Watermarking Layout Topologies

Edoardo Charbon and Ilhami Torunoglu

Cadence Design Systems Inc., San Jose, California, 95134

Abstract

Watermarking is a technique currently being developed to effectively protect Intellectual Properties of various types. In this paper a formalization of the watermarking problem is presented in the context of IC physical design. A class of algorithms is proposed for implanting arbitrary codes in the inherent structure of layout topologies. Similarly, a method is given to reconstruct the original watermark for a given design. The concepts of robustness against forgery and theft tracking are analyzed in light of the proposed algorithms. Examples show the suitability of the approach.

1 Introduction

With the explosion of electronic Intellectual Property (IP) commerce, it is becoming critically important to develop mechanisms for effectively protecting IP contents. Currently, designers enforce their copyrights by means of Non-Disclosure Agreements and patents. However, tracking espionage, if at all possible, may involve prohibitive costs. On the other hand, actively preventing IPs from being illegally copied, forged and re-used for a profit may also prove an extremely difficult task.

An alternative approach is theft deterrence. The goal of deterrence is to discourage theft and espionage by making it possible to detect and track IP forgery quickly and effectively. Deterrence can be accomplished by embedding a unique code, or watermark, in the design exploiting its features. Watermarking is a well-known technique, traditionally used in banknotes and other documents to discourage counterfeiting. It usually consists of semi-transparent symbols embedded on paper. Recently, similar concepts have been applied to digital audio-visual IPs. See [1] for a review of the field. Fundamental requirements for A watermark are that it be (1) transparent, i.e. not interfering with the design functionality, (2) robust, i.e. hard to remove or forge, and (3) detectable, i.e. easy to extract from the design. The algorithms used for managing watermarks must not necessarily be proprietary, while the code should be secret for any released IP.

In this paper we present a class of algorithms for synthesizing and detecting watermarks in arbitrary electronic IPs using the features characteristic of physical design. Electronic circuits are generally complex, highly optimized circuits developed in one or more technologies. In order to maximally exploit the advantages of a particular technology, there is little room for overhead, both in form of additional circuits and/or signals. At least two types of watermarking schemes exist. The first scheme, known as active watermarking, consists of integrating the watermark as a part of the design process, thus allowing the creation of an arbitrarily high number of unique watermarks. The second scheme, known as passive watermarking, is aimed at adding the watermark to a design making use of existing structures, thus requiring no redesign but allowing limited tracking flexibility. Both approaches are robust, since the deletion of the watermark results in removing the wanted functionality. Moreover, forgery can be traced to the source, since watermarks at the physical design level are associated with the last licensed buyers who ultimately caused the breach.

A complete protection scheme based on watermarking consists of two phases: watermark synthesis and watermark detection. The synthesis phase is fully characterized by (a) a set of algorithms translating design features onto a unique watermark, (b) $t_r$, the worst-case time required to forge and/or delete the watermark, and (c) $P_u$, the odds that a design carries an unintended watermark in part or in its totality. The detection phase is fully characterized by (d) $P_{\text{tr}}$, the probability of a miss and (e) $P_f = P_w$, the probability of a false alarm. Typical specifications of a complete IP protection scheme could be: $t_r \geq 2$ years, $P_u = 10^{-50}$, $P_{\text{tr}} \leq 10^{-60}$.

In general, watermarks can be designed to be inherently redundant, in order to combat partial forging and/or deletion. Redundancy is designed to boost the confidence in positive watermark identification, by reducing $P_{\text{tr}}$, $P_u$ and $P_f$.

The paper is organized as follows. The problem is formulated in Section 2. Section 3 presents techniques for the characterization of circuit topology. Sections 4 and 5 propose a method to synthesize and detect watermarks from topological information. Section 6 illustrates the method through examples.

2 Problem Formulation

For a finite alphabet $\Sigma$, e.g. $\Sigma = \{0, 1\}$. Let $\Sigma^*$ be the set of all strings in the alphabet. Define watermark set $\mathcal{W} \subseteq \Sigma^*$ as a collection of strings over $\Sigma$ that identify the design. Let us now select string $k \in \Sigma^*$, the key of the identification process, then algorithm $\alpha$ is defined as

$$\alpha_k : \Sigma^* \to \mathcal{W}$$

$$\mathcal{W} = \{\sigma(k)\}$$

(1)

where $\sigma$ is a compact representation or signature of the design and $\mathcal{W}$ its watermark.

Consider the physical implementation of a given design and define its granularity. Call atomic blocks those devices which cannot be further sub-divided and define $\Omega$ as the set of such devices. Define $S$ as the set of all circuit implementations over $\Omega$ and $s \in S$ as one such implementation. In $s$, every atomic block $\omega \in \Omega$ may have multiple instantiation in one design. Assume there exists a compact representation for the design. Let $\mathcal{M}$ be the mapping of the design features onto such representation:

$$\mathcal{M} : S \to \Sigma^*$$

$$\sigma(s) = \mathcal{M}(s)$$

(2)

Note that $\mathcal{M}$ is a non-unique, possibly lossy mapping, while $\sigma(s)$, and consequently $\alpha$, are general in nature.

Let us define $\mathcal{F}$ as a mapping which transforms implementation $s$ onto a new implementation $s'$ as:

$$\mathcal{F} : S \to S$$

$$s' = \mathcal{F}(s)$$

(3)

If $\sigma(s') = \sigma(s)$, then $\mathcal{F}(s)$ is said to be signature-invariant. If $\mathcal{F}(s)$ is signature-invariant over a subset of $\Omega$, it is partially signature-invariant.

3 Circuit Topology

This section outlines a set of convenient mappings $\mathcal{M}$ capable of coding a topology $s$ onto signature $\sigma(s)$ as required by the scheme described in
Section 2. Most layout implementations are associated with a topology. A topology describes the relative position and orientation of any object pairs \((\omega_1, \omega_2) \in \Omega^2\). A mapping \(F\) which does not modify the topology of a design is said to be topology-invariant. If no wiring is present, layout topologies can be represented compactly through sequence pairs \([2]\) if non-slicing or through slicing structures \([3]\) otherwise.

In the presence of wiring, the approach usually adopted is that of partitioning the nets into rectangular structures whose topology coding is computed as suggested in \([4]\). The problem with this approach is its relatively low robustness to net tampering, thus resulting in an increased overall watermarking weakness. To address this problem we propose the use of an alternate wiring representation.

Let us represent the atomic blocks of a layout in terms of primitives called bubbles \([5]\). A bubble is a point associated with a given layer. Let \(B\) be the set of all bubbles in the design. Let us define \(B\) as the mapping which transforms every atomic block \(\omega \in \Omega\) onto an \(m\)-tuple of bubbles \(\beta \in B^m\). Note that \(m\) is a finite natural number, moreover for simplicity but without loss of generality, suppose that \(m\) is constant over \(\Omega\).

\[
B : \Omega \rightarrow B^m
\]

\[
\beta = B(\omega).
\]

Let the centerline or path of a wire be a continuous curve of finite length which begins and ends in a bubble. A path can be composed of a number of (partially) superimposed curves, each with its own width and layer associated with it. See Figure 1a. One can easily recognize that all the structures present in a layout may be represented in terms of the described primitives \([6]\). Moreover, note that \(|B|\) grows linearly with the number of atomic blocks and pins.

The design rules of a given technology can be seen as minimum spacing constraints between the perimeters of bubbles and paths. Alternatively, after proper scaling of the design rules, one can consider bubbles as points, and paths as curves of zero-thickness. For simplicity we have adopted this convention. Let us define rough routing as the specification of a continuous finite-length curve for each wire. Moreover, let topological routing be an equivalence class of rough routings connecting its pins. Two rough routings of the wire are equivalent when one can be obtained from the other by continuous deformation with no violations of any of the scaled design rules.

Define the edge on a given layer to be the line segment joining the centers of two bubbles on the layer. A wire and an edge are said to intersect topologically when the wire intersects the edge in every rough routing of a given topological routing. Let us partition any given layer into simply connected regions which contain no bubbles and whose boundaries are finite sets of edges. Let us call such regions simple regions. Figure 1b shows an example of such space partitioning along with two rough routes, \(A\) and \(B\), in the same equivalent class. Let us define \(E_\ell\) as the set of all simple regions in layer \(\ell\). Let us define a planar subset \(E \subset E_\ell\) as a set in which any pairs of distinct edges \(e_i, e_j\) either do not intersect each other or they intersect at only one of the vertices.

A triangulation \(T_\ell\) for layer \(\ell\) is a subset of \(E_\ell\) that satisfies the following conditions

1. \(\{e_i, e_j\} \in T_\ell, \forall i \neq j\) do not intersect or have one common vertex.
2. \(\forall P \subset E_\ell\), which forms a convex polygon, let \(P \supset P\) be the set of all edges whose vertices are contained in the polygon. Let \(\overline{P}\) be the set of all the vertices of \(P\). Then, the addition of any edge \(e_k \notin \overline{P}\) connecting two vertices in \(\overline{P}\) will violate 1.

Triangulation \(T_\ell\) is essentially a maximal planar set of edges when it has a convex boundary or convex hull. For more details on the properties of triangulations, the reader is referred to \([7]\).

Let us assume now that an arbitrary triangulation \(T_\ell\) is in place for each layer and let \(B_\ell\) be the set of all bubbles associated with \(T_\ell\). For convenience, although not needed, let us set four bubbles at the extremities of the union of all the layers, so as to encompass every layer. Moreover, assume that every rough routing has a source and a sink, i.e. two bubbles representing the beginning and the end of the route. Rough routings can now be represented as a sequence of edges which are crossed by traveling from the source to the sink. Consider rough route \(A\) in Figure 1b. Its representation according to the above convention is: \(\{6, 48, 47, 47, 47, 47, 14, 13, 3\}\), where bubbles 6 and 3 are respectively the source and sink, while \(V_1V_2 = \overline{V_1V_3}\) is the edge connecting vertices \(V_1\) and \(V_2\). All the bubbles are named sequentially \(a\) priori through an arbitrary scheme. Correct bubble identification is always guaranteed provided that such scheme is consistently used.

This representation captures the topology of a rough route in a non-unique fashion. Hence, in order to obtain a unique sequence for a given topological routing, one has to convert the sequence onto a canonical form. This is done simply removing adjacent identical edges, which form so-called loops. For example, the canonical form for rough route \(A\) is: \(\{6, 46, 47, 14, 3\}\), which is a unique sequence for the topological routing associated with both rough routes \(A\) and \(B\). The unique canonical form of an arbitrary topological routing \(\tau\) is called topological signature \(\sigma_\tau\). Call \(N_\tau\) the length of \(\sigma_\tau\). Loop removal is more complex when a large number of rough wires exists across a given space, the process in this case must be recursively performed.

The triangulation of any given design is not unique. In fact some but not all of the bubbles may be implemented in every layer, and every layer may or may not have different triangulation schemes. Consider now the layout implementation of Figure 2a. For simplicity but without loss of generality, assume only one layer and rectangular blocks. Let us assign a bubble to every pin and every Steiner point. A possible triangulation is shown in Figure 2b. In this case, the topological signature of \(A\) according to the usual convention is: \(\{1, \overline{23}, 4, 4, 5, 4, 67, 8\}\). A similar signature can be derived from multi-layer networks.

4 Watermark Synthesis

This section presents algorithm \(a(\sigma, k)\) for the generation of watermarks from the set of all topological signatures derived from the design as described in Section 3. There are several mechanisms which can be used, or any combination of them, for algorithm \(a\). All these mechanisms are aimed at selectively eliminating parts of the topological signatures according to a scheme controlled by \(k\).

Before proceeding with the description of the techniques, let us define \(\Sigma\) as the alphabet which includes all possible nodes and edges \(\overline{V_1V_3}\).
Then, the resulting signature is: \( \sigma_\tau = [0, 25, 54, 4]^T \). Then, the resulting signature is:

\[
\sigma_\tau = [0, 25, 54, 4]^T .
\]

Now that we have computed the topological signatures for all the existing topological routings, we need to combine them to obtain the final watermark. Given \( N \) topological signatures \( \sigma_{\tau, \tau} = 1, \ldots, N \), watermark \( w \); in form of a vector, results from the combination of such signatures as following

\[
w = [k_0, \sigma_1, k_0, \sigma_2, \ldots, k_N, \sigma_N] ,
\]

where \( k_\ell, \ell = 1, \ldots, N \) is again an arbitrary many-to-many mapping. A simple approach is to set \( k_\ell = I, \ell = 1, \ldots, N \), where \( I \) represents the identity matrix. Alternatively, \( k_\ell \) can be defined explicitly or implicitly as outlined above. The definition of \( k_\ell \) may be based on the criticality of net \( \eta \). For example, \( k_\ell = I \) when \( \eta \) is a critical net. Alternatively, assume that net \( \eta \) be constrained not to be accounted for, then \( k_\ell = 0 \).

As an illustration, consider nets \( \eta_1 \) and \( \eta_2 \), suppose \( \sigma_{\eta_1} = [9, 13, 7, 4, 54, 10]^T \) and \( \sigma_{\eta_2} = [67, 78, 12, 4]^T \). Then, assuming that \( k_0 = 0 \) and \( k_\ell = I \), the resulting watermark is:

\[
w = [67, 78, 12, 4]^T .
\]

Alternatively, consider the assignment \( k_0 = I \) and \( k_\ell = 0 \), resulting implicitly based on the following statement: consider all signatures \( \sigma_\ell \) whose size is a prime number. Then, the resulting watermark is:

\[
w = [9, 13, 7, 4, 54, 10]^T .
\]

## 5 Watermark Detection

A watermark should be detected in a design which has been deprived of any information other than geometries and connectivity. Hence, detection must be performed in two steps: topology extraction and signature identification.

Using standard slicing techniques [3], the layout is partitioned in rectilinear areas encompassing exactly one atomic block\(^1\). The complexity of this operation is \( O(\ell^2 \log \ell) \) where \( \ell^2 \) is the number of objects in the layout. Then, the contents of each slice are mapped in \( O(1) \) time onto the corresponding \( m \)-tuple of bubbles via Equation (4). All the bubbles in the layout are labeled and catalogued in order from left to right and from top to bottom. Then, using optimal algorithms, a triangulation is performed in \( O(\ell^2 \log \ell) \) time [7, p. 241]. Finally, the topological signature \( \sigma_\tau \) is extracted using a standard line segment intersection algorithm for the computation of the edges being cut by each topological routing. The complexity of this operation is again \( O(\ell^2 \log \ell) \) [7, p. 285].

In summary the complexity of the topology extraction method is \( O(n \log n) \), where \( n \) is the number of atomic blocks and pins present in the layout. As an illustration, consider the circuit in Figure 2a. After the scanning phase, one obtains a standard slicing structure, in this case a tripartite one. After the triangulation is performed, a structure similar to that of Figure 2b is obtained. Signature \( \sigma_\ell \) is then computed in a straightforward way.

In all watermark synthesis schemes, one wants to identify a given code with a certain level of confidence or conversely to compute the matching rate of the original signature with the extracted one, even when fragmentary. If no manipulation has occurred, using the original algorithm \( \alpha \) and key \( k \) on signature \( \sigma_\ell \), one can obtain the extracted watermark \( w_{\ell,k} \) which will be identical to the original watermark \( w \).

Let us now assume that the design has been tampered with. Since \( w \) was originally synthesized using algorithm \( \alpha \), only some sections of \( \sigma_\ell \) were used to generate \( w \); however these sections may be shifted and/or

\(^1\)Features not used in the watermarking process are ignored.
partially scrambled. To cope with this problem a technique known as genome search is used for identifying and counting the number of fragments in \( \sigma_x \) which also appear in \( w \). Let \( \mathcal{Y} \subseteq \Sigma^* \) be the set of all the signature fragments present in the original watermark \( w = \bigcup_{2 \leq i \leq \lambda} x_i \).

```plaintext
function genome_search(\( \sigma_x, \mathcal{Y} \))
foreach \((x \in \mathcal{Y})\) \{ y = best_match(\( \sigma_x, x, \text{length}(x) \)) overlap += overlap(x, y) / length(x) \[ \mathcal{Y} \]
\}
```

Function `best_match` selects subsequence of \( \sigma_x \) of length \( c \) which best matches \( b \). The matching criterium usually employed is the number of identical symbols. \( \text{length}(x) \) returns the length of \( x \), and \( \text{overlap}(x, y) \) computes the number of identical symbols in \( x \) and \( y \). This algorithm returns an estimate of the probability that the design contains in fact watermark \( w \).

Let us now focus on the measures defining the uniqueness and robustness of the proposed scheme. The set \( B \) of all the bubbles in the design and their location on the various layers is determined by the slicing algorithm and by mapping \( \mathcal{B} \). The topological signatures associated with any networks in the design are unique for a given triangulation. However, for each layer the number of possible triangulations grows factorially as \( |B_B| = 4! / 3! \). Next, choose layer \( \ell \) such that

\[
[B_B] = \frac{[B_B]}{\ell = 1, \ldots, L}
\]

Hence, by a conservative estimate, \( N_{\ell} \), the total number of possible triangulations over all layers, is \( N_{\ell} \geq (|B_B| - 1) / 3! \). Consider now all \( N_{\ell} \) topological routings in \( \mathcal{E} \). The routings consist of \( N_i \) \( i \)-terminal nets, \( i = 2, \ldots, N_{\ell_{\text{max}}} \). Note that all \( N_{\ell} \) topological routings can be represented in terms of \( N_t \) two-terminal sub-routings, with \( N_t = \sum_{i=2}^{N_{\ell_{\text{max}}}} N_i (i - 1) \). Hence, the number of possible topological signatures \( N_\sigma \) is given by

\[
N_\sigma \geq N_{\ell} \left( \frac{N_t}{2} \right) \text{ thus } P_u \leq \frac{1}{N_\sigma},
\]

where \( P_u \) is the probability of uniqueness. In case no tampering has occurred, then \( P_{\text{untampered}} \geq 0 \). As an illustration, consider a design with \( |B_B| = 20, N_{\ell_{\text{max}}} = 10, N_2 = 3, N_3 = 5, N_4 = 2 \). Hence, \( N_t = 19 \) and \( N_{\ell} \geq (20 - 1)! / 1! = 3.5 \times 10^{31} \), \( P_u \leq 2.9 \times 10^{-29} \).

In order to model tampering activities, let us consider the following occurrences: (1) routing modification, (2) atomic block modification, and (3) atomic block move and/or addition/deletion. (1) does not modify triangulation, however it may cause changes in the topological signature. There exists three types of possible effects on the signature: literal addition, deletion and swap. More than one literal may be involved in the change at any time, however, when this occurs, the change can be modeled in terms of a composition of simple literal modifications. (2) and (3) result in a change in the triangulation, thus potentially having an effect on one or more topological signatures. However, the effects can again be modeled in terms of simple literal operations.

Let \( P_{lt} \) be the probability that a literal change occurs in a topological signature. Moreover, let \( P_{lt} \) be the probability that a literal be included in the watermark. Then, the probability that a section of length \( t \) in watermark \( w \) mutates is

\[
P_t = \sum_{i=1}^{t} \left( \frac{|B_B|}{\ell} \right) [P_{lt}] \times [(1 - P_{lt})^t] \equiv 2 \times 10^{-9}.
\]

### Table 1: Benchmarks

| Circuit | Dev. | I/O | \(|T_x|\) | Segm. | CPU | \(P_u\) | \(P_m\) |
|---------|------|-----|----------|-------|-----|--------|--------|
| afa     | 86   | 15  | 534      | 101   | 0.35s| \(10^{-163}\) | \(1 \times 10^{-6}\) |
| cwords  | 26   | 16  | 200      | 25    | 0.1s | \(10^{-99}\) | \(4 \times 10^{-7}\) |
| create2 | 84   | 0   | 478      | 120   | 0.37s| \(10^{-129}\) | \(8 \times 10^{-7}\) |
| ccel6   | 21   | 12  | 134      | 23    | 0.07s| \(10^{-99}\) | \(3 \times 10^{-7}\) |
| ddx     | 374  | 0   | 2138     | 530   | 1.32s| \(10^{-900}\) | \(3 \times 10^{-6}\) |

### 6 Results

The techniques proposed in this paper were applied to a set of MCNC 86 benchmarks. For every circuit a topological signature was produced and then a watermark was created using the method outlined in Section 4. The experiments were performed on a 133MHz Pentium PC running under the Linux operating system.

We experimented with a wide range of circuits, from small library components to large circuits consisting of regular as well as non-regular structures. Table 1 lists all relevant experimental data for the circuits used, namely size, number of I/O pins, size of triangulation, and number of routing segments. The computed bounds on the probability of uniqueness \( P_u \) and on the miss rate \( P_m \) are also reported. For the computation of the miss rate it was assumed that \( P_t = P_{lt} = 10^{-1} \), while for the computation of \( P_{lt} \) the use of an arbitrary triangulation scheme was assumed. The results show the suitability of the method to the purpose of identifying to original designer in a case of patent or copyrights infringement.

### 7 Conclusions

A scheme has been proposed for Intellectual Property protection via watermarking. Watermarking is a technique which allows to stamp a given design with a unique structural code. The code can be reliably detected for the purpose of claiming ownership of designs illegally appropriated and/or forged. The method’s robustness and reliability have been shown through several industrial examples.

### References