

2) An off-the-shelf optimization software/algorithm to solve such a problem.

3) A set of constraints that should be applied to the design.

4) A well-formed grammar to add extra-constraints to the previous ones for building the required watermarked design. This is the main watermarking tool, it is composed of a one-way encryption functions that converts the watermark of the code to a set of well-formed constraints.

The watermark is presented to the constraint generator. In the generator, the watermark is first encrypted, then transferred through a hash function (to shorten its length). Finally, it is converted to a set of extra constraints, through the well-formed grammar, forming a new set of constraints which is added to the available ones. Both the design and the set of constraints are fed to the black-box optimizer resulting in a watermarked solution. The watermark is then a set of extra constraints that will limit the set of possible solutions to a smaller set. The watermark becomes stronger as the "*watermark subset*" is smaller.

Kahng *el al.* [17] illustrated this approach using the a simple satisfiability

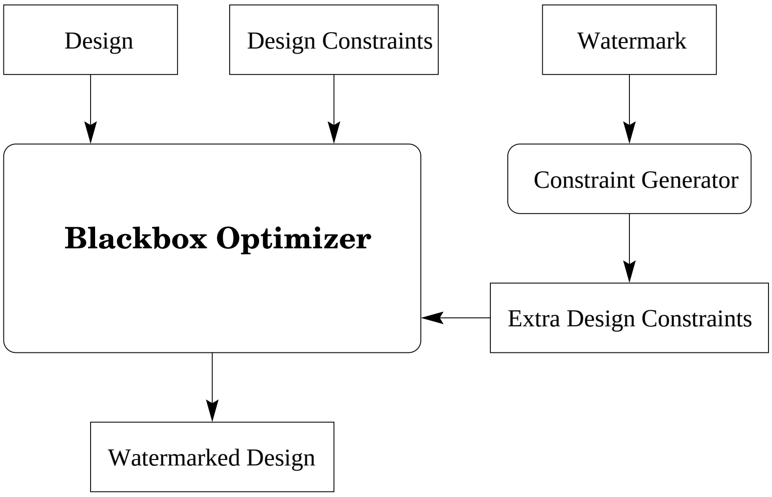

Figure 4: Constraint-Based IP Watermarking

(SAT) problem [13]. Many problems in hardware design are modeled as a classical NP-complete constraint-satisfaction problem. For instance, let SAT (U, C) be a finite set of variables U and a collection $C = \{c_1, c_2, ..., c_n\}$ of clauses over U. The SAT problem relies on finding the set of all satisfying assignment ("truth assignment") of C that satisfies all the clauses in U. Adding extra constraints to such problem direct the solution to identify uniquely the watermarked solution.

The proposed scheme is the dominant approach for hardware IP watermarking designs, and although we classify it as a static approach yet some of its applications can be dynamic. Due to the generic nature of the approach, it was applied to different levels of the IP design flow. At the system level, for instance, it was used to watermark memory graph coloring problems [14], as well as graph partitioning problems [46] and linear programming problems [26]. At lower design levels, the approach was used even more heavily woth routing [27], placement, and floor planning [19]. In [34], Qu developed the first public watermarking approach based on the above technique. His approach depends on implanting two watermarks, a public one to be seen by everyone, and a private one in case this public one was attacked.

The constraint based technique was also used for fingerprinting by Lach *et al.* [23], and Qu *et al.* [35]. The first algorithm divides the design into parts and applies constraints watermarking to them. Adding a loose set of different constraints on these parts will produce different fingerprinted solutions needed. Qu *et al.* [35] finger printed the design by dividing the solution into two parts. The first part introduces a set of independently relaxable constraints before solving the problem. In the second part, each of these constraints is optimized alone to guarantee many solutions would be achieved. Caldwell *et al.* [1] proposed a third fingerprinting approach based on iterative optimization techniques to generate different solutions for the same SAT problem.

Evaluating such technique is a complicated issue, because of the different derivatives and applications of the technique. We tried to evaluate the main algorithm giving different points that need to be tackled. The approach does not have any secret part covered, i.e., it is not compliant with Kerckhoffs' secrecy rule. Yet, the technique converts the signature to extra constraints. Exposing the well-formed grammar that is used to generate the constraints in any legal dispute, would weaken other watermarked designs. To solve this, constraint generating tools that rely only on the key should be built and verified.

The quality of the watermarked design is another problem that might be caused by the constraints. For instance, some signature constraints might contradict, or at least degrade the quality of the generated solution. To solve this problem, Qu *et al.* [33] proposed the so-called "*fair watermarking*". They used the same approach before, but only embed a part of the signature to keep the quality of the solution. Another attempt was done by Wong *et al.* [45] who proposed three optimization-intense watermarking techniques based in the constraint based approach to watermark decision problems. Decision problems are usually hard to watermark, because the result should be a decision, 'Yes' or 'No', for instance, and not like optimization problems, where we might have many correct solutions. Their watermarking technique tries to embed only a part of the signature that will still keep the authorship proof, yet will not change the decision solution using iterations [45]. The main problem faced by these approaches, is the need of iterations to watermark the design, which might affect the overhead needed to add the watermark.

The idea of injecting the watermark in a non-linear problem by nature gives it high strength. The watermark becomes a property of the design more than added information, which makes it extremely hard to remove or mask. Yet, Le-Van *et al.* [22] showed that several constraint-based watermarking schemes can be broken easier than previously thought. The authors used two different approaches to analyze these schemes. The first approach was directed against graph coloring schemes [14], they did not remove the signature but modified it so that another arbitrary signature could be extracted by resolving the problem with a maximum of two extra colors. In the other scheme, they attacked the FPGA watermarking proposed by Lach *et al.* [24], where they located the embedded signature and then removed it. The authors though, did not try to attack this approach at lower levels such as placement [19] or routing [27]. We believe deleting the watermark at these levels is much harder. The intruder in both techniques considered should possess knowledge of the generated solution, something that is not usual, as the designs are usually sold at lower design levels, and regenerating such a solution would be extremely problematic. One of the techniques that is used to overcome this attack is the so-called "*localized watermarking*" [21], where the signature is divided into a number of small watermarks are randomly augmented in the design. This gives the watermark an extra strength by both, forcing the intruder to resolve each of the smaller NP-harder problems, as well as watermarking different design parts in order to keep the design from partitioning. Finally, The algorithm is missing an efficient tracking technique that would help to track the watermark at different design levels.

4.3 Hierarchical Watermarking

Charbon, and Torunoglu [5, 6, 7] introduced a hierarchical watermarking scheme based on a generic approach that can be used on different design levels. The authors proposed the usage of multiple watermarks in different abstraction levels in order to get a more secure system. The scheme increases watermark robustness as the intruder needs to delete the watermark in different levels. This hierarchy should not pose a high overhead on the design cycle nor on the final watermarked product area or performance, since the authorship information can be divided into many levels. Using such a scheme means that the approaches below are mostly complementary. This means that the designer should try to add his watermark to different levels of the design, hierarchical watermarking, as well as to different modules of the design to insure proper security and robustness to the design.

5 Conclusions

Sharing IP designs poses high security risks. IPs need time and effort to be designed and verified, but they can be easily stolen or forged. Digital watermarking, used with most of the digital shared media, is considered a solution for the copyright protection of IP blocks. It was introduced as a way to protect both the owner and the customer rights against forging or illegal distribution of the IP blocks. In this paper, we have first introduced a set of evaluation criteria then surveyed the current state-of-the-art in IP digital watermarking, and finally compared the different techniques available, discussing major advantages and disadvantages.

IP watermarking schemes still need more development to be integrated in the design cycle. Future IP watermarking schemes should be robust enough to secure design, yet they should not imply a high overhead neither on the design process nor on the final watermarked product. We believe that the different techniques introduced should be use both hierarchically [5] to protect the design at different levels, as well as modularly to protect different parts of the IP design. Adding a hierarchy of watermarks through the design cycle can give a more robust watermark against attacks. Starting form high levels of the design (i.e., system level) and integrating the watermark through many design levels insures robustness, which decreases the risks of destroying the watermark. These watermarks should be easily detectable at lower design levels to insure proper tracking. Efficient watermarking schemes should also use a public-key encryption algorithm in the watermarking process, thus allowing third party entities (such as brokers) to get into the distribution cycle without security hazards. Finally, IP watermarking developers are missing a strong benchmark like those available, e.g., for photos. Such benchmark would be a balanced measure for the strength of different approaches. Benchmarking an IP watermarking scheme is harder than for instance photos as the watermark might be spread in many design levels, given the different nature of the design span of SOCs.

References

[1] A. E. Caldwell, H. J. Choi, A. B. Kahng, S. Mantik, M. Potkonjak, G. Qu, J. L. Wong, “Effective Iterative Techniques for Fingerprinting Design IP”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, February 2004, Vol. 23, No. 2, pp. 208-215.

[2] E.D. Cashwell, and C.J.Everett, “A Practical Manual on the Monte Carlo Method for Random Walk Problems”, Pergamon Press, 1959.

[3] R. Chapman, and T. S. Durrani, “IP Protection of DSP Algorithms for System on Chip Implementation”, IEEE Transactions on Signal Processing, March 2000, Vol. 48, No. 3, pp. 854-861.

[4] I. J. Cox, M. L. Miller, and J. A. Bloom, “Digital Watermarking”, Morgan Kaufmann Publishers, 1998.

[5] E. Charbon, “Hierarchical Watermarking in IC Design”, In Proc. IEEE Custom Integrated Circuits Conference, Santa Clara, California, USA, May 1998, pp. 295 - 298.

[6] E. Charbon, and I. Torunoglu, “Watermarking Layout Topologies”, In Proc. IEEE Asia South-Pacific Design Automation Conference, Hong Kong, January 1999, pp. 213-216.

[7] E. Charbon, and I. Torunoglu, “Intellectual Property Protection Via Hierarchical Watermarking”, In Proc. International Workshop on IP Based Synthesis and System Design, Grenoble, France, December 1998.

[8] S. Cravar, “On Public-key Steganography in the Presence of an Active Warden”, Technical Report RC20931. IBM Research Division, T. J. Watson Research Center, July 1997.

[9] M. Dalpasso, A. Bogliolo, and L. Benini, “Virtual Simulation of Distributed IP-based Designs”, In Proc. ACM/IEEE conference on Design Automation Conference, New Orleans, Louisiana, USA, June 1999, pp. 50-55.

[10] M. Dalpasso, A. Bogliolo, and L. Benini, “Hardware/Software IP Protection”, In Proc. ACM/IEEE Design Automation Conference, Los Angeles, California, USA, June 2000, pp. 593-596.

[11] W. Diffie and M.E. Hellman, “New Directions in Cryptography”, IEEE Transactions on Information Theory, 1976, Vol. 22, No. 6, pp. 644-654.

[12] Y.C. Fan, and H.W. Tsao, “Watermarking for intellectual property protection”, IEE Electronics Letters, September 2003, Vol. 39, No. 18.

[13] M. R. Garey and D. S. Johnson, “Computers and Intractability–A Guide to the Theory of NP-Completeness”, W. H. Freeman And Company, New York, 1979.

[14] I. Hong and M. Potkonjak, “Techniques for Intellectual Property Protection of DSP Designs”, In Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, Munich, Germany, May 1998, Vol. 5, pp. 3133-3136.

[15] Intellectual Property Protection Development Working Group, “Intellectual Property Protection: Schemes, Alternatives and Discussion”, VSI Alliance, White Paper, Version 1.1, August 2001.

[16] A. B. 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, San Jose, California, USA, November 1999, pp. 600-604.

[17] A. B. Kahng, J. Lach, W. H. Mangione-Smith, S. Mantik, I. L. Markov, M. Potkonjak, P. Tucker, H. Wang, and G. Wolfe, “Constraint-Based Watermarking Techniques for Design IP Protection”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, ctober 2001, Vol. 20, No. 10, pp. 1236-1252.

[18] 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 Proc. ACM/IEEE Design Automation Conference, San Francico, California, USA, June 1998, pp. 776-781.

[19] A. B. Kahng, S. Mantik, I. L. Markov, M. Potkonjak, P. Tucker, H. Wang and G. Wolfe, ”Robust IP Watermarking Methodologies for Physical Design”, In Proc. ACM/IEEE Design Automation Conference, San Francico, California, USA, June 1998, pp. 782-787.

[20] A. Kerckhoffs, “La cryptographie militaire”, Journal des sciences militaires, vol. IX, January 1883, pp. 5-38, February 1883, pp. 161-191.

[21] D. Kirovski, M. Potkonjak, “Localized watermarking: Methodology and Application to Template Mapping”, In Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, Istanbul, Turkey, June 2000, Vol. 6, pp. 3235-3238.

[22] T. Le-Van, and Y. Desmedt, “Cryptanalysis of UCLA Watermarking Schemes for Intellectual Property Protection”, In Information Hiding, LNCS 2578, Springer-Verilog, 2002, pp. 213-225.

[23] J. Lach, W.H. Mangione-Smith, M. Potkonjak, “Fingerprinting Techniques for Field-Programmable Gate Array Intellectual Property Protection”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, October 2001, Vol. 20, No. 10, pp. 831-836.

[24] J. Lach, W.H. Mangione-Smith, M. Potkonjak, “Robust FPGA Intellectual Property Protection through Multiple Small Watermarks”, In Proc. IEEE/ACM Design Automation Conference, New Orleans, Louisiana, USA, June 1999pp. 831-836.

[25] K. McElvain, “LGSynth93 Benchmark Set: Version 4.0”, Http://www.cbl.ncsu.edu/pub/Benchmark_dirs/LGSynth93/, 1993.

[26] S. Megerian, M. Drinic, and M. Potkonjak. “Watermarking Integer Linear Programming Solutions”, In Proc. IEEE/ACM Design Automation Conference, New Orleans, Louisiana, USA, pp. 8-13, June 2002.

[27] 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. IEEE International ASIC/SOC Conference, Washington, DC, USA, September 2001, pp. 406-410.

[28] A. L. Oliveira, “Techniques for the Creation of Digital Watermarks in Sequential Circuit Designs”, IEEE Transactions om Computer-Aided Design of Integrated Circuits and Systems, Vol. 20, No. 9, September 2001, pp. 1101-1117.

[29] F. A. P. Petitcolas, R.J. Anderson, and M. G. Kuhn, “Attacks on Copyright Marking Systems”, In Information Hiding, LNCS 1525, Springer-Verlag, 1998, pp. 219-239.

[30] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, “Information Hiding- A Survey”, Proceeding of IEEE, July 1999, Vol. 87, No. 7, pp. 1062-1078.

[31] F. A. P. Petitcolas,“Watermarking Schemes Evaluation”, IEEE Magazine on Signal Processing, Vol. 17, No. 5, September 2000, pp. 58-64.

[32] G. Qu, and K. Potkonjak, “Analysis of Watermarking Techniques for Graph Coloring Problem”, In Proc. IEEE/ACM International Conference on Computer-Aided Design, San Jose, California, USA, November 1998, pp. 190 - 193.

[33] G. Qu, J. L. Wong, and M. Potkonjak, “Fair Watermarking Techniques”, In Proc. of Asian-Pacific Design Automation Conference, Yokohama, Japan, January 2000, pp. 55-60.

[34] G. Qu, “Publicly Detectable Watermarking for Intellectual Property Authentication in VLSI Design”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 21 , No. 11, Nov. 2002, pp. 1363 - 1368

[35] G. Qu, M. Potkonjak, “Fingerprinting Intellectual Property Using Constraint-Addition”, In Proc. IEEE/ACM International Design Automation Conference, Los Angeles, California, USA, June 2000, pp. 587-592.

[36] A. Rashid, J. Asher, W. H. Mangione-Smith, and M. Potkonjak, “Heirarical Watermarking for Protection of DSP Filter Cores”, In Proc. IEEE Custom Integrated Circuits Conferance, Orlando, Florida, USA, May 1999, pp. 39-42.

[37] R. Rivest, “RFC 1321: The MD5 Message-Digest Algorithm”, Network Working Group, 1992.

[38] G. J. Simmons, “The History of Subliminal Channels”, IEEE Journal on Selected Areas in Communications, May 1998, Vol. 16 , No. 4, pp. 452-462.

[39] G. J. Simmons, “Results Concerning the Bandwidth of Subliminal Channels”, IEEE Journal on Selected Areas in Communications, May 1998, Vol. 16 , No. 4 , pp. 463-473.

[40] E. M. Sentovich, K. J. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P. R. Stephan, R. K. Brayton, and A. Sangiovanni-Vincentelli, “SIS: A System for Sequential Circuit Synthesis”, Technical Report, Dept. of Electrical Engineering and Computer Science, University of California, Berkeley, California, USA, 1992.

[41] Synopsys Inc., "Verilog Model Compiler datasheet", http://www.synopsys.com/products/lm/ip/vmc_ds.html, 1999.

[42] I. Torunoglu, and E. Charbon, "Watermarking-Based Copyright Protection of Sequential Functions", IEEE Journal of Solid-State Circuits, February 2000, Vol. 35, No. 3, pp.434-440.

[43] Topdown Inc., TopProtect datasheet, 1999, http://www.topdown.com/topprotect/tp_dsheet.htm.

[44] VSI Alliance, "VSI Alliance Architecture Document: Version 1.0", VSI Alliance, 1997.

[45] J. L. Wong, G. Qu, and M. Potkonjak, "Optimization-Intensive Watermarking Techniques for Decision Problems", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, January 2004, Vol. 23, No. 1, pp 119-125.

[46] G. Wolfe, J. L. Wong, and M. Potkonjak, "Watermarking Graph Partitioning Solutions", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, October 2002, Vol. 21, No. 10, pp 1196-2204.