An Improved Publicly Detectable Watermarking Scheme Based on Scan Chain Ordering

Aijiao Cui and Chip-Hong Chang

Centre for High Performance Embedded Systems, Nanyang Technological University,
50 Nanyang Drive, Research Techno Plaza, 3rd Storey, Border X Block, Singapore 637553

Abstract—This paper proposes an improved version of watermarking scheme at the Design-for-Testability (DfT) stage for VLSI Intellectual Property (IP) Protection. The improved scheme overcomes the weaknesses of previous scan chain watermarking scheme by imposing the extra ordering constraints generated by the IP owner’s signature on all scan flip-flops impartially. IP authorship can be publicly authenticated in the field by injecting a given test vector and matching a permuted output response vector against a transformed reference pattern. Both the output response and the reference sequence are related to a pseudorandom sequence generated by a public-key cryptographic algorithm. Experimental results show that the improved method has a low probability of coincidence and low test power overhead.

I. INTRODUCTION

The System-on-Chip era is marked by the wide adoption of reusable IP cores. Trading of IP cores in an open environment poises new problems in tracking illegal IP distribution. Watermarking techniques [1]-[4] have been proposed as an active approach to protect the copyrights of VLSI designs. Based on the watermark detection, these techniques can be classified under static or dynamic watermarking schemes [5]. The dynamic watermarking [3], [4] avoids the risk of exposing the constraint generator in static watermarking [1], [2] during verification by allowing the ownership to be detected by running the watermarked IP with some specific input patterns. If the dynamic watermarking scheme is applied at the DfT stage [3], [4], the ownership can be directly detected in the field even after the IP core has been integrated into SoC and packaged.

In [3], we proposed a dynamic watermarking scheme based on the power-driven scan chain ordering problem. When a specific input vector is injected, the output response will contain the watermark at some selected scan flip-flop positions. This scheme is found to possess some weaknesses. To verify the authorship, the positions of these specific flip-flops must be known, which needs special measure to ensure its security for public authentication. Also, the test patterns distributed with the watermarked IP may create a security leak due to the biased selection of the watermarked flip-flops based on the weighted transitions of the test and response vectors. These make the watermark vulnerable to attack without removing the entire scan chain. These weaknesses arise because not all flip-flops in the scan chain have equal privileges to be allocated to every position specified by the extra constraints imposed by the watermark.

In this paper, we propose an improvement to overcome the above weaknesses in our DfT watermarking scheme [3]. Both the switching power minimization criterion and the extra constraints generated by the watermark information are considered for a more ‘random’ allocation of all flip-flops in the scan chain instead of giving some flip-flops the prerogative to assumed positions dictated by the specific output response vector [3]. A public-key cryptosystem is also used to assure that the authorship can be authenticated publicly without divulging the watermark information.

II. REVIEW OF PREVIOUS WORK

This section reviews our earlier watermarking scheme based on scan chain ordering [3]. The example in [3], as shown in Fig. 1, is used for the illustration. At the DfT stage, the extra constraints are imposed on the ordering of the flip-flops so that the watermark bits, \( w_1, w_2 \) and \( w_3 \) will be extracted from some reordered flip-flop positions, \( mp_1, mp_2 \) and \( mp_3 \), which are randomly generated by the signature. The watermarking process finds a permutation, \( \pi \), of the scan flip-flops that will minimize the test switching power. When a specific input test vector is applied onto the scan-in pin, \( S_i \), the watermark bits will be detected at positions \( mp_1, mp_2 \) and \( mp_3 \) of its output response vector. The watermarked scan chain, \( \pi(w)(R) \), of the scan flip-flops that will minimize the test switching power. When a specific input test vector is applied onto the scan-in pin, \( S_i \), the watermark bits will be detected at positions \( mp_1, mp_2 \) and \( mp_3 \) of its output response vector. The watermarked scan chain, \( \pi(w)(R) \), of the scan flip-flops that will minimize the test switching power.

To verify the authorship, the IP owner must be able to show that the bits extracted from \( p_4, p_5 \) and \( p_6 \) match the bits \( w_1, w_2 \) and \( w_3 \). By revealing the positions hosting the watermark bits publicly for authentication, it is easier for the attacker to erase the watermark information on those flip-flops without removing the scan chain. The watermark constraint is imposed by a power-driven scan chain ordering algorithm. Based on the number of weighted transitions [6], a heuristic Nearest Neighbor (NN) greedy algorithm is used to reallocate flip-flops into the scan chain. For the positions that host the watermark bits, the flip-flops are constrained to be selected from a subset of the unallocated flip-flops according to the values of the watermark bits. For example, in Fig. 1, as \( p_6 \) is a watermarked position, the search for the nearest neighbor to the flip-flop in \( p_6 \) (which is \( r_2 \) in this case) is limited only to the scan flip-flops \( r_1, r_4 \) and \( r_5 \). However, from the weighted transitions of the test patterns, the nearest neighbor to \( r_2 \) is actually \( r_3 \), which is not a member of \( \{r_1, r_4, r_5\} \). Thus, from the test patterns, if the flip-flop is not the real nearest neighbor to the preceding flip-flop in the watermarked scan chain, the watermarked flip-flop and its information will be exposed, making the
scheme vulnerable. These weaknesses will be overcome by the improved scheme proposed in the next section.

III. IMPROVED WATERMARKING SCHEME

Let \( R = \{ r_i \}_1^N \) be the original scan chain with \( N \) scan flip-flops and \( P = \{ p_i \}_1^N \) be the \( N \) flip-flop positions. A weighted connected graph, \( G(\mathcal{V}, \mathcal{E}) \) can be built from the complete test set. Each vertex of the graph represents a scan flip-flop. An edge \( E(r_i, r_j) \) connecting two flip-flops, \( r_i \) and \( r_j \) carries a weight equal to the total number of bit differences between \( r_i \) and \( r_j \) for the entire test set. The watermark is embedded to find a permutation \( \pi_{wm}(R) \) with minimal test switching power such that a specific output vector can be obtained when a user-specific test vector is applied onto \( S_{in} \). The watermarking process is detailed as follows:

A. Watermark Generation

The IP ownership is displayed by a meaningful text string \( M \). This arbitrary length binary string is shortened to a fixed length hash-code, \( H \), using the secure hash algorithm, SHA-1 [7]. Directed by \( H \), a pseudorandom number generator (PNG) [7] is used to generate a sequence of \( N \) unique numbers, \( Q = \{ q_i \}_1^N \), \( 1 \leq q_i \leq N \). This is the watermarked constraint used for the scan chain reordering.

B. Generation of Reference Output Vector

A designated input vector, \( X = \{ x_i \}_1^N \), \( x_i \in \{0, 1\} \), is shifted into the initial scan chain to obtain an output vector, \( Y = \{ y_i \}_1^N \), \( y_i \in \{0, 1\} \). Suppose there are \( N_0 \) “0” bits and \( N_1 \) “1” bits in \( Y \). The \( N_0 \) and \( N_1 \) flip-flops are assigned to two sets, \( R_0 \) and \( R_1 \), respectively. Let \( Z \) be an \( N \)-bit binary vector where the first \( N_0 \) bits are “0” and the remaining \( N_1 \) bits are “1”, i.e., \( Z = \{ z_i \}_1^N \), \( z_i = 0 \) for \( 1 \leq i \leq N_0 \) and \( z_i = 1 \) for \( (N_0+1) \leq i \leq N \). \( Q \) is encrypted using a private key, \( K_e \), of a public-key cryptosystem (PKC) [7] to generate a random number sequence, \( C = f_{\phi}(K_e, Q) = \{ c_i \}_1^N \), \( c_i = q_i \), \( 1 \leq j \leq N \). The bit sequence \( Z \) is then ordered according to \( C \) to generate \( U = \{ u_i \}_1^N \), where \( u_i = z_i \).

C. Watermark Insertion

During the watermarking process, the allocation of a flip-flop to a position in the scan chain is directed by the integer sequence \( Q \) and the binary vector \( U \). Let \( p_j = \pi_{wm}(r_i) \) denote the mapping of a vertex, \( r_i \in R \), to a position, \( p_j \in P \). The first position, \( p_j \), will be arbitrarily allocated to any flip-flop from \( R_0 \), where \( \phi \in \{0, 1\} \) and \( \phi \neq u_k \). Then the allocated flip-flop, \( \pi_{wm}^{-1}(p_j) \), is removed from \( R_0 \). For all other positions, \( p_i \), \( 2 \leq i \leq N \), the NN algorithm is used to select a flip-flop from \( R_0 \), \( \phi \neq u_k \), that has the least edge cost in the graph \( G \) to the vertex \( \pi_{wm}^{-1}(p_{i-1}) \). The selected flip-flop \( \pi_{wm}^{-1}(p_i) \) is then removed from the set \( R_0 \). This process is repeated until every scan flip-flop in \( R \) has been allocated to a unique position in the scan chain. The watermark insertion procedure is summarized by the pseudo code in Fig. 2.

D. Public and Field Verification of Authorship

The verifier is given \( X' = \pi_{wm}(X) \), the number sequence, \( C \) and the public key of the IP owner, \( K_d \), for authorship authentication. A response vector, \( Y' \), is obtained by loading \( X' \) onto the watermarked scan chain in the test mode. If \( \hat{N}_0 \) and \( \hat{N}_1 \) are numbers of “0” and “1” bits of \( Y' \), respectively, an \( N \)-bit binary vector, \( \hat{Z} \) with \( \hat{N}_0 \) “0”’s followed by \( \hat{N}_1 \) “1”’s is created and permuted according to the order of \( C \) to obtain a reference sequence, \( \hat{U} \). \( \hat{C} \) is decrypted using the public key, \( K_d \) to obtain \( Q = f_d(K_d, \hat{C}) \). \( Q \) is then used to permute \( Y' \) to obtain a binary sequence \( U' \) such that the \( q_i \)-th element of \( \{ q_i \}_1^N \) is equal to the \( i \)-th element of \( Y' \), i.e., \( U' = \{ u'_i \}_1^N = \{ y'_i \}_1^N \) if \( U' \) perfectly matches or is highly correlated with \( \hat{U} \), the authorship is verified as \( Q \) can only be generated by the IP owner.

E. A Simple Example

Consider the scan chain in Fig. 3. \( V_i \) and \( R_i \) on the left hand side denote the \( i \)-th test and response vectors, respectively for the original scan chain, \( \pi(R) = r_1 r_2 r_3 r_4 r_5 r_6 r_7 \). The connected graph for the scan chain is shown in Fig.
4(a). Suppose $Y = \text{“0110011”}$ is obtained when $X = \text{“0010110”}$ is loaded into the scan chain. From $Y$ and $\pi(R)$, $R_0 = \{r_1, r_4, r_5\}$, $R_1 = \{r_2, r_3, r_6, r_7\}$ and $Z = \text{“0001111”}$. The vertices corresponding to the flip-flops in $R_0$ and $R_1$ are marked with dot lines and continuous lines, respectively in Fig. 4. Assume that $Q = \{3, 4, 2, 7, 5, 1, 6\}$ and $C = f_s(K_s, Q) = \{4, 1, 6, 3, 5, 7, 2\}$. Then, according to $Z$ and $C$, $U = \text{“1010110”}$.

![Fig. 3. Example of improved DfT watermarking scheme.](image)

For $u_1 = -1$, the flip-flop allocated to $p_1$ must come from $R_1$ to produce a “1” bit in the first flip-flop of the watermarked scan chain. If $r_2$ is arbitrarily selected, $R_1$ is updated to $\{r_2, r_6, r_7\}$. As $u_2 = u_4 = 0$, flip-flops from $R_0$ will be withdrawn to compete for $p_2$. From Fig. 4(b), the edge connecting $r_1$ to $r_2$ has the smallest cost among all the qualified edges connected to $r_2$, so $r_1$ is allocated to $p_2$ and removed from $R_0$. Similarly, $r_3$ is allocated to the next position, $p_3$ since $u_3 = u_2 = 0$ and $e(r_1, r_3) < e(r_1, r_4)$. The process is repeated until all flip-flops are allocated. The ordering of the final watermarked scan chain is $\pi_{wm}(R) = r_2 \ldots r_1 r_3 r_4 r_5 r_6 r_7$. This is shown in the lower part of Fig. 3. To verify the IP authorship, the vector $X = \text{“0010011”}$ is loaded into the watermarked scan chain in the test mode to obtain $Y = \text{“1000111”}$, from which $\hat{Z} = \text{“0001111”}$ is produced. According to $Y$ and the given $C$, the sequence $\hat{Z}$ is permuted to obtain $\hat{U} = \text{“1010110”}$. $\hat{C}$ is then decrypted with the public key $K_s$ to obtain $Q = \{3, 4, 2, 7, 5, 1, 6\}$. $Y'$ is permuted according to $Q$, to obtain $U'$. If $U'$ matches $\hat{U}$, the authorship is proved. Otherwise, the watermark is either not present or has been modified or erased.

![Fig. 4. Watermarking process according to the cost graph.](image)

**IV. WATERMARK RESILIENCE ANALYSIS**

The probability of coincidence, $P_c$, is a key measure of the strength of a watermarking scheme. It denotes the probability that a non-watermarked design carries the watermark by coincidence. For our proposed scheme, if the output sequence under the input vector $X'$ has the same permutation of “0” and “1” bits as $Y'$, the design is said to possess the watermark. $P_c$ can then be expressed as:

$$P_c = \frac{P_n^{10} \cdot P_n^{11}}{P_n^{2N}} = \frac{N! \cdot N!}{N!}$$

A lower $P_c$ implies a stronger authorship proof. From (1), the longer the scan chain length and the more balance between the numbers of “0” and “1” bits in $Y'$, the stronger the watermarking scheme.

False positive response occurs when $Y'$ is obtained from the injection of some input vectors other than $X'$ into the watermarked scan chain. If there are $N_T$ randomly generated test vectors are applied onto the scan chain, then we can define:

$$P_d(\tau = 1) = \frac{N_T(\tau) \cdot N_T(\tau)}{N_T}$$

where $P_d(\tau = 1)$ gives the exact false positive rate. When $\tau$ decreases, $P_d$ increases and a threshold for $\tau$ can be determined empirically that with high degree of confidence, $Y'$ will not appear in the output responses for a given number of input vectors tested. Hence the authenticity of the design can be assured as long as $P_d(\tau)$ is very low.

In what follows, four typical attacks are analyzed with Alice as the IP owner and Bob as the attacker. Variables with subscripts “A” and “B” are associated with Alice and Bob, respectively.

A. **Ghost Search**

Without touching Alice’s design, Bob may load $X'_B$ into Alice’s watermarked design during test mode to obtain $Y'_B$ and generate the sequence $Z'_B$. He may refer to an encrypted number sequence, $C_B$ to permute $Z'_B$ to generate $U_B$. Bob will then find a set of numbers, $Q_B'$ to realize the mapping of $Y'_B$ to $U_B$. This attack scenario can not be substantiated as it is computationally infeasible for Bob to find a key to decrypt $C_B$ to the sequence $Q_B$ or reverse a keyed one-way PNG to generate $Q_B$. Alternatively, Bob may first generate $Q_B$ using the PNG and then map the $Y'_B$ to a bit sequence, $U_B$ according to $Q_B$. However, it is computationally intractable for Bob to find a key pair ($K_s$, $K_B$) for the encryption of $Q_B$ and the decryption of $C_B$ [7].

B. **Denial Attacks**

Bob may deny that Alice has inserted her ownership information into the design. He may claim that Alice may have randomly found $X'_A$ and the corresponding sets of $Q_A$, $Z_A$, $C_A$ and $U_A$. To repudiate Bob’s accusation, Alice needs to repeat the verification process in Section III.D and show that the probability for the output response to a randomly generated input vector other than $X'_A$ to match $Y'_A$ is very low, i.e., $P_d(\tau = 1)$ must be very low.

C. **Removal Attacks**

It is obvious that no watermark can survive upon removal of the scan chain. The IP core of the scan chain is
The proposed watermarking scheme incurs 4% for all designs. The average overhead is less than 2%, which shows that the proposed watermarking scheme incurs reasonable overhead and without obvious degradation to the circuit performance. The switching power overhead is less than 0.4%. Therefore, the watermarking process can be used in IP protection.

The switching power overhead is less than 0.4% for all designs. The average overhead is less than 2%, which shows that the proposed watermarking scheme incurs reasonable overhead and without obvious degradation to the circuit performance. The switching power overhead is less than 0.4%. Therefore, the watermarking process can be used in IP protection.

### 5. EXPERIMENTAL RESULTS

In the experiment, the DfT tool, DFTadvisor and the ATPG tool, FastScan by Mentor Graphics are used to generate the initial vectors, Y from X on the originally optimized scan design. The watermarking scheme is applied on sequential circuits from ISCAS89, ISCAS99 and LGSynth93 benchmark suites. All calculations were performed on a 750–MHz Sun UltraSPARC-III with Solaris operating system and 2 GB of memory.

Tables 1 shows the results of watermarking 14 benchmark circuits. The set of pseudorandom numbers is generated according to the scan chain length, N. WT and WT denote the number of weighted transitions of the originally optimized scan chain and that of the watermarked scan chain, respectively. ∆WT is calculated as the percentage increments from WT to WT. A negative percentage indicates performance improvement due to watermarking. The switching power overhead is less than 4% for all designs. The average overhead is less than 2%, which shows that the proposed watermarking scheme incurs very low overhead on the test power.

The watermark strength is evaluated by PC in Table I. The value of Pc decreases rapidly as the scan chain length, N increases. When N > 1000, Pc reduced to less than 0.01. To determine the false positive rate, Pp, 200 randomly generated input vectors were applied on each watermarked design. None of the output responses was detected to perfectly match the known Y. When τ is reduced to 0.6, Pp remains zero for all designs. When τ is reduced to 0.5, all the watermarked designs have Pp ≥ 50%. So it is reasonable to assume that when more than 40% of the bits are corrupted, the watermark can not be detected. If we assumed that α = 150 flip-flops in the scan chain can be moved around with reasonable effort and without significantly degrading the circuit performance. The probability of successfully altering at least 40% of Y by a random derangement of scan flip-flops is zero as long as N > α/0.4 = 375, which can be easily satisfied by a design with a reasonable long scan chain.

<table>
<thead>
<tr>
<th>Circuit</th>
<th>N</th>
<th>Nc</th>
<th>Pc</th>
<th>WTorg</th>
<th>WTwm</th>
<th>∆WT(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>difseq</td>
<td>305</td>
<td>141</td>
<td>164</td>
<td>7.98E-91</td>
<td>8039952</td>
<td>8336818</td>
</tr>
<tr>
<td>tseng</td>
<td>385</td>
<td>183</td>
<td>204</td>
<td>4.99E-115</td>
<td>10990496</td>
<td>11356970</td>
</tr>
<tr>
<td>B15</td>
<td>449</td>
<td>219</td>
<td>230</td>
<td>0.93E-134</td>
<td>110319565</td>
<td>111431425</td>
</tr>
<tr>
<td>B21</td>
<td>490</td>
<td>255</td>
<td>235</td>
<td>1.31E-146</td>
<td>181147289</td>
<td>184424095</td>
</tr>
<tr>
<td>valu</td>
<td>495</td>
<td>225</td>
<td>270</td>
<td>1.1E-147</td>
<td>43449702</td>
<td>44909045</td>
</tr>
<tr>
<td>pmac</td>
<td>590</td>
<td>292</td>
<td>297</td>
<td>6.2E-177</td>
<td>15545981</td>
<td>15638009</td>
</tr>
<tr>
<td>s15850</td>
<td>597</td>
<td>318</td>
<td>292</td>
<td>1.1E-178</td>
<td>8185476</td>
<td>85208775</td>
</tr>
<tr>
<td>s13207</td>
<td>669</td>
<td>320</td>
<td>541</td>
<td>2.0E-200</td>
<td>41259551</td>
<td>41947977</td>
</tr>
<tr>
<td>B22</td>
<td>735</td>
<td>378</td>
<td>318</td>
<td>2.1E-179</td>
<td>82540476</td>
<td>81582612</td>
</tr>
<tr>
<td>frisc</td>
<td>886</td>
<td>421</td>
<td>465</td>
<td>7.98E-91</td>
<td>168320271</td>
<td>171775831</td>
</tr>
<tr>
<td>s38541</td>
<td>426</td>
<td>145</td>
<td>152</td>
<td>4.78E-91</td>
<td>493516003</td>
<td>493516003</td>
</tr>
<tr>
<td>s38417</td>
<td>1636</td>
<td>815</td>
<td>821</td>
<td>1.0E-300</td>
<td>703090689</td>
<td>703090689</td>
</tr>
<tr>
<td>s35932</td>
<td>1728</td>
<td>849</td>
<td>879</td>
<td>1.0E-300</td>
<td>105100815</td>
<td>105100815</td>
</tr>
</tbody>
</table>

### REFERENCES


