Hierarchical Watermarking Method for FPGA IP Protection

Tingyuan Nie, Lijian Zhou and Yansheng Li

Communication and Electronic Engineering Institute, Qingdao Technological University, Qingdao 266033, China

Abstract

With the increasing intellectual property (IP) abuse in System-on-a-Chip design, watermarking technique is playing an important role for IP protection. In this paper, we combine watermarking methods at different levels to construct a hierarchical watermarking scheme for field programmable gate array (FPGA) IP protection. We first embed the watermark into the netlist by using a look-up table–SRL transformation. Then we embed the watermark into the bitstream of the same design by using *JBits*. We test our method on three public FPGA benchmarks. The experimental results show that the overhead of watermarking is significantly reduced due to our judicious strategies. The watermark embedded at high level is well propagated to lower level. Our technique provides a robust and secure watermarking solution.

Keywords

Field programmable gate array, IP protection, Watermarking method.

1. Introduction

The fast advancing integrated circuit (IC) processing technologies have enabled the integration of full systems on a single chip, forming a new paradigm of the "System-on-a-Chip" (SoC) technology. Field programmable gate arrays (FPGAs) have started taking a significant share of the IC market. The design reuse could not only effectively decrease the design cycle but also lead to development cost reduction of SoC designs [1]. The IP reuse is widely considered to be the most economically efficient way to close the increasing gap between the development capacity of designers and the IC productivity. However, sharing IP designs poses significantly high security risks. Most of the IPs need time and effort to be designed and verified, but the IPs of the original designer can easily be copied to and exploited by the rival competitor without the need to obtain the original designer's approval. The rival competitors reverse engineer the original design with smaller cost, which seriously damages the profits of designers. According to Intel's statistics, rival competitors can save over 90% of the development costs and about half of the development time. Creators and owners of IP designs want assurances that their content will not be illegally redistributed by consumers, and consumers also want assurance that the content they buy is legitimate. Therefore, how to protect the IC intellectual property is becoming a pressing issue.

There are several mechanisms for IP protection, such as physical tagging, digital watermarking, and fingerprinting [2]. The proposed mechanisms are commonly used and effective.

Tagging: Placing an electronic tag inside a chip to identify the authenticity. The advantages of such technique are its high reliability and good tracking performance. Marsh *et al*. proposed a tagging technique to protect the application specific integrated circuit (ASIC) core [3]. They placed a security tag in the core to store signature, and detected copyright information through an external receiver. The tag only plays the role of deterrence. It may be ruined or removed once exposed.

A variant of tagging is physical unclonable functions (PUFs), a potential candidate for implementing the unique extrinsic IC identifiers. A PUF is a physical function that provides a mapping between its inputs and outputs based on the unique fluctuations in the unclonable device material properties such as timing or current [4-6].

Watermarking: Digital watermarking was originally designed as a technology used for the protection of intellectual property rights of digital media or official document [7]. The concept was expanded to IC protection later. The main idea is to embed a watermark or a digital signature into the design of the owner permanently. The watermark can be detected after the IC is manufactured. Watermarking approach is popularly used for IP protection due to its robustness.

Kahng *et al*. proposed the first constraint-based watermarking technique in which the author's watermark is mapped into a set of constraints. These additional constraints are then embedded as watermarks into the original design [8]. Cui *et al*. added constraints/redundancy to the original design in preprocessing, and acquired a watermarked design through logic synthesis, optimization, and mapping [9-11]. These methods extremely depend on the structure of the design that may result in increasing watermarking overheads. Watermarking

methods based on finite state machine (FSM) at behavioral level are also proposed [12,13]. However, the watermark can be easily removed if the adversary is aware of its input sequence and initial state. A subtle, even negligible, "post-processing" (area relocation, reroute, shield or buffer insertion) is applied to the physical design for watermark insertion [14,15]. The approach at circuit level protects ICs through injection of process variations or IC locking/activation [16,17].

Fingerprinting: Users can get IP cores with their unique identification through the fingerprinting technique. The biggest advantage of using fingerprint is to identify individual users and facilitate the tracking of IP infringement. The challenge of fingerprinting is the need to produce a large number of IP cores with the same functional and technical indicator.

The first IP fingerprinting technique in the literature was presented by Lach *et al.* [18]. The approach was based on solution partitioning. They partitioned an initial solution into a few parts and provided several different realizations for each part. One can realize a fingerprinting design with different combinations. Another research is to generate some relatively small problems based on a pre-solved solution [19]. New solutions will be produced by applying the iterative incremental optimization on the small problems. Cost for solving such small instance is usually much lower than is for the original, but when the request for different solutions is huge, the overhead cannot be ignored. Qu *et al*. proposed a technique that superimposes additional constraints on the design problem to guarantee a large amount of high-quality solutions at all levels of design process [20].

In this paper, we improve two watermarking methods at different levels by adopting judicious strategies. We combine them to construct a hierarchical watermarking scheme for FPGA IP protection. We evaluate its performance and propagation in the experiments.

This paper is organized as follows. In Section 2 is reviewed previous related works. Our proposed hierarchical watermarking method is given in Section 3. Watermarking identification is presented in Section 4. Section 5 presents the experimental results, Section 6 analyzes the security of the watermarking method, and the concluding remarks appear in Section 7.

2. Related Work

There are mainly two kinds of FPGA IP protection techniques available in the open literature. One is constraint-based watermarking method and the other one is additive watermarking method. Constraint-based watermarking techniques represent a signature as a set of additional constraints which are applied to hardware optimization and synthesis problem. Additive watermarking methods are watermarking procedures, where a signature is added to the IP core. The watermark is not embedded into the function of the core; yet, it can be masked as a part of the functional part.

Constraint-based watermarking methods were proposed in Refs [8,21,22]. Lach *et al*. introduced the concept of FPGA watermarking and carried out a series of studies. The essence of their approaches was to embed the encoded watermarks into unused look-up tables (LUTs) such that do not affect the original design. They further concealed the watermarking by rerouting design regions around these LUTs. The disadvantage of this approach is that the watermark embedded is not a functional part of the design, so it is easy to be removed. Later, Lach *et al*. proposed other improvement schemes to refine the robustness of watermarking method.

Additive watermarking methods were applied as follows. Saha *et al*. presented a watermarking strategy by subdividing the LUT locations into sets of 2 × 2 tiles on FPGA bitfiles [23]. The number of used LUTs in a set was used as a signature. From an initial level, additional LUTs were added to achieve the fill level according to the signature. Their inputs and outputs were connected to the unconsidered neighboring cells. Liang *et al*. proposed an FSM-based watermark algorithm at behavioral level [24]. The algorithm extracted the maximal delay state set through state transformation relations among circuit signals. The watermark was mapped into additional delay constraint sequence by constraint generator, and the value in the sequence was added to the maximal delay state set in the circuit. Jain *et al*. proposed a zero overhead watermarking technique based on timing constraints [25]. This approach selected certain nets to embed watermarks by modifying their time constraints. Schmid *et al*. presented an approach at netlist level that watermarks FPGA designs by restricting the dynamically addressable part of logic tables to insert signature bits. They prevented deletion by converting functional LUTs to LUT-based Random access memories (RAMs) or shift registers (SRs) [26]. Zhang *et al*. proposed a zero-overhead watermarking technique by using content-copy method (CCM) to group the watermark and embedding the mark into the ILUTs (not fully used LUTs) [27]. Ziener *et al*. proposed a power watermarking method that detects signature (watermark) at the power supply pins of the FPGA [28,29]. They integrated the signature into functional parts of the watermarked core, and detected it from a voltage trace with high reliability.

For hierarchical watermarking protection, Rashid *et al*. proposed a method to watermark finite impulse response (FIR) digital filter cores at different levels [30]. A unique

watermark was embedded by altering filter coefficients during algorithmic level, and sequentially embedded using circuit transformation during architectural level. Charbon *et al*. introduced a hierarchical watermarking scheme based on a generic approach that can be used at different design levels [31,32]. The watermark insertion and detection was based on the construction of topological transition of the design.

3. Hierarchical Watermarking Method

Many effective watermarking methods have been proposed at different abstraction levels of FPGA design. However, the weakness of watermarking methods implemented at one abstract level makes it vulnerable. If watermarking at one level is broken, the design is at the risk of being counterfeited. We propose a hierarchical watermarking prototype which embeds the watermark both at netlist design level and bitstream design level. The proposed method is resilient to tampering and forging due to its inherent nature.

3.1 Watermark Generation

Watermark generation is a critical step in our proposed watermarking approach. We propose a generation method that produces watermark with enough pseudo-random. A signature provided by the IP owner is firstly hashed into a shorter text. Then the text is encrypted to create a "seed" by RSA algorithm. Finally, the seed is sent to RC4 algorithm to generate a watermark.

3.2 Watermarking at the Netlist Level

An FPGA design implemented by a hardware description language (HDL) should be synthesized to a netlist file. There are a large number of unused LUTs distributed in the netlist file. However, it is not safe to simply embed watermarks into unused LUTs. In fact, there are free input ports even in a used LUT. This implies there is free storage space in the LUT that can be used for watermarking. It is difficult to perceive that such LUTs were embedded watermark or not.

An FPGA device is mainly formed by logic elements (LEs). An LE contains at least one LUT which is essentially a RAM. A four-input LUT can be considered as a 16 × 1-bit RAM. If watermarks are directly embedded into LUTs, it probably eliminated by the optimization of electronic design automation (EDA) tools. Fortunately, an LUT in Xilinx can be configured as an SR. The optimization tools would not optimize some special components such as SRs. Watermarks stored in SRs will safely remain. It is a good choice that one writes watermarks into free spaces of LUTs and converts LUTs to SRs. But the selection of LUTs should be carefully done in order to not degrade the performance of the FPGA design.

An LUT in a netlist file generally has four inputs and one output to store 16-bit length information. In Figure 1, we illustrate a two-input OR function unit implemented using an LUT. In this logic, inputs I3 and I4 are utilized, while I1 and I2 are idle. The output takes up four bits of the table (displays in the "O" line), the remaining 12-bit space is available to embed watermarks.

Firstly, we select LUTs with idle inputs to insert watermark. Notably, the selection is carefully done to avoid performance degradation of the design. It follows two strategies: (i) the LUT must not be in a critical path and (ii) the ratio of the number of idle inputs must be larger than a constraint l (in this paper, l = 0.3). The selected LUTs are then configured into shift register LUTs (SRLs) . The configuration is illustrated in Figure 2. Ports of I3, I4, and Q are connected to adjacent LUTs, and SHIFTIN (D) is connected to SHIFTOUT (Q) to realize the functionality of the OR logic. In order to further conceal the place of watermarks, remaining inputs are set to low power, e.g., ground (GND). Watermarks as well as original contents of the LUT are written into the SRL by an initialization

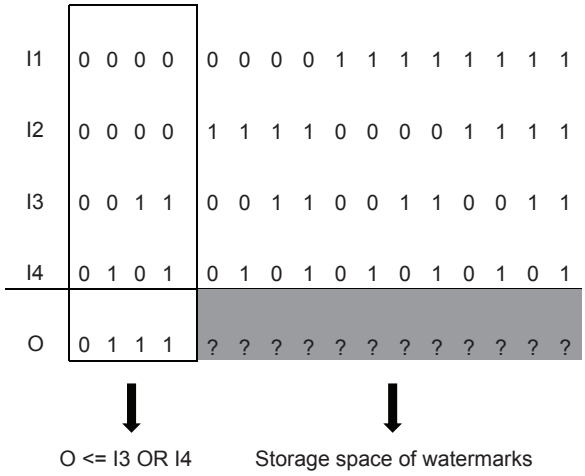


Figure 1: Logic table of a two-input LUT.

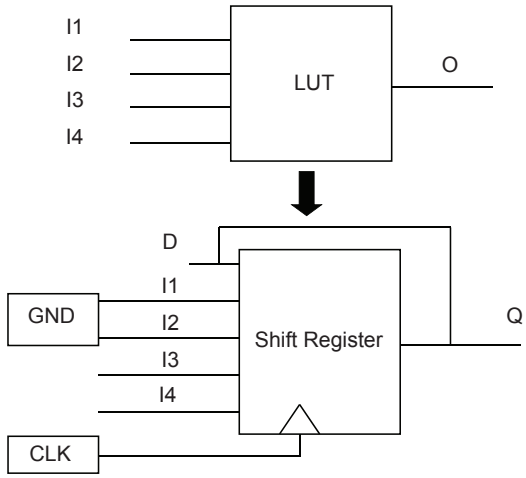


Figure 2: An example replacing an LUT with a shift register.

process. The watermarking process at netlist level is implemented by the LUT–SRL substitution.

3.3 Watermarking at the Bitstream Level

In order to provide a better IP protection solution, we implement the following IP protection at bitstream level to complete the hierarchical watermarking scheme. The idea of watermarking at bitstream level is to write the same watermark into the margin of configurable logic block (CLB) in the bitstream file. As a result, the final watermarked FPGA design contains the watermark propagated from the netlist level as well as the watermark embedded at the bitstream level. Thus, the proposed method provides a more secure watermarking scheme. Users can select the bitstream encryption option on FPGA to further protect the IP core.

An ordinary file editor cannot read or write any information from the bitstream of an FPGA design. Fortunately, there is a convenient Xilinx tool called *JBits* which can manipulate the bitstream file [33]. *JBits* is an Application Program Interface (API) to the Xilinx configuration bitstream. This API permits Java applications to dynamically modify Xilinx Virtex-II bitstreams. One may choose using more suitable tools like FPGA Editor to edit the bitstream file. *JBits* performing reconfigurations can be compiled and run very quickly. We use the command "ReadLUTs" to read the content of the LUT. We use the command "WriteLUTs" to overwrite the selected LUT when the watermark is attached to the original content.

In detail, a bitstream file of Xilinx device is structured in packets. Each packet is loaded configuration data of the circuit to instantiate into the FPGA. A packet may consist of one or more configuration words to control the configuration process. A packet is sliced into the smallest configuration unit: Frame. An ordinary Xilinx Virtex-II packet is constructed by two or four slices. Each slice has two LUTs, G-LUT and F-LUT, as shown in Figure 3. A four-input LUT stores 16-bit data; therefore, a packet can be loaded into 64-bit data. We duplicate the watermark and embed it in several places. The strategy makes the watermark more difficult to be perceived and removed.

3.4 Watermarking Algorithm

The proposed hierarchical watermarking algorithm consists of 11 steps, described as follows:

- Step 1: Synthesize an FPGA design to acquire the netlist file by using EDA tools
- Step 2: Generate the watermark by encrypting signature of IP owners
- Step 3: Generate a number sequence for different user by using a random number generator and different seed

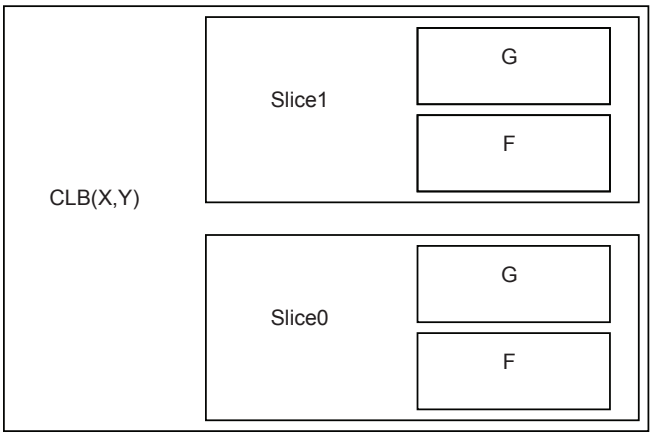


Figure 3: The structure of CLB.

- Step 4: Duplicate the watermark into a few copies and assign them to different LUT locations according to user's number sequence
- Step 5: Confirm the assigned LUT is appropriate to insert watermark or not
- Step 6: Configure the LUT into SRL, attach watermarks to original contents, and rewrite the original LUT by the SRL initialization
- Step 7: Repeat 5 and 6 until watermarks are inserted totally
- Step 8: Compile the watermarked netlist file to the bitstream file
- Step 9: Use *JBits* to extract LUT contents from the bitstream file
- Step 10: Embed watermarks into the margin of LUTs
- Step 11: Generate the final watermarked IP core in form of bitstream file and record the watermarking information for detection.

4. Watermarking Identification

When there is an infringement complaint of IP reuse, the owner must extract signature from the IP core to state for his copyright.

Ziener *et al*. introduced a watermark detection method through the power supply pins of FPGA [28]. Zhang *et al*. presented a fast authorship verification method by extracting the contents of the watermarked LUTs from a bitstream file [34]. Generally, the user provides an FPGA IP core in the form of encrypted bitfile. When the bitfile is decrypted, watermarks can be extracted from specific LUTs recorded in previous watermarking process. In this paper, we use two methods to extract watermarks from the bitfile. The first method uses the proposal of Ref. [34] to extract watermarks directly at bitstream level. The second method is to reverse engineer the design to netlist file by the tools [35] and extract watermarks by the SRL–LUT substitution. In practice, the first method is more effective for FPGA

IP identification. As long as the watermark is extracted at any level, one can dispute the copyright attribution of the IP.

5. Experimental Results

The proposed watermarking method is tested on Xilinx ISE 9.1 FPGA design platform, using a workstation with a 3.3 GHz CPU and 4 GB memory. We implemented the watermarking scheme and evaluated its performance at netlist level and bitstream level, respectively.

In the experiment, we chose the Virtex-II family of Xilinx device. We selected three public FPGA cores, data encryption standard (DES), Controller Area Network (CAN), and AES, as test bench [36]. Among them, DES and advanced encryption standard (AES) are two crypto algorithm cores, while CAN is a control network protocol from Bosch. Their properties are shown in Table 1. The table lists the number of LUTs and the minimum clock cycle duration (MCCD) for each IP core.

We embedded four sizes of watermark into each IP core at the netlist level: 64 bits, 128 bits, 256 bits, and 512 bits. Table 2 shows the watermarking results. Column 2 displays the different watermark sizes for each core. Column 3 shows the increased percentage of LUT utilization; the overhead of LUT utilization is trivial, from 0% to 0.92%, which can be neglected. Column 4 shows the increased percentage of MCCD, from −1.48% to 12.48%. The result indicates that the longer watermark is embedded, the more resource needs. The MCCD shifts irregularly from negative to positive due to LUT–SRL conversion. The varying ratio of MCCD of DES is much larger than the other two cores because of the smaller chip size. Notably, one can select an engineering change order (ECO) mode to optimize the watermarked FPGA design if there are some violations to design rules.

We continued the watermarking process when the watermarked netlist was compiled into the bitstream. Table 3 gives the final results of the hierarchical watermarking. We use Hamming distance to measure transpositions required to transform the original design to the watermarked design. The larger the Hamming distance, the more distinct is the watermarked design. The result of 512-bit watermarking is shown in Columns 3 and 4. Notably, the Hamming distance percentage, which indicates the ratio of Hamming distance and core size, is 50.90%, 39.00%, and 43.60%. We think that the watermarking solution provided by our method is distinct enough from the original one. The coincidence probability (P_c) indicates the probability of a non-watermarked solution carrying the watermark by coincidence. The probability of selecting M LUTs to be watermarked from N candidates is given by P_M^N. Let l indicate the length of the watermark. The P_c in the proposed method is given as:

$$P_c = P_M^N \times (P_0)^t \times (P_1)^{l-t} = P_M^N (\frac{1}{2})^l, \quad (1)$$

where P_0, P_1 is the probability that a selected bit. It is to be watermarked to "0" or "1," approximately $P_0 = P_1 \approx 0.5$, and t is the number of bits that are watermarked to value "0." As shown in Column 4, the value of coincidence probability is very low. It indicates our method provides a very distinct watermarking solution.

We compared the watermarking overhead of our method with that of netlist-level watermarking approach proposed in Ref. [26]. The results are shown in Table 4. It is clear that our method reduced the overhead significantly, compared with the method of the reference. The average resource utilization (LUT) decreased about 95.66%, while the average time overhead (MCCD) decreased about 85.75%. This is due to our judicious watermarking strategy.

Table 1: Parameters of experimental IP cores

IP core	LUT	LUT	MCCD (ns)
DES	618	618	8.416
CAN	1467	1467	8.9
AES	2154	2154	11.665

IP – Intellectual property; LUT – Look-up table; MCCD – Minimum clock cycle duration; DES – Data encryption standard; CAN – Controller area network; AES – Advanced encryption standard

Table 2: Watermarking results at netlist level

IP core	Watermark (bit)	LUT (%)	MCCD (%)
DES	64	0.16	11.49
	128	0.16	12.48
	256	0.16	9.45
	512	0.17	10.06
CAN	64	0.31	−1.48
	128	0.43	2.09
	256	0.92	5.99
	512	0.67	6.02
AES	64	0	4.91
	128	0.05	4.33
	256	0.05	5.24
	512	0.08	5.06

IP – Intellectual property; LUT – Look-up table; MCCD – Minimum clock cycle duration; DES – Data encryption standard; CAN – Controller area network; AES – Advanced encryption standard

Table 3: The hierarchical watermarking results

IP core	Watermark (bit)	Core size	Hamming distance	Hamming distance (%)	*Pc*
DES	512	79104	40262	50.90%	2.88E-196
CAN	512	187776	73268	39.00%	9.65E-210
AES	512	275712	120223	43.60%	3.50E-220

IP – Intellectual property; DES – Data encryption standard; CAN – Controller area network; AES – Advanced encryption standard

Table 4: Watermarking overhead comparison

Watermarks (bit)	Resources (%)			Timing (%)		
	Our method	Reference [26]	Improvement (%)	Our method	Reference [26]	Improvement (%)
64	0.157	2.794	94.40%	4.973	38.424	87.06%
128	0.213	4.959	95.70%	6.3	50.779	87.59%
256	0.377	9.463	96.02%	6.893	38.227	81.97%
Average	0.249	5.739	95.66%	6.055	42.477	85.75%

We identified the watermark using aforementioned methods at the bitstream level. The extracted watermark is intact. We also extracted the watermark from bitstream file to identify the propagation of the watermark embedded at netlist level. As shown in Figure 4, if only one copy of the watermark is embedded at netlist level, it propagates more than 98% to bitstream level. If two copies are embedded, the watermark propagates entirely to bitstream level.

The simulation result shows that our method has no power overhead compared with the original design.

The experimental result shows that our technique provides a robust and secure solution for FPGA IP protection. The technique is implemented with little resource and timing overhead, and has a good propagation from high level to lower level.

6. Security Analysis

The proposed method is secure against removal attacks and robust against partial mark removal. It is difficult for adversary to remove watermarks because he cannot know the correct watermarking location. Even if he partly removed watermarks from the FPGA design, it is possible to extract the intact watermark because our technique embeds duplicated watermarks proportionately.

Adversary may also try to tamper with the watermarked solution by removing original signature and adding his own signature. The adversary is hard to resolve the problem from the scratch particularly our method occurred relatively early in the process. Also note that since the adversary does not know which parts of the design correspond to the author's signature, tampering attacks might not be able to ruin the proof of authorship before the design quality is significantly degraded.

Adversary may attempt to subvert the original watermark by inappropriately watermarking other solutions with the watermark. He needs a signature which he can convince others belong to the IP owner. Such attacks are prevented by our secure watermark generation scheme. Any attempt to forge the original signature becomes impossible.

Adversary may steal the IP and claim it as his own. To be convincing, he must find a ghost signature that yields a sufficiently convincing value. The simplest attack is the brute force attack, which is to find a signature that yields a convincing proof of authorship. This attack becomes computationally infeasible when the coincidence probability is sufficiently low, as shown in Table 3. Adversary may choose a differential attack in which he attempts to find a signature that corresponds to a set of constraints. It requires reversing all the secure cryptographical functions, which is considered to be a practically impossible task.

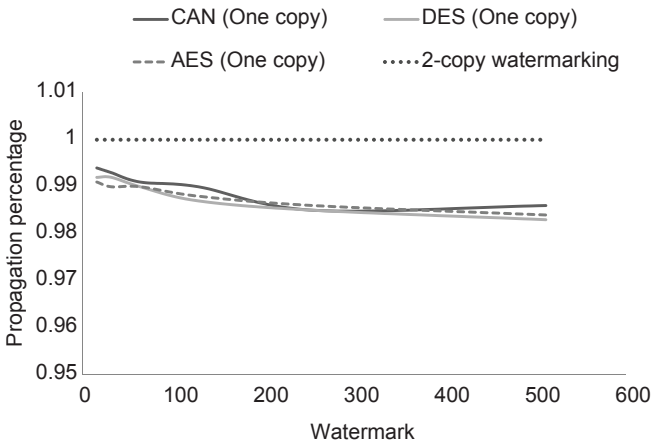


Figure 4: Propagation of watermarking.

7. Conclusions

We presented a hierarchical watermarking method for FPGA IP protection. At netlist level, we converted the used LUT to the LUT-based SR strategically and embedded the watermark by an LUT–SRL substitution. At bitstream level, we wrote the watermark into selected LUTs by an EDA tool *JBits*. The strategy of the hierarchal watermarking provides a robust and secure solution for IP protection. The embedded watermark is propagated entirely through the design flow. The experimental result shows that the overhead of resource and timing is significantly reduced, compared with the related work.

8. Acknowledgment

The project is sponsored by SRF for ROCS, SEM and Shandong Province Higher Educational Science and Technology Program (J09LG10).

References

1. Y. Oshima, "Legal protection for semiconductor intellectual

property (IP),” in IEEE Asia and South Pacific Design Automation Conference, Kitakyushu, Japan, pp. 551-5, Jan. 2003.

2. Virtual Socket Interface Alliance, “Intellectual Property Protection White Paper: Schemes, Alternatives and Discussion Version 1.0,” Sep. 2000.
3. C. Marsh, and T. Kean, “A security tagging scheme for ASIC designs and intellectual property cores,” in IP SoC (IP-Based System Design), pp. 6-7, 2006.
4. M. Majzoobi, F. Koushanfar, and M. Potkonjak, “Techniques for design and implementation of secure reconfigurable PUFs,” ACM T-RETS, Vol. 2, no. 1, pp. 1-33, 2009.
5. M. Potkonjak, S. Meguerdichian, and A. Nahapetian, “Differential public physically unclonable functions: Architecture and applications,” in Design Automation Conference, San Diego, USA, pp. 242-7, June. 2011.
6. M. Majzoobi and F. Koushanfar, “Time-Bounded Authentication of FPGAs,” IEEEE T-IFS, Vol. 6, no. 3, pp. 1123-35, 2011.
7. W. Hongxia, and L. Changxing, “JPEG images authentication with discrimination of tampers on the image content or watermark,” IETE Technical Review, Vol. 27, no. 3, pp. 244-51, May. 2010.
8. A. B. Kahng, J. Lach, W. H. Mangione-Smith, S. Mantik, I. L. Markov, and M. Potkonjak, *et al*., “Constraint-Based Watermarking Techniques for Design IP Protection,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 20, no. 10, pp. 1236-52, Oct. 2001.
9. A. Cui, C. H. Chang, and S. Tahar, “IP watermarking using incremental technology mapping at logic synthesis level,” IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., Vol. 27, no. 9, pp. 1565-70, Sep. 2008.
10. M. Khan, and Tragoudas S, “Rewiring for watermarking digital circuit netlists,” IEEE Trans of Computer-Aided Design of Integrated Circuits and Systems, Vol. 24, no. 7, pp. 1132-7, 2005.
11. K. T. Cheng, L. A. Entrena, “Combinational and sequential logic optimization by redundancy addition and removal,” IEEE Trans. In Computer-Aided Design of Integrated Circuits and Systems, Vol. 14, no. 7, pp. 909-16, 1995.
12. F. Koushanfar, and Y. Alkabani, “Provably secure obfuscation of diverse watermarks for sequential circuits,” in Hardware-Oriented Security and Trust (HOST), Anaheim, CA, USA, pp. 42-7, Jun. 2010.
13. A. Cui, C. H. Chang, S. Tahar, and A. T. Abdel-Hamid, “A Robust FSM Watermarking Scheme for IP Protection of Sequential Circuit Design,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 30, no. 5, pp. 678-90, 2011.
14. T. Nie and M. Toyonaga, “An Efficient and Reliable Watermarking System for IP Protection,” IEICE Trans. Fundamentals, Vol. E90-A, no. 9, pp. 1932-9, Sep. 2007.
15. G. Sun, Z. Q. Gao, and Y. Xu, “A Watermarking System for IP Protection by Buffer Insertion Technique,” in International Symposium on Quality Electronic Design (ISQED), CA, USA, pp. 671-5, 2006.
16. J. A. Roy, F. Koushanfar, and I. L. Markov, “Protecting Bus-based Hardware IP by Secret Sharing,” in Design Automation Conference (DAC), CA, USA, pp. 846-51, 2008.
17. W. P. Griffin, A. Raghunathan, and K. Roy, “CLIP: Circuit Level IC Protection Through Direct Injection of Process Variations,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 20, no. 5, pp. 791-803, 2012.
18. J. Lach, W. H. Mangione-Smith, and M. Potkonjak, “FPGA Fingerprinting Techniques for Protecting Intellectual Property,” in IEEE 1998 Custom Integrated Circuits Conference, CA, USA, pp. 299-302, May 1998.
19. A. E. Caldwell, H. Choi, A. B. Kahng, S. Mantik, and M. Potkonjak, *et al*., “Effective Iterative Techniques for Fingerprinting Design IP,” in 36th ACM/IEEE Design Automation Conference Proceedings, LA, USA. p. 843-8, 1999.
20. G. Qu and M. Potkonjak, “Fingerprinting Intellectual Property Using Constraint Addition,” in 37th ACM/IEEE Design Automation Conference, CA, USA, pp. 587-92, 2000.
21. J. Lach, W. H. Mangione-Smith, and M. Potkonjak, “Signature Hiding Techniques for FPGA Intellectual Property Protection,” in IEEE/ACM International Conference on Computer Aided Design, CA, USA, pp. 186-91, Nov. 1998.
22. A. B. Kahng, J. Lach, W. H. Mangione-Smith, S. Mantik, I. L. Markov, and M. Potkonjak, *et al*., “Watermarking techniques for intellectual property protection,” in Design Automation Conference, CA, USA, pp. 776-81, 1998.
23. D. Saha, and S. Sur-Kolay, “Fast Robust Intellectual Property Protection for VLSI Physical Design,” in 10th International Conference on Information Technology. Washington, DC, USA: IEEE Computer Society, pp. 1-6, 2007.
24. W. Liang, X. Sun, Z. Ruan, and J. Long, “The Design and FPGA Implementation of FSM-based IP Watermark Algorithm at Behavioral Level,” Information Technology Journal, Vol. 10, no. 4, pp. 870-6, 2011.
25. A. K. Jain, L. Yuan, P. R. Pari, and G. Qu, “Zero overhead watermarking technique for FPGA designs,” in GLSVLSI, Washington, DC, USA, pp. 147-52, 2003.
26. M. Schmid, D. Ziener, and J. Teich, “Netlist-Level IP Protection by Watermarking for LUT-Based FPGAs,” in IEEE International Conference on Field-Programmable Technology (FPT 2008), Taipei, Taiwan, pp. 209-16, 2008.
27. J. Zhang, Y. Lin, Q. Wu, and W. Che, “Watermarking FPGA Bitfile for Intellectual Property Protection,” Radioengineering, Vol. 21, no. 2, pp. 764-71, 2012.
28. D. Ziener, and J. Teich, “Power Signature Watermarking of IP Cores for FPGAs,” Journal of Signal Processing Systems, Vol. 51, no. 1, pp. 123-36, Apr. 2008.
29. D. Ziener, F. Baueregger, and J. Teich, “Multiplexing Methods for Power Watermarking,” in IEEE Int. Symposium on Hardware-Oriented Security and Trust (HOST 2010), Anaheim, USA, pp. 36-41, Jun. 2010.
30. 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, pp. 39-42, May 1999.
31. E. Charbon, “Hierarchical Watermarking in IC Design,” in IEEE Custom Integrated Circuits Conference, Santa Clara, California, USA, pp. 295-8, May. 1998.
32. E. Charbon, and I. Torunoglu, “Intellectual Property Protection Via Hierarchical Watermarking,” in International Workshop on IP Based Synthesis and System Design, Grenoble, France, Dec. pp. 776-81, 1998.
33. Xilinx Inc. JBits SDK. Available from: http//www.xilinx.com/labs/projects/jbits/[Last accessed date 2010 Jun].
34. J. Zhang, Y. Lin, W. Che, Q. Wu, Y. Lu, and K. Zhao, “Efficient verification of IP watermarks in FPGA designs through lookup table content extracting,”, IEICE Electronics Express, Vol. 9, no. 22, pp. 1735-41, 2012.
35. F. Benz, A. Seffrin, and S. A. Huss, “Bil: A tool-chain for bitstream reverse-engineering,” in 22nd International Conference on IEEE Field Programmable Logic and Applications (FPL), pp. 735-8, 2012.
36. Avaialbel from: pencores.org., http://opencores.org [Last accessed date 2011 Jul].

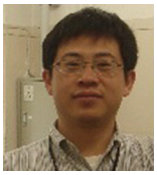
AUTHORS

Tingyuan Nie received B.S. degree in computer science from Wuhan University of Technology in 1993, and M.S. degree and Ph.D. in computer engineering degree from Kochi University, Japan in 2005 and 2008, respectively. He is an associate professor with the Communication and Electronic Engineering Institute, Qingdao Technological University, Qindao, China. His research interests include VLSI CAD algorithms, watermarking, and fingerprinting methodology for IPP, and IC security. He is a member of IEEE for 11 years by 2013.

E-mail: tynie@qtech.edu.cn

Lijian Zhou received B.S. degree and M.S. degree in the School of Computer and Information Technology from Northeast Petroleum University of Technology in 1993 and 1996, respectively, and Ph.D. in computer engineering degree from Ocean University of China in 2007. She is an associate professor with the Communication and Electronic Engineering Institute, Qingdao Technological University, China. Her research interests include image processing and watermarking methodology for IPP.

E-mail: zhoulj_qd@163.com



Yansheng Li received B.S. degree and M.S. degree in information engineering from Qingdao University in 1997, and M.S. degree and Ph.D. in information system degree from Kakojima University, Japan in 2005 and 2008, respectively. He is an associate professor with the Communication and Electronic Engineering Institute, Qingdao Technological University, China. His research interests include IC integration and IC security.

E-mail: liyansheng@qtech.edu.cn

DOI: 10.4103/0256-4602.123116; Paper No. TR 41_13; Copyright © 2013 by the IETE