IP watermarking using incremental technology mapping at logic synthesis level

Cui, Aijiao; Chang, Chip Hong; Tahar, Sofiène

2008

Cui, A., Chang, C. H., & Tahar, S. (2008). IP Watermarking Using Incremental Technology Mapping at Logic Synthesis Level. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 27(9), 1565-1570.

https://hdl.handle.net/10356/93106

https://doi.org/10.1109/TCAD.2008.927732

IP Watermarking Using Incremental Technology Mapping at Logic Synthesis Level

Aijiao Cui, *Student Member, IEEE*, Chip Hong Chang, *Senior Member, IEEE*, and Sofiène Tahar, *Senior Member, IEEE*

***Abstract*—This paper proposes an adaptive watermarking technique by modulating some closed cones in an originally optimized logic network (master design) for technology mapping. The headroom of each disjoint closed cone is evaluated based on its slack and slack sustainability. The notion of slack sustainability in conjunction with an embedding threshold enables closed cones in the critical path to be qualified as watermark hosts if their slacks can be better preserved upon remapping. The watermark is embedded by remapping only qualified disjoint closed cones randomly selected and templates constrained by the signature. This parametric formulation provides a means to capitalize on the headroom of a design to increase the signature length or strengthen the watermark resilience. With the master design, the watermarked design can be authenticated as in nonoblivious media watermarking. Experimental results show that the design can be efficiently marked by our method with low overhead.**

***Index Terms*—Digital watermarking, incremental technology mapping, intellectual property (IP) protection (IPP), logic synthesis.**

I. Introduction

THE INCREASE in integrated circuit (IC) complexity and the shorter design turnaround time have boosted reuse-based design methodology to speed up new product development in the system-on-chip era. Although significant time and effort have been invested in the creation of reusable intellectual property (IP) cores, piracy extorts sizable revenue from IP producers and poses a severe threat to reusable design. Digital watermarking has evolved as a mature technology to protect the copyright of multimedia content [1]. When this technique is applied to VLSI design [2]–[15], it augments the IP owner's opportunity to reclaim his loss of revenue. The imperceptibility of multimedia watermark stems from the imperfection of the human sensory system. In contrast, watermarking for IP protection (IPP) imposes that the watermarked IP must also remain functionally correct. The transparency of IP watermarking is achieved by making the changes induced by the watermark "invisible" to an experienced designer so that they could not be easily detected. At the same time, the cost and performance (quality) of the watermarked IP shall not be unduly compromised. The watermarked IP should also be resilient to malicious attacks without deteriorating the design functionality and performance to an extent that renders the design unusable.

Modern VLSI design flow involves many optimization procedures that require solving various NP-hard constraint satisfaction problems (CSPs). The solution space of these problems is normally enormous. Exploitation of the excess solution space of CSP has led to the first work in *constraint-based watermarking* for IPP by Lach *et al.* [3], [4] and Hong and Potkonjak [5], which was later formally articulated in [6] and [7]. In this approach, the ownership credibility is determined by the probability of coincidence P_c [8]. The proof requires revoking the CSP instance to demonstrate that the additional constraints are satisfied by the watermarked design. This process tends to expose the signature and secrecy of the well-formed grammar used to generate the constraints, making other similarly marked designs vulnerable to attacks. Except for some local watermarking schemes [9], [10], [13], [14], all watermark bits in global watermarking are closely coupled into a set of design constraints, which cannot be independently extracted to detect the locality of design changes due to partial obliteration.

This paper presents a new constraint-based watermarking technique at logic synthesis level [14]. Instead of passively accepting the success or failure until the watermarking is completed, we use an originally optimized design to extract the excess bandwidth relative to the original timing specifications. Unlike other logic synthesis watermarking [9], [10], where the watermarked solution quality is limited by the technology mapping of overlapping maximum cones, we introduce independent disjoint closed cones for incremental technology mapping to maximally exploit the excess bandwidth. As timing criticality is obscure in a small disjoint closed cone, a slack sustainability is formally defined to determine its headroom for remapping. Using both slack and slack sustainability to qualify cones for remapping prevents the watermark bits from being conspicuously hosted in only the noncritical paths. This incremental mapping preserves as much optimality of the master design and can provide a well-defined control mechanism to trade the embedded capacity for a bounded overhead. To avoid the exposure of the grammar for constraint generation, an alternative verification analogous to the watermark retrieval of nonoblivious image watermarking is made possible by our embedding method. This watermark retrieval method also possesses some degree of fragility that enables the detection of maliciously corrupted watermark bits.

Manuscript received June 1, 2007; revised September 30, 2007 and February 25, 2008. Published August 20, 2008 (projected). This paper was recommended by Associate Editor S. Nowick.

A. Cui and C. H. Chang are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: cuiaijiao@pmail.ntu.edu.sg; echchang@ntu.edu.sg).

S. Tahar is with the Department of Electrical and Computer Engineering, Concordia University, Montreal, QC H3G 1M8, Canada (e-mail: tahar@ece.concordia.ca).

Digital Object Identifier 10.1109/TCAD.2008.927732

0278-0070/$25.00 © 2008 IEEE

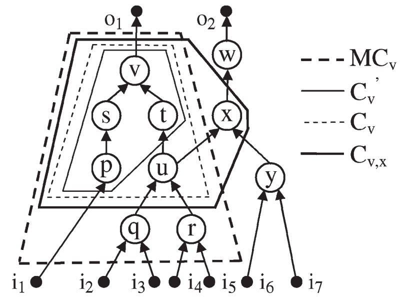

Fig. 1. Examples of a DAG network, a max cone (MC_v), a closed cone (C'_v), an open cone (C_v), and a complex cone ($C_{v,x}$).

II. Preliminaries

A bound logic network can be represented as a directed acyclic graph (DAG) $G(V, E)$, where each node $v \in V$ represents an instance of a cell library. A directed edge $e_{i,j} \in E$ exists if $\exists v_i, v_j \in V$ such that the output of cell v_i is an input of cell v_j. The sets of fan-in and fan-out nodes of v_j are denoted by $FI(v_j)$ and $FO(v_i)$, respectively. A *primary input/output* (PI/PO) has no fan-in/fan-out node. A *cone* at node v, denoted as C_v, is defined as a subgraph consisting of v and at least one of its predecessors. If all the predecessors of v are contained in the cone, C_v is called the *max cone* at v [16].

The set of nodes outside C_v that drive the nodes (or are driven by the nodes) in C_v will be denoted by $FI(C_v)$ [or $FO(C_v)$]. Any node in C_v that drives (is driven by) at least a node outside C_v or a PO (PI) is called the output (input) node of C_v. C_v is a *simple* cone if it has only one output node, i.e., v. Any output node of C_v that drives at least one internal node in C_v, if it exists, is known as an implicit output node. A cone is *closed* if it has no implicit output nodes. If a cone contains any implicit output node, it is said to be *open*. A *complex* cone $C_{v1,v2,\ldots,vn} = C_{v1} \cup C_{v2} \cup, \ldots, C_{vn}$ is a conjunction of two or more connected cones with at least one common transitive fan-in node between any two cones. A simple open cone can be made closed by either excluding all its implicit output nodes or including the transitive fan-in and fan-out nodes of the implicit output nodes until the complex cone so formed contains no implicit output node. The distance $d(v, u)$ is the minimum number of transitive edges connecting nodes u and v. The distance $d(C)$ is defined as the maximum distance among all pairs of input and output nodes of cone C.

Fig. 1 shows the DAG of a technology-mapped network. Node u is an implicit output node of C_v as it drives an internal node t. By definition, C_v is a distance-2 open simple cone due to $d(p, v) = 2$ and node u. C'_v is a closed cone obtained by eliminating u from C_v. $C_{v,x}$ is a closed complex cone obtained by expanding C_v to cover x, which is the fan-out node of u.

The following definitions are adopted from [17]. The *arrival time* of the signal at node v, denoted by $t_a(v)$, is the latest time when the signal at node v reaches its final stable state:

$$t_a(v) = \max_{u \in FI(v)} \left[t_a(u) + t_d(u, v) \right]. \tag{1}$$

The *required time* of the signal at node u, denoted by $t_r(u)$, is the earliest time when the signal at node v is required to be stable. $t_r(u)$ can also be calculated recursively by

$$t_r(u) = \min_{v \in FO(u)} \left[t_r(v) - t_d(u, v) \right]. \tag{2}$$

```
disjoint_closed_cone($v$, $\tau$) {
  $C_v = \{v\}$;
  for each ($u \in FI(C_v)$ and $d(u, v) \le \tau$ and $u$ is unmasked) {
    if ($u \ne PI$) {
      mask($u$);
      $C_v = C_v \cup \{u\}$; }
    flag = 1;
    for each ($x \in FO(u)$ and $x \notin C_v$) {
      if ($t_s(x) \le t_s(v)$ or $x$ is masked) {
        unmask($u$);
        $C_v = C_v - \{u\}$;
        flag = 0;
        break; }
    }
    if (flag =1) {
      mask($FO(u)$);
      $C_v = C_v \cup FO(u)$; }
  }
  if ($C_v$ is trivial) return $\varnothing$;
  else return $C_v$;
}
```

Fig. 2. Generation of distance-τ disjoint closed cone.

The *slack* of a node v, denoted by $t_s(v)$, is given by

$$t_s(v) = t_r(v) - t_a(v). \tag{3}$$

III. Proposed IP Watermarking Scheme

A. Watermark Insertion

An arbitrary length binary signature W is encrypted using a public-key cryptosystem and then reduced by a message digest to a constant length sequence W_e. To maximally utilize the embedding capacity of a design in incremental technology mapping, W_e is embedded into an originally optimized circuit generated by a logic synthesis tool with a cell library. This cover design is called the master design G.

To minimize and isolate the topology perturbations to small regions of selected cells in technology mapping, disjoint closed cones instead of max cones are used as watermark hosts. There must be at least one substitutable template for any cell in a selected cone to enable remapping. There must also be enough spatially uncorrelated closed cones to reduce the odds that a randomly selected cone coincides with the watermark hosts. To provide a pool of fair candidates, each cell in the master design is first expanded to a disjoint closed cone of distance τ, where τ is empirically determined based on the circuit complexity, the watermark length, and the cell library resources.

Fig. 2 shows the procedure to expand a cell v into a simple or complex closed cone of distance τ. It can be implemented by a depth-first traversal from v to include all its predecessors into C_v until the distance from the predecessors to v is τ or any PI or implicit output node is met. When an implicit output node is encountered, either a simple or complex closed cone is to be generated. As multivertex mapping requires more effort to preserve the timing constraints, to justify for a complex cone, all implicit output nodes of the closed cone must have greater slacks than node v. Therefore, a complex cone $C_{v1,v2,\ldots,vn}$ of distance τ is generated only if the following criterion is met:

$$t_s(v_i) > t_s(v_1) \qquad \forall v_i \in FO(C_{v1,v2,\ldots,vn}), \quad v_i \neq v_1 \tag{4}$$

where v_1 is the cell to be expanded into a distance-τ closed cone.

As an example, consider node v in Fig. 1. For $\tau = 2$, the complex closed cone $C_{v,x}$ is generated if $t_s(x) > t_s(v)$. Otherwise, node u is excluded, and a simple closed cone C'_v is generated.

As the embedding capacity is limited by the ability of a design to preserve its timing constraints in remapping, to maximally utilize the available embedding capacity for robust watermarking, a metric is developed to formally qualify the disjoint closed cones generated for watermark embedding. The slack of a cone $C_v = t_s(v)$ provides a global view of the margin available for remapping the cells along the critical path of the max cone rooted at v. However, a distance-τ closed cone C_v can preclude cells in the critical path of the max cone of v that are further than distance τ from v. To determine how well a closed cone C_v preserves its output timing, we can remap only C_v of G to obtain a remapped solution G'. The *sustainability* of a closed cone C_v is defined as

$$S(C_v) = 1 + \frac{\Delta t_a(v)}{t_a(v)} = 1 + \frac{t_a(v) - t'_a(v)}{t_a(v)} \tag{5}$$

where $t_a(v)$ and $t'_a(v)$ are the arrival times of the same node v in G and G', respectively. We add a bias of one to the fractional delay so that when $S(C_v) = 1$, the original slack is retained, hence the term slack sustainability. Timing is improved by remapping if $S(C_v) > 1$ and aggravated if $S(C_v) < 1$.

A resynthesis is needed for each closed cone to evaluate its sustainability. To increase the efficiency, a correlated metric to the sustainability is statically derived from the master design without the need for the physical resynthesis process

$$\hat{S}(C_v) = \hat{S}(v \in C_v) = \frac{1}{|FI(v)|} \sum_{u \in FI(v)} \left(\hat{S}(u \in C_v) + \delta(v,u) \right) \tag{6}$$

where $|FI(v)|$ is the number of fan-in nodes of v, and

$$\delta(v,u) = \frac{t_d(u,v)}{t_a(u) + t_d(u,v)}. \tag{7}$$

$\delta(v,u)$ accounts for the fractional delay a cell v contributes to its critical path through its fan-in node u. $0 < \hat{S}(C_v) \leq 1$. The recursion of $\hat{S}(u)$ terminates at $u = FI(C_v)$, and $\hat{S}(u) = 0 \; \forall u \in FI(C_v)$. If C_v is a max cone, $\hat{S}(C_v) = 1$. The higher the $\hat{S}(C_v)$, the better the closed cone preserves its timing slack upon remapping. The greater the slack, the easier the delay constraint is met by remapping. The headroom of a closed cone is thus defined as

$$w(C_v) = t_s(v) \times \hat{S}(C_v). \tag{8}$$

```
watermark_insert($G$, $Lib$, $W$, $K$, $\tau$, $\alpha$) {
    Compute and store timing information of $t_s$, $t_a$ in $G$.
    $i = 1$; Initialize disjoint cone array, $Q$;
    for each ($v \in G$ ) {
        $Q[i]$ = disjoint_closed_cone($v$, $\tau$);
        if ($Q[i] \neq \emptyset$) {
            $cone\_id[i]$ = identifier of $v$;
            $i = i + 1$; }
    }
    $w_T$ = threshold($\alpha$, $G$);
    $Q$ = qualify_closed_cone($Q$, $w_T$);
    $W_e = MD(PKC(W, K))$;
    $m = |W_e|$; $q = |Q|$;
    Initialize array $TP_1$, $TP_0$, $Q_1$ and $Q_0$;
    $index[1..m]$ = pseudo_random_generator($W_e$, $q$);
    for ($i$ = 1 to $m$) {
      if ($W_e[i] = 1$) {
            $TP_1 = TP_1 \cup$ any template in $Q[index[i]]$;
            $Q_1 = Q_1 \cup Q[index[i]]$;
            mark the selected $Q[index[i]]$ and its forbidden template in $G$;
      } else {
            $TP_0 = TP_0 \cup$ any template in $Q[index[i]]$;
            $Q_0 = Q_0 \cup Q[index[i]]$;
            mark the selected $Q[index[i]]$ and its sentinel template in $G$; }
    }
    $G'$ = remap($G$, $Q_1$, $Q_0$, $TP_1$, $TP_0$, $Lib$);
    return $G'$;
}
```

Fig. 3. Watermark insertion by incremental technology mapping.

To qualify closed cones for watermark embedding, an empirical threshold w_T is defined such that all cones with headroom less than w_T are pruned. To reduce the probability of qualifying closed cones that could adversely affect the performance of the watermarked design, w_T is judiciously set to α times the minimum slack of all path groups of G

$$w_T = \alpha \times \min_i \left\{ t_s\left(PO_i(G)\right) \right\}. \tag{9}$$

As the excess slacks on less critical path groups are expected to be traded for the diminished slacks on more critical path groups, α provides the tradeoff control of the embedding capacity of a design and its watermarked solution's timing specification.

Each qualified closed cone with $w(C_v) \geq w_T$ is uniquely identified. To obfuscate the locality of the watermark, m qualified closed cones are randomly selected by a cryptographically strong pseudorandom number generator seeded with the m-bit encrypted watermark W_e. This keyed one-way function ensures that with high probability, a different set of closed cones will be selected when the bitstream in W_e is permuted. To embed a logic "1", a selected closed cone is coerced to remap by prohibiting the use of one randomly selected template inside the cone. To embed a logic "0", one template of the selected cone is preserved. The selected 1- and 0-watermarked cones are stored in Q_1 and Q_0, and their designated forbidden and preserved templates are stored in TP_1 and TP_0, respectively, for watermark recovery. The watermarked solution, G' is generated by remapping only Q_1 and Q_0 of G according to their template constraints, TP_1 and TP_0, respectively. The algorithm for watermark insertion is shown in Fig. 3.

B. Watermark Detection

To prove the presence of watermark in an IP, the original CSP instance is recalled to verify that the additional constraints imposed are satisfied by the watermarked design [8]. This process exposes the grammar used to generate the constraints.

Here, we present another watermark detection method, which possesses some features of fragile watermarking. Other designs similarly marked with the same signature will not become more vulnerable by the information divulged in this process.

To facilitate the retrieval of the watermarked closed cones, correspondences between internal nets of the watermarked design and designated nets of the master design are identified by functional equivalence. The logic cones $C' \in G'$ and $C \in G$ are said to be equivalent iff there exists a bijection between $FI(C')$ and $FI(C)$ and between $FO(C')$ and $FO(C)$. Let $f(v)$ be the logical function of cell v; then, $C' \equiv C$ iff $|FI(C')| = |FI(C)|$ and $|FO(C')| = |FO(C)|$, and $\exists u \in FI(C')$ such that $f(u) \equiv f(v)\ \forall v \in FI(C)$, and $\exists u \in FO(C')$ such that $f(u) \equiv f(v)\ \forall v \in FO(C)$, where "$\equiv$" denotes equivalence of logical functions.

Since incremental technology mapping preserves the functionality of the interface ports of remapped cones, the logic functions of these nets can be retrieved from the fan-in and fan-out nodes of closed cones saved in the master design G. To recover the watermarked cone C' from a marked design G', nodes with the same logic functions are extracted. Let $C' \equiv C$ be a cone extracted from the watermarked design G' with n fan-ins and k fan-outs. $f(WP_i) \equiv f(P_j)$ for $i, j = 1, 2, \ldots, n+k$, where WP_i and P_j are the interface ports of C' and C, respectively. If C is used to embed a logic "0" and the designated template is found in C', a valid logic "0" is retrieved. If C is used to embed a logic "1" and the designated template is not found in C', a valid logic "1" is recovered. Otherwise, the embedded bit has been corrupted. When equivalent fan-in and fan-out nodes of a watermarked cone cannot be found, it implies that either the cells within the cone or its neighboring cells have been modified. The authorship is proved by a perfect or high match between the recovered bit stream and the embedded watermark.

IV. Experimental Results

The strength and robustness of an IP watermarking scheme are generally evaluated by the probability of coincidence P_c and the probability of removal P_r. As it is difficult to unambiguously or unconditionally measure the credibility of IC IPPs in practice, we conservatively consider the potential high-quality solutions that are less likely to violate the design constraints. The closed cones that span this subspace have been qualified by the threshold w_T in our watermarking flow. Thus, the probability of m uniquely labeled closed cones being selected from q closed cones that satisfy the headroom constraints is $1/C_m^q$. Let p_0 and p_1 be the probabilities that a template in the selected closed cone is preserved or changed in the remapping process, respectively. Assume that W_e has an equal number of "1" and "0" bits, the probability that a solution carries the watermark by coincidence is given by

$$P_c = \frac{1}{C_m^q}(p_0)^{\frac{m}{2}}(p_1)^{\frac{m}{2}} \approx \frac{1}{C_m^q}\left(\frac{1}{2}\right)^m. \qquad (10)$$

Assume that it is equally probable that a designated template is extricated or preserved when a closed cone is remapped, i.e., $p_0 = p_1 \approx 0.5$. From (10), it is obvious that increasing the signature length increases the watermark strength only if the design itself offers sufficient redundancy. Given a fixed signature length m, a high exploitable redundancy space q also increases the watermark strength, provided that α in (9) is judiciously selected to preserve the original timing constraints.

Since combinational circuit watermarking at logic synthesis level is not resilient against resynthesis attacks, the watermarked design is made available as a technology-specific firm or hard IP. HDL codes that can be directly exploited for resynthesis without substantial design and verification effort shall not be revealed. This makes node manipulation difficult for our method as it will affect the circuit timing even if the attacker can reverse engineer parts of the circuit netlist to recover some local functions with reasonable effort. To delete a large portion of the watermark bits and still preserve the solution quality, the attacker has to substantively perturb the watermarked netlist, which results in a task of effort comparable to complete circuit optimization. Assume that a brute-force attack is performed to alter γ cells at random followed by a timing analysis to ensure that the constraints are still satisfied with reasonable effort and without changing the circuit functionality. To simplify the analysis, we further presume that it needs only to alter any cell of a watermarked cone to successfully erase one watermark bit. The probability of erasing i bits of an m-bit watermark from a design with N disjoint cones is given by

$$P_r(n=i) = C_i^\gamma \prod_{j=0}^{i-1}\left(\frac{m-j}{N-j}\right) \cdot \prod_{j=0}^{\gamma-i-1}\left(\frac{N-m-j}{N-i-j}\right). \qquad (11)$$

The probability of removing more than $k (k \leq m)$ watermark bits is given by

$$P_r(n \geq k) = \sum_{i=k}^{m} P_r(n=i). \qquad (12)$$

In the following experiments, we used Synopsys Design Compiler and its standard cell library to synthesize the circuits from ISCAS85, ISCAS89, ISCAS99, and LGSynth93 benchmark suites. All experiments were run on a 750-MHz Sun UltraSPARC-III with 2 GB of memory running a Solaris operating system. The master design of each circuit was obtained by using the timing constraints derived by the synthesis tool from the initially unconstrained designs. A total of 100 different signatures of length m were used for the watermark embedding on each master design, and the average percentage increases in the area (ΔA), delay (ΔD), and power (ΔP) of the watermarked design over the master design were reported in Tables I and II. The columns "#cells", "*N*", "*q*", "P_c", and "P_r" are, respectively, the total number of combinational cells, the number of disjoint closed cones, the number of qualified closed cones with $\tau = 1$, the probability of coincidence, and the probability of successfully removing more than three quarters of the watermark bits by randomly altering m cells, i.e., $k = 3$ m/4, $\gamma = m$. For "ex1010" and "B22", $\tau = 2$ was also experimented with 64-bit and 128-bit signatures.

For a given master design, when more watermark bits were inserted, higher overheads were incurred. As the number of cells increases, the overheads diminish and become negligible. In Table II, it is also observed that the area and power overheads

TABLE I
WATERMARKING RESULTS ON ISCAS BENCHMARKS

Circuit	#cells	N	q	m	ΔA(%)	ΔD(%)	ΔP(%)	P_c	P_r
C2670	430	92	92	32	1.09	-2.11	0.97	4.10E-35	4.97E-09
C5315	721	125	125	32	3.19	-0.43	1.80	3.76E-40	1.78E-12
C7552	1069	243	243	64	1.79	-3.25	0.96	1.33E-79	3.30E-23
C6288	1979	466	145	64	1.29	1.07	1.83	4.95E-62	1.61E-37
S5378	734	140	84	32	1.02	0.74	0.56	1.49E-33	9.83E-14
S9234	896	191	187	64	1.86	5.53	0.30	5.83E-71	8.65E-18
S38417	4927	1223	1127	64	0.30	-1.53	0.32	2.02E-125	3.87E-58
				128	0.65	0.00	0.44	4.63E-211	1.60E-85
S38584	6520	1204	1141	64	0.47	-1.81	-0.27	8.98E-126	8.29E-58
				128	0.73	-2.72	-0.12	8.66E-212	7.53E-85
B22	5499	1313	1290	64	-0.10	-0.32	0.11	2.82E-129	1.22E-59
				128	0.07	1.87	0.11	5.37E-219	1.44E-88
B22 ($\tau = 2$)	5499	1172	1150	64	-0.11	-0.65	0.11	5.35E-126	3.08E-57
				128	0.01	-0.65	0.11	2.98E-212	1.08E-83
B21	3605	886	875	64	0.40	-0.65	1.63	3.76E-118	2.66E-51
				128	1.01	-0.65	2.04	5.29E-196	1.32E-71

TABLE II
WATERMARKING RESULTS ON LGSYNTH93 BENCHMARKS

Circuit	#cells	N	q	m	ΔA(%)	ΔD(%)	ΔP(%)	P_c	P_r
i7	269	61	61	16	10.85	-8.70	10.17	7.52E-20	1.38E-06
i2	190	57	57	16	3.68	28.30	-2.11	2.64E-19	3.29E-06
i9	239	30	30	16	6.34	5.70	12.73	1.05E-13	1.40E-02
frg2	389	58	58	16	1.34	2.34	1.09	1.91E-19	2.63E-06
rot	374	76	76	16	1.13	-1.64	1.98	1.41E-21	8.37E-08
apex5	420	78	78	16	3.49	-0.80	5.72	8.87E-22	6.03E-08
alu4	748	122	120	32	3.68	-1.48	4.35	1.70E-39	3.33E-12
apex6	381	72	72	32	3.31	-0.80	0.47	8.16E-31	3.07E-06
x3	430	96	91	32	2.34	0.00	0.85	6.28E-35	1.64E-09
k2	542	100	76	32	1.74	0.47	0.64	8.64E-32	5.67E-10
i8	416	80	74	32	2.59	-2.06	1.33	2.60E-31	1.93E-07
dalu	356	93	93	32	3.06	4.74	0.67	2.69E-35	3.75E-09
des	1680	267	173	64	0.89	17.84	1.05	2.69E-68	2.62E-25
spla	2015	364	265	64	1.38	0.00	1.04	2.26E-82	3.85E-32
mm30a	844	215	215	64	0.96	20.83	0.61	1.28E-75	1.84E-20
pair	1087	314	314	64	2.69	5.59	3.70	1.08E-87	6.76E-29
i10	1431	335	335	64	1.32	-0.48	1.48	1.07E-89	2.54E-30
pdc	2288	452	403	64	1.46	-0.62	1.60	2.48E-95	7.39E-37
				128	2.57	-1.85	2.18	2.75E-147	6.50E-42
ex1010	5411	1316	1316	64	1.53	10.13	11.72	7.61E-130	1.09E-59
				128	2.05	10.97	16.27	3.66E-220	1.15E-88
ex1010 ($\tau = 2$)	5411	1082	1082	64	1.28	9.28	10.35	2.97E-124	1.52E-55
				128	1.90	10.97	15.70	1.18E-208	2.99E-80
elliptic	3626	793	682	64	1.53	-6.29	0.00	6.32E-111	6.14E-49
				128	1.82	11.98	0.22	7.28E-181	8.88E-67

TABLE III
COMPARISON OF LOGIC SYNTHESIS WATERMARKING METHODS

Circuit	M	Proposed		[10]		[11]
		ΔA (%)	P_c	ΔA (%)	P_c	ΔA (%)
i7	18	14.83	**2.91E-21**	**2.88**	4.22E-14	
	36	15.91	**1.65E-28**	**1.44**	6.98E-28	
i2	22	**6.62**	**6.83E-23**	10.74	1.87E-13	
	43	**8.26**	**1.48E-26**	26.45	1.00E-23	
i9	19	**6.74**	3.49E-14	8.57	**2.33E-14**	
alu4	25	**2.44**	**7.14E-34**	6.82	5.84E-11	
	49	**4.18**	**1.37E-49**	11.82	3.34E-20	
frg2	14	**0.24**	6.02E-18			2.04
	21	**0.52**	**1.43E-22**	2.98	3.01E-10	
	41	**4.93**	**2.30E-27**	8.61	4.15E-18	
rot	20	1.88	8.75E-25			**1.53**
	24	**2.27**	**1.58E-27**	4.53	2.73E-12	
	48	**3.89**	**7.13E-36**	8.36	1.04E-22	
apex6	20	**3.75**	3.06E-24			11.24
	26	**3.75**	**5.40E-28**	5.37	6.41E-16	
	51	**6.58**	**5.75E-34**	10.74	2.29E-29	
C2670	19	**2.08**	8.34E-26			2.55
	29	**2.49**	**2.63E-33**	7.27	9.51E-16	
	58	**4.75**	**1.94E-43**	12.42	7.22E-29	
x3	18	**1.61**	8.08E-25			5.92
	28	**1.68**	**1.67E-32**	7.89	5.93E-16	
	55	**2.03**	**9.70E-43**	14.66	5.12E-29	
k2	33	**2.02**	**3.24E-32**	2.91	1.84E-10	
	66	**2.83**	**1.42E-32**	6.05	3.00E-19	
i8	15	2.28	1.67E-20			**0.86**
	34	3.46	**4.24E-32**	**-4.45**	1.06E-10	
	67	6.77	**3.77E-30**	**-10.83**	4.53E-23	
dalu	43	4.40	**1.81E-40**	**4.19**	1.07E-19	
	86	**8.24**	**1.36E-36**	11.26	3.88E-36	
C5315	55	**4.53**	**2.24E-53**	6.30	2.92E-32	
	110	**7.57**	8.50E-53	14.57	**4.40E-59**	
pair	19	**-0.27**	1.46E-36			0.38
	58	**1.95**	**3.39E-82**	6.73	3.99E-31	
	115	**4.36**	**1.34E-123**	14.62	6.88E-57	
C6288	97	**1.93**	**9.36E-69**	10.72	7.01E-51	
C7552	98	**3.68**	**4.15E-100**	8.25	1.48E-52	
	196	**6.32**	**2.27E-110**	17.93	4.42E-96	
des	112	2.12	**5.23E-82**	**0.79**	3.21E-55	
i10	119	**2.24**	**7.27E-130**	4.26	5.98E-50	
	238	**3.88**	**1.34E-158**	8.67	4.48E-94	
apex5	4	1.48	4.38E-08			**0.38**
S9234	19	**0.38**	4.09E-32			2.61
S5378	20	**1.73**	8.88E-26			2.60
mm30a	20	**-0.04**	1.30E-34			1.36

of the 128-bit watermarked "pdc" were both less than 2.6%, and the timing was actually improved (negative percentage difference). For an even larger circuit "B22", the area and power overheads reduced to only 0.07% and 0.11%, respectively, while the delay increased slightly by 1.87% for $\tau = 1$. For $\tau = 2$, the watermark introduced only a negligible 0.01% area overhead, and the timing was improved. For most designs, as more watermark bits were inserted, a lower P_c was obtained. If $q > 1000$, $P_c < 10^{-200}$ for $m = 128$. When a design possesses more than 600 disjoint cones, $P_r < 10^{-50}$ for $m = 128$. These results show that it is more difficult to successfully remove a sufficient number of watermark bits from a larger design. We have also experimented with $k = m/2$. It is found that the attacker can easily remove at least half of the watermark bits for a small number of watermarked circuits with $N/m \leq 2$, and the P_r values remain very low when $N/m > 3$. When $N > 700$ and $m = 128$, the probability of successfully removing more than 64 watermark bits is less than 10^{-19}. In general, τ can be increased to lower the overhead of a watermarked solution. However, as τ increases, the number of qualified disjoint closed cones is reduced, and P_c and P_r become higher.

In Table III, we compare the overheads of our watermarking method with two other methods in [10] and [11]. The comparison was performed by excerpting the results in [10] and [11] for the same benchmark circuits. In [10], irregular signature lengths were used. Two different signature lengths based on the cases of 4% and 8% of gates being constrained as pseudoprimary outputs [10] were used to compare the embedding overheads and P_c values with [10]. In [11], P_c was not evaluated. Thus, only the area overhead was compared using the same watermark length equivalent to the number of additional constraints in [11]. An empty entry indicates an unavailable result. It is found that for most designs, our method provides the strongest proof of authorship with a lower overhead.

V. Conclusion

We have presented a constraint-based watermarking method at the logic synthesis level for IPP. In our approach, good localities for watermark insertion are identified based on the notion of slack sustainability. The formal qualification of hosting cones, in conjunction with an incremental technology mapping, has several distinct advantages. First, the original design constraints can be satisfied with a very low embedding overhead. Second, the signature length can be adaptively tailored to the embedding capacity of a design to optimize the ownership proof, watermark obscurity, and resilience with low impact on circuit quality. Third, the existence of the watermark can be explicitly detected by a direct extraction method by comparing the master design and the watermarked design. The correlation check also provides the cue on an attacker's attempt of obliteration.

The experimental results show that our method generally performs better than other watermarking methods at the logic synthesis level in terms of the overhead and watermark strength under the same signature length. The results also indicate that the probability of successful random obliteration descends rapidly with increased disjoint-cones-to-signature-length ratio. There is potential to integrate our method with dynamic watermarking at a higher design level [15] to offer greater flexibility to reuse the watermarked IP cores with low risk of successful attacks.

References

[1] C. H. Chang, Z. Ye, and M. Zhang, "Fuzzy-ART based adaptive digital watermarking scheme," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 1, pp. 65–81, Jan. 2005.

[2] A. T. Abdel-Hamid, S. Tahar, and E. M. Aboulhamid, "A survey on IP watermarking techniques," in *Design Automation for Embedded Systems*, vol. 10. London, U.K.: Springer-Verlag, 2005, pp. 1–17.

[3] J. Lach, W. H. Mangione-Smith, and M. Potkonjak, "Fingerprinting digital circuits on programmable hardware," in *Information Hiding*, vol. 1525. London, U.K.: Springer-Verlag, 1998, pp. 16–31.

[4] J. Lach, W. H. Mangione-Smith, and M. Potkonjak, "FPGA fingerprinting techniques for protecting intellectual property," in *Proc. IEEE Custom Integr. Circuits Conf.*, Santa Clara, CA, May 1998, pp. 299–302.

[5] I. Hong and M. Potkonjak, "Techniques for intellectual property protection of DSP designs," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Seattle, WA, May 1998, vol. 5, pp. 3133–3136.

[6] G. Qu and M. Potkonjak, "Hiding signatures in graph coloring solutions," in *Proc. 3rd Int. Workshop Inf. Hiding*, Dresden, Germany, Sep. 1999, pp. 348–367.

[7] G. Qu and M. Potkonjak, *Intellectual Property Protection in VLSI Design: Theory and Practice*. Boston, MA: Kluwer, 2003.

[8] 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 Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 20, no. 10, pp. 1236–1252, Oct. 2001.

[9] D. Kirovski, Y. Y. Hwang, M. Potkonjak, and J. Cong, "Intellectual property protection by watermarking combinational logic synthesis solutions," in *Proc. IEEE/ACM Int. Conf. CAD*, San Jose, CA, Nov. 1998, pp. 194–198.

[10] D. Kirovski, Y. Y. Hwang, M. Potkonjak, and J. Cong, "Protecting combinational logic synthesis solutions," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 12, pp. 2687–2696, Dec. 2006.

[11] S. Meguerdichian and M. Potkonjak, "Watermarking while preserving the critical path," in *Proc. ACM/IEEE Des. Autom. Conf.*, Los Angeles, CA, Jun. 2000, pp. 108–111.

[12] M. Moiz Khan and S. Tragoudas, "Rewiring for watermarking digital circuit netlists," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 24, no. 7, pp. 1132–1137, Jul. 2005.

[13] D. Kirovski and M. Potkonjak, "Local watermarks: Methodology and application on behavioral synthesis," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 22, no. 9, pp. 1277–1283, Sep. 2003.

[14] A. Cui and C. H. Chang, "Stego-signature at logic synthesis level for digital design IP protection," in *Proc. IEEE Int. Symp. Circuits Syst.*, Kos, Greece, May 2006, pp. 4611–4614.

[15] A. T. Abdel-Hamid, S. Tahar, and E. M. Aboulhamid, "A public-key watermarking technique for IP designs," in *Proc. Des. Autom. Test Eur.*, Munich, Germany, Mar. 2005, vol. 1, pp. 330–335.

[16] J. Cong and H. Huang, "Depth optimal incremental mapping for field programmable gate arrays," in *Proc. ACM/IEEE Des. Autom. Conf.*, Los Angeles, CA, Jun. 2000, pp. 290–293.

[17] J. Y. Jou and D. S. Chou, "Sensitisable-path-oriented clustered voltage scaling technique for low power," *Proc. Inst. Electr. Eng.—Computers Digital Techniques*, vol. 145, no. 4, pp. 301–307, Jul. 1998.

Aijiao Cui (S'06) received the B.Eng. degree and the M. Eng. degree in electronics from Beijing Normal University, Beijing, China, in 2000 and 2003, respectively. She is currently working toward the Ph.D. degree in electrical and electronic engineering in the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore.

From July 2003 to December 2004, she was a Teaching Assistant with Beijing Jiaotong University, Beijing, China. Her main research interest is digital watermarking for IP protection.

Chip Hong Chang (S'92–M'98–SM'03) received the B.Eng. (Hons.) degree from the National University of Singapore, Singapore, in 1989, and the M.Eng. and Ph.D. degrees from Nanyang Technological University (NTU), Singapore, in 1993 and 1998, respectively.

He served as a Technical Consultant in the industry prior to joining the School of Electrical and Electronic Engineering, NTU, in 1999, where he is currently an Associate Professor. He has held joint appointments at the university as the Deputy Director of the Centre for High Performance Embedded Systems (CHiPES) since 2000 and the Program Director of the Centre for Integrated Circuits and Systems (CICS) since 2003. His current research interests include low-power arithmetic circuits, digital filter design, application-specific digital signal processing, and digital watermarking for IP protection. He is the author of three book chapters and more than 130 research papers in international refereed journals and conference proceedings.

Dr. Chang is a Fellow of the IET, U.K. He has been involved in organizing and program committees of various international conferences. He is listed in the 2008 and 2009 *Marquis Who's Who in the World*. He serves as an Editorial Advisory Board Member of *The Open Electrical and Electronic Engineering Journal*.

Sofiène Tahar (M'96–SM'07) received the Diploma degree in computer engineering from the University of Darmstadt, Darmstadt, Germany, in 1990, and the Ph.D. degree with distinction in computer science from the University of Karlsruhe, Karlsruhe, Germany, in 1994.

From 1995 to 1996, he was a Postdoctoral Fellow with the Université de Montréal, Montréal, QC, Canada. Currently, he is a Professor with the Department of Electrical and Computer Engineering, Concordia University, Montreal, QC, Canada. He is the Founder and the Director of the Hardware Verification Group at Concordia University. From 2001 to 2006, he held a Junior Concordia University Research Chair in Formal Verification of Microelectronics Systems. In 2007, he was appointed Senior Concordia University Research Chair in Formal Verification of System-on-Chip. He has made contributions and published papers in the areas of formal hardware verification, microprocessor and system-on-chip verification, VLSI design automation, and communication architectures and protocols.

Dr. Tahar is a Professional Engineer in the Province of Quebec. He has been organizing and involved in various international conference program committees as well as national research grant selection committees. In 1998, he received a Canada Foundation for Innovation (CFI) Researcher Award. In 2007, he was named University Research Fellow upon receiving the University Research Award.