Study on IP Protection Techniques for Integrated Circuit in IOT Environment

Wei Liang, Jing Long, Dafang Zhang, Xiong Li and Yin Huang

Abstract The growth of electronic chip technique has led to frequent occurrence of intellectual property (IP) disputes. It seriously affects rapid and healthy development of semiconductor industry. To address the disputes, many IP protection methods are proposed in these years, such as IP watermarking. It is a novel technique to hide secrets in IP core to prove original ownership. This chapter focuses on two issues: how to hide secrets in IP circuit and how to authenticate IP ownership. Four types of IP watermarking methods will be concretely introduced in this chapter. (1) FPGA based IP watermarking technique. (2) FSM based IP watermarking technique. (3) DFT based IP watermarking technique. (4) Self-recoverable dual IP watermarking technique. The experiments show that the proposed schemes have low resource overhead by comparing with other schemes. Meanwhile the resistance to attacks of the watermark is encouraging as well.

1 Introduction

With the rapid development of internet of things (IOT), more and more transistors can be integrated into a single chip. The number has exceeded 10 billion in 2015. In Fig. 1, the complexity growth rate improves by 58% every year, but the productivity only grows by 21%. There is an increasingly deeper gap between chip-making capacity and design capacity. So, component based IP design method becomes prevalent due to its high efficiency [1]. IP reuse technology belongs to this type of design method.

W. Liang (✉)
Department of Software Engineering, Xiamen University of Technology, Xiamen 361024, China
e-mail: idlink@163.com

J. Long · D. Zhang
College of Computer Science and Electronic Engineering, Hunan University, Changsha, Hunan 410082, China

X. Li · Y. Huang
School of Computer Science and Engineering, Hunan University of Science and Technology, Hunan 411201, Xiangtan, China

© Springer Nature Singapore Pte Ltd. 2018

B. Di Martino et al. (eds.), *Internet of Everything*, Internet of Things,
https://doi.org/10.1007/978-981-10-5861-5_9

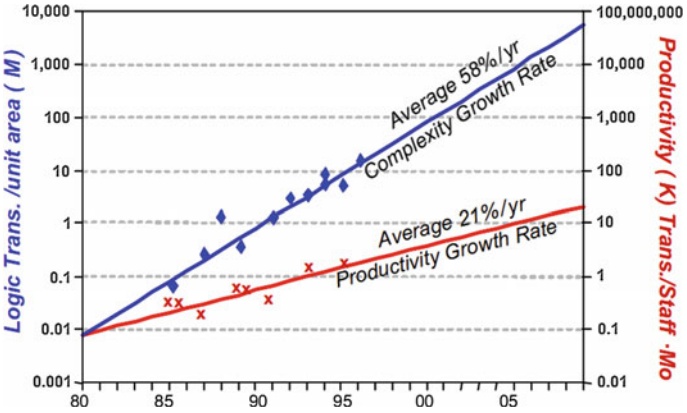


Fig. 1 Production gap between manufacture capacity and design capacity

It can save design cost, shorten design cycle, reduce market risk. Nowadays, it is a prevalent method in chip design.

Hardware device is the fundamental equipment in internet of things. So, the security of hardware integrated circuit should be guaranteed as well as software in internet of things. Nowadays, it is easy to reuse IP cores and manufacture various electronic products. The reused IPs may be misappropriated and utilized unauthorizedly for illegal profits. It directly leads to frequent occurrence of IP disputes every year [2]. Statistical data shows that financial loss caused by IP disputes achieves $50 billion every year [3]. Besides, it also brings damage to enterprise reputation and cooperative relationship. So, it is urgent to protect reused IPs from infringement. This subject has attracted many concerns in academia and semiconductor industry.

IP protection techniques can be classified into four categories: tagging, fingerprinting, watermarking and hardware encryption [4].

(1) Tagging. This technique places an electronical "label" into a chip for reliable and traceable identification. Marsh et al. [5] presented a tagging technique to protect Application Specific Integrated Circuit (ASIC) IP cores. A secure "label" identifying copyright information is placed into a chip. An "external receiver" is required to detect this label. But this method can only deter adversaries due to independence of the label. Besides, it might be damaged or removed. Another technique is physically unclonable function (PUF). It utilizes unique physical characteristics in IC manufacture to generate a radio frequency identification (RFID) "label", which is integrated into a chip to avoid cloning. The security is greatly enhanced, but expensive design cost and working environment of RFID have hindered its development [6].

(2) Fingerprinting. It makes different users get IPs with different identities. The uniqueness of IP fingerprinting realizes clear division of responsibility in IP disputes. But it will generate many IPs with the same functionality and technical index, but with different implementation. Lach et al. proposed an IP fingerprinting technique [7]. It divides a design into a set of parts that have the same characteristics. Each

part has several different implementations. IP module for embedding fingerprint is generated by combining different implementations of these parts. But this technique can only be realized at specific design level of very large scale integration (VLSI). Its application is limited due to the low resistance against collusion attacks.

(3) Watermarking. As a widely-used technique, watermarking is firstly applied in multimedia for copyright protection. In the field of VLSI, watermark is permanently stored in design as an invisible code for IP protection. Guneysu et al. [8] presented standard, protocol and design idea of reconfigurable digital watermark. Li et al. [9] concretely introduced development of IP watermarking and classified it into four categories at physical level, structural level, behavioral level and systematic level.

(4) Hardware encryption. Roy et al. [10] proposed an ending piracy of integrated circuits. A secret key is hidden into circuit. A chip cannot pass the test procedure and enter market if not activated. Besides, the authors also presented a bus based locking and unlocking scheme to protect hardware IPs. Although this technique increases hardware overhead (pins, area, etc.), it has good hiddenness and high security. But the protection is effective only in chip manufacture and test. The traceability after chip product being sold is not involved.

IP watermarking technique is a burgeoning interdisciplinary subject. It involves theories in various field, including microelectronics, signal processing, coding theory, cryptography, etc. So, it is of great significance and economical value to develop IP watermarking techniques. We have proposed four watermarking schemes to protect IP designs.

2 FPGA-Based IP Watermarking Technique

Field programmable gate array (FPGA) IP generally involves four design level, respectively physical design level, structural design level, behavioral design level and systematic design level. Many IP watermarking techniques are realized at these design level, but IP watermarking techniques at physical design level are the most. Kahng et al. presented to map a watermark into a set of constraints and embedded the watermark using satisfiability (SAT)—a classical NP-complete constraint-satisfaction problem. Yip et al. [11] authenticated a FPGA IP watermark by using public key. Nie et al. [12] proposed a post-layout IP watermarking scheme. The post-layout is abstracted into a graph using graph theory and topology theory. Searching algorithm and optimization algorithm are used in watermark embedding. Khan et al. [13] embedded watermark by rewiring circuit with one or more redundancy addition/removal steps. The watermarked circuit has the same functionality with that of the original after removing these redundancies. If constraints such as timing are satisfied, watermarked circuit could take the place of the original one. But, adding a redundant connection may cause some new redundancies. In order to solve security problem of FPGA based IP design, Wei Liang's team [14–16] proposed several effective and robust methods in watermark embedding and detection. Xu et al. [17] mapped a watermark into positions and some watermark bits (0 and 1). These bits

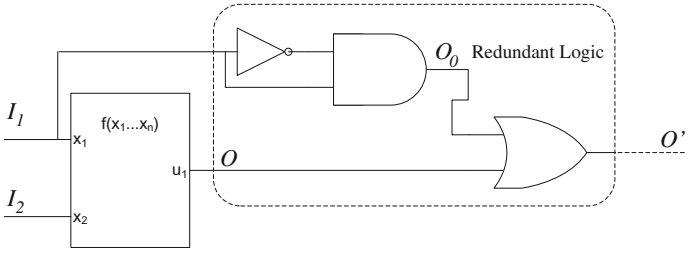


Fig. 2 An example of redundant watermark embedding

are embedded into design in form of redundant logic circuits, as shown in Fig. 2. It causes much less resource overhead. This method can insert more watermark than existing methods due to watermark compression.

At behavioral design level, Raj et al. proposed a watermarking technique for IP identification based on testing in SOC design [18]. It provides high watermark coverage rate, but resistance against collusion attack and hardware overhead require further improvements. Castillo et al. [19] proposed HDL based IP protection scheme by watermarking lookup table (LUT) structure of FPGA. A watermark is inserted between unused LUT and used LUT. However, watermark detection requires adding extra logic. With specific input sequences, this logic will output the watermark data. By comparing to scheme of Raj et al. [18], this scheme is more convenient in watermark extraction. But the added logic is vulnerable to be attacked or removed.

2.1 Secret Key Generation

Most FPGA based IP watermarking utilizes lookup tables (LUTs) structure. Generally, a secret key is required to determine watermark positions. As the key is sensitive information in watermark embedding and extraction, it should be safely reserved. Generally, the generation of secret key requires considering dispersity of positions. It is proper to make the watermarks distribute evenly in the design with high robustness. So, the secret key generation is divided into three steps: resource searching, resource recording and key generation.

Resource searching. An FPGA device always includes configurable logic blocks (CLBs). A CLB includes four slices and there are two LUTs in a slice (e.g. Virtex II FPGA). Firstly, it is necessary to determine the number of unused LUTs in FPGA design. All configurable logic blocks (CLBs) are read in this procedure. Each LUT in CLB is traversed by "Z" shape until all of them are accessed.

Resource recording. During searching procedure, utilization of LUTs in FPGA device is recorded with a two-dimensional table. "0" or "1" respectively denotes a LUT is unused or used.

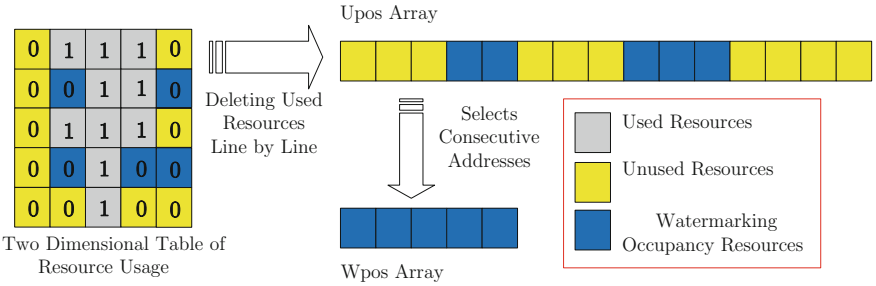


Fig. 3 Secret key generation

Secret key generation. As shown in Fig. 3, recorded utilization of LUTs is recombined as a linear list *Upos*. A block of continuous addresses is selected by random number generator. The linear list stores data of unused LUTs. So, it is highly possible to select continuous addresses that are close to original design. The selected addresses *Wpos* are stored in a key file.

2.2 Watermark Embedding

For FPGA based IPs, the watermarks can be inserted into IP design manually. Namely, some proper positions are searched in physical layout through the design tool (e.g. Xilinx ISE). The watermarks are embedded by configuring the function in caption of the selected positions. Another way utilizes programmable interface provided by device manufacturer. The resource researcher and watermark embedder can be programmed to implement watermarks in bitfile automatically.

A functional soft IP core is described by VHDL language. The design tool (e.g. Xilinx ISE) allocates resources for the IP core. After that, the third-part synthesis or simulation tools (e.g. Synplify and ModelSim) are utilized to map the IP and simulate its functionality. Finally, the physical layout is generated. Constraints in this design, such as timing and area should be set to optimize the design. The watermark positions of LUTs are easily located with the secret key. The watermarks are then inserted into these LUTs by configuring specific function. Besides, some redundant connections are added to hide real watermark positions. Figure 4 shows procedure of FPGA based IP watermark embedding and illustrates some critical steps.

2.3 IP Watermark Extraction

A watermarked IP design may be misappropriated in semiconductor market or illegally used in some products by adversaries. IP owner can apply for a neutral third

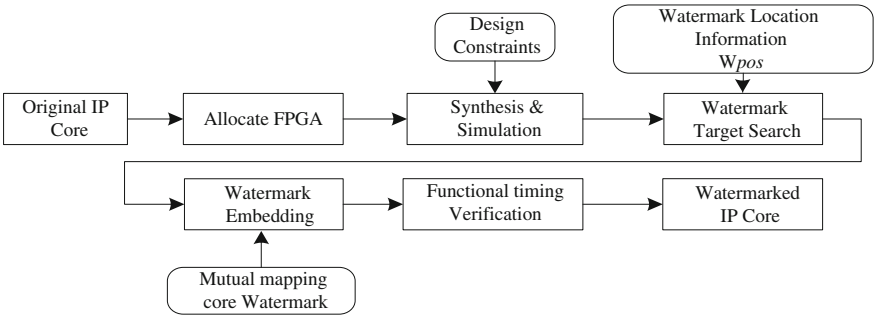


Fig. 4 Procedure of FPGA based IP watermark embedding

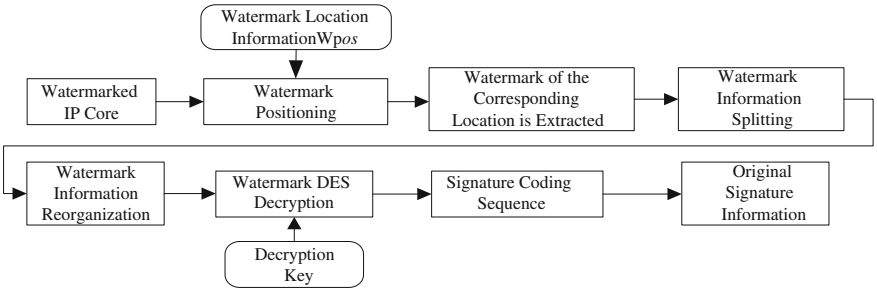


Fig. 5 Procedure of FPGA based IP watermark extraction

party to authenticate the suspected IP ownership. He submits the secret key to the third-party institute and conducts watermark extraction. If a declared watermark is extracted from the suspected IP, the ownership is proven.

Watermark extraction includes watermark locating, splitting, processing, as shown in Fig. 5.

Watermark locating. Generally, IP core is delivered at a low design level (e.g. physical layout) since the use of IPs at physical level is more convenient and easier. So, watermark extraction will locate the watermark positions in *Wpos* and read LUT contents in these positions.

Watermark splitting. The extracted sequence contains encrypted copyright information and mutual mapping factor to reconfigure watermark. With the reserved width, we split the sequence into several parts to reconfigure the original watermark.

Watermark processing. Original watermark is encrypted for better security. With encryption key, the encrypted watermarks in above step can be decrypted. The extracted watermark is compared to the declared watermark for verification. If two watermarks are consistent, the ownership is proven.

Table 1 IP watermarking performance indexes in resource utilization and growth

IP	Device	Length of W	Non-watermarked design			Watermarked design			Growth rate
			L	*L-Num*	$\Delta L(\%)$	*L*	*L-Num*	$\Delta L(\%)$	$\Delta S(\%)$
DES	Xc2v1000	32	3376	10238	32.98	3367	10240	33.04	0.266
STROM	Xc2v1500	64	7308	15357	47.32	7382	15360	47.68	0.272
CACHE	Xc2v2000	128	13234	21521	61.46	13236	21504	61.57	0.305
RS	Xc2v4000	256	25956	46089	56.38	26024	46080	56.44	0.304

2.4 Experiments and Analysis

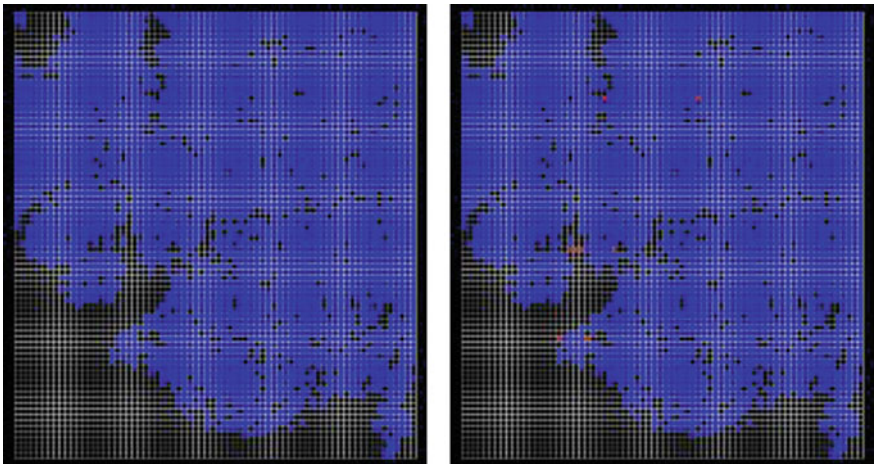
In this section, we will evaluate and analyze the proposed watermarking algorithm in terms of resource overhead and ability against attacks.

2.4.1 Resource Overhead

In watermark embedding procedure, original watermark information is encrypted by DES algorithm and then hashed. The data can be compressed by using Hash algorithm. Consequently, despite the length of original watermark, the hashed result is 128 bit constantly. The resource overhead will not increase when the length of watermark bits is greater than 128.

Table 1 records some performance indexes in resource utilization and growth. *W* denotes embedded watermark. *L* is the total number of utilized LUTs. *L-Num* represents the total number of LUTs in FPGA device. $\Delta L(\%)$ denotes the rate of utilized LUTs and $\Delta S(\%)$ is the growth rate of utilized resource after watermark insertion. The growth rate of utilized resource is constantly close to 0.3% after embedding watermark, which satisfies the requirements of resource overhead. Since the proposed algorithm utilizes unused LUTs for watermark insertion, the watermark will cause resource overhead. However, the watermarked resources will not be accessed when the system is running. Therefore, the power overhead will not increase. The experiments show that the proposed algorithm has good performance on resource overhead and power consumption.

To evaluate the features of low overhead and high watermark volume, we analyze the resource distribution in original design and watermarked design. Xilinx Virtex II XC2V2000 FPGA device is used in experiments. The RS IP core is selected as the target IP design. Figure 6 shows the resource distribution. The proposed model can improve the number of embedded watermark bits. The rate of resource utilization can be also calculated. Meanwhile, we analyze the resource variation and the resource aggregation is better.



(a) Resource distribution of original RS IP design

(b) Resource distribution of watermarked RS IP design

Fig. 6 Resource distribution of RS IP designs

2.4.2 Security Analysis

The security of IP core mainly reflects the ability of watermark withstanding malicious tampering or attacks. The normal attack methods include removal attack, physical attack, forgery attack and collusion attack etc. The removal attack removes the watermark directly by certain means. For the brute force attack, it searches the inserted secret information by force. The forgery attack inserts the illegal watermark to IP core which should not exist originally. The passive aggression represents that the attacker who can detect the watermark and recognize every mark, but fails to decipher the mark code. The security and performance analysis of proposed algorithm in this paper is conducted under the illegal removal attack and noise attack modes.

Ability against Reverse Analysis Attacks. In the proposed scheme, the watermarks insertion can be implemented by configuration of logic function. It is difficult for illegal attackers to get logic function in programmable logic circuit by reverse analysis attacks. To perform reverse analysis attacks, they should firstly obtain all configuration data of FPGA design. There are two ways to get configuration data generally. One is to steal the bit stream and another one is to read configuration data in RAM by using micro-probe.

With the way of stealing bit stream, attackers need to import the programmable data in every time of system booting. The way makes it possible to analyze circuit function from bit stream. In our proposed scheme, a stabilized power is used to keep the information in storage nonvolatile. The configuration data is no need to be imported again in system booting. In this case, the attackers cannot steal the bit stream of IP circuit.

Besides the way of stealing bit stream, attackers may use micro-probe to read configuration information in RAM. Therefore, the RAM units and the output signal in our scheme are set at the low level of chip. The attackers cannot probe related configuration logic by micro-probe. Consequently, IP circuits with our proposed watermark scheme has good ability against reverse analysis through stealing bit stream, especially reverse analysis on layout.

The noises in above experiments are Gaussian noise. In following experiments, we focus on noise attacks of GGD type and MSS type. The noise intensity is denoted by P, $0<P<1$. Figure 7c compares the proposed scheme with the method based on one dimensional chaotic mapping (ODCM). The experimental results in Fig. 6c show that the performance of ODCM against GGD noise attack is low with the increase of P. The reason is that the position aggregation parameter becomes small after suffering GGD noise attacks when P increases. In this case, the error probability of IP circuit increases correspondingly. In Fig. 7d, when P becomes larger, our scheme has better ability against MSS noise attack by comparing with that in literature [20].

Noise Attacks. The signals of the watermarked circuits with our scheme are not in Gaussial distribution. Where ξ denotes the optimal threshold for attack of noises. Using optimization methods in [21] when $\xi = 0.2, 0.4, 0.6, 0.8$, we compare the

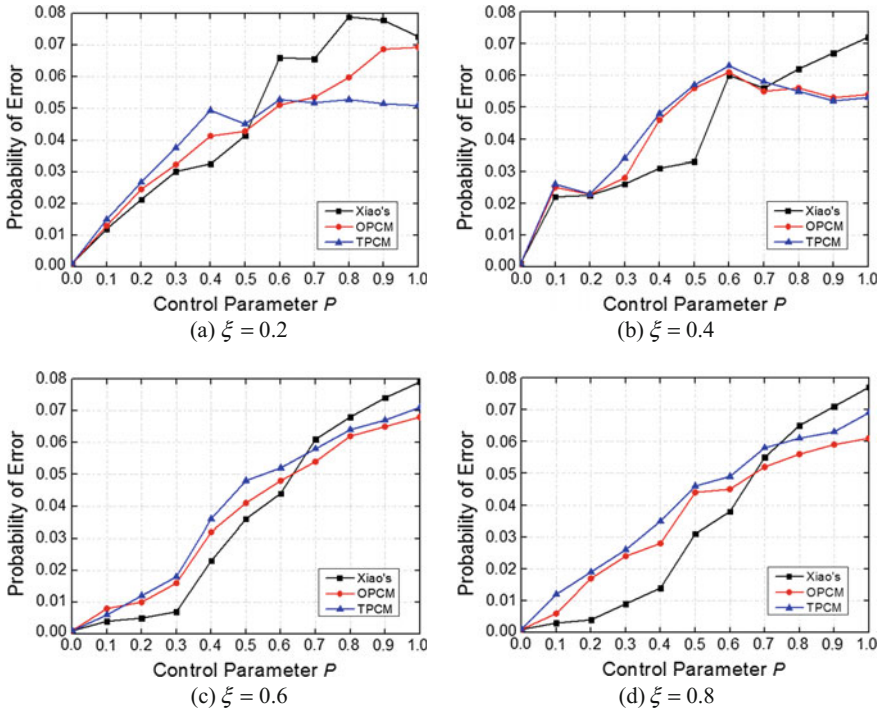


Fig. 7 When $\xi = 0.2, 0.4, 0.6, 0.8$, the performance of various algorithms in terms of resistance to noise attack

performance of resistance against noise attack. The performance after suffering noise attacks can be obtained by using numerical method. Figure 7a shows a comparison of OPCM [14] and TDCM scheme with that in literature [20]. With $P < 0.6$ and low noise intensity, the security of two schemes are better than that in [20]. Figure 7b shows the ability against noise attacks of our proposed two schemes are better than the method based on one dimensional chaotic mapping when $P > 0.9$. In contrast, the proposed method is superior to previously proposed approaches in terms of resistance against noise attack.

3 FSM-Based IP Watermarking Technique

Finite state machine (FSM) based IP watermarking has also been widely studied. Torunoglu et al. [22] utilized unused transitions in state transition graph (STG) for watermarking. As shown in Fig. 8, some new transitions are added in original STG. The watermark is indicated by creating a Euler path. Oliveira et al. [23] divided a 128-bit watermark into a set of bit fragments, as input sequence. A designer modifies state in STG to insert watermark. To enhance the detectability of FSM-based IP watermark, Abdel-Hamid et al. [24] added watermarks into FSM of sequential circuit. This scheme generates various transition adding solutions under control of different key. With initial state and input sequence of watermark, it is convenient to detect watermark from output sequence. Cui et al. [25] proposed an adaptive watermarking technique by modeling some closed cones in an originally optimized logic network (master design) for technology mapping. IP watermark in this scheme achieves low overhead and good resistance against attacks.

For complex logic circuit, STG is implemented by modifying some components of circuit. Different with traditional modification of STG, addition of delay state will not affect output value. If a watermark is implemented in this way, watermark removal will be complex and time-consuming. It makes an alteration to state coding. When value of state variable is not a watermark, new value of state variable will be changed accordingly. By adding two transcoders, newly generated state variable will

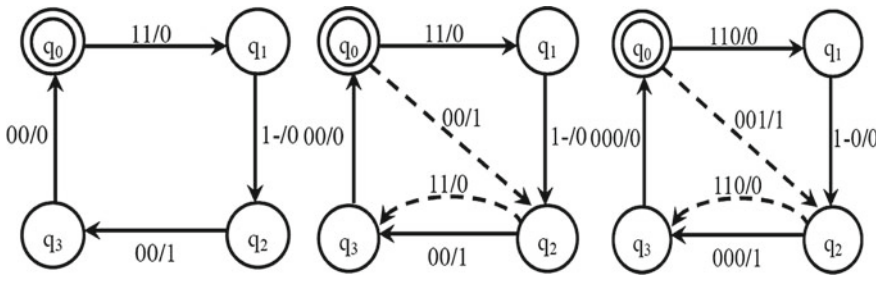


Fig. 8 An example of FSM based IP watermarking

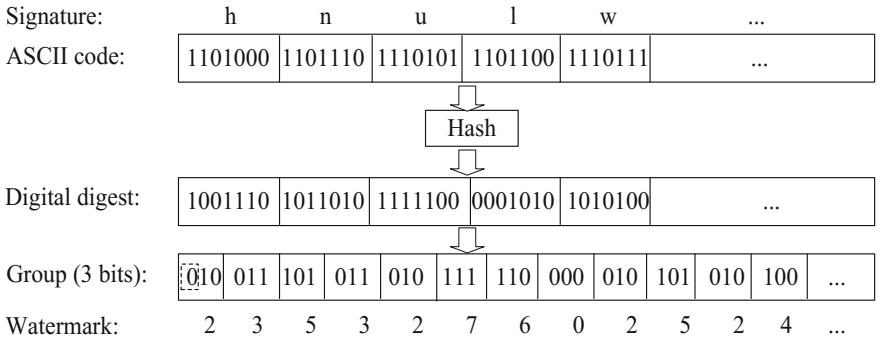


Fig. 9 Watermark generation

change the output. Besides, its value changes as well for any input except $a_1, ..., a_m$. The transcoder is realized by a series of linear transformation. A variable set $X = \{x^1, ..., x^i, ..., x^j, ..., x^n\}$ is transformed as $X\prime = \{x^1, ..., x^i, ..., x^i \oplus x^j, ..., x^n\}$. Any function $F : X \rightarrow \{0, 1\}$ is mapped to $F\prime : X \rightarrow \{0, 1\}^k$. A series of elementary transformations finally realizes any linear transformation by adding two EXOR gates and a gate in transcoder.

Consequently, a FSM-based watermarking scheme is proposed to protect reused IP. When IP dispute occurs, IP owner extracts the maximal delay state set through state transformation relations among circuit signals. Finally, it proves IP ownership.

3.1 Watermark Generation

Since only binary signals can be traced in IP design, a signature should be transformed into a binary sequence. The generated sequence will be then disordered by using a hash function. The digest is divided by three bits (left zero padding) and each group denotes a decimal number (0–7). Let a signature be "hnulw…" and watermark generation process is shown in Fig. 9. "0" in dotted rectangle is left zero padding.

3.2 Watermark Embedding

In this section, we introduce watermark insertion by modifying state delay information in STG, as described in follow steps.

Input: a watermark W and an IP core

Output: watermarked IP core

Step 1: Traverse each state $s_i \in S$ of STG with a sequence of inputs $a_1, ..., a_m$ and collect a set of state transitions R (T).

Step2: Analyze all the state delay information in delay states $R(T)$ and set an appropriate threshold T_N as criteria for selecting $R'(T)$. $R'(T)$ includes states suitable for modification. Selection of T_N depends on the type of an IP core.

Step3: Randomly select γ state delay values from $R(T)$ by considering length of a watermark and create a set of delay states $R'(T)$ for watermark insertion.

Step4: Analyze delay state values in $R'(T)$ at specific positions. Replace the last number of each delay state value with a watermark fragment. This operation is repeated until all watermark fragments are inserted. In this case, a watermarked IP design is generated finally.

3.3 Watermark Extraction

When an IP dispute occurs, IP owner can apply to authenticate the ownership by extracting watermarks from the suspected IP. The watermark is embedded into STG of IP design. The concrete extraction procedure is illustrated as follows.

Input: a watermarked IP core

Output: digest of watermark W

Step1: Extract and analyze STG of the watermarked IP core.

Step2: Traverse each state $s_i \in S$ in STG with a sequence of inputs $a_1, \ldots, a_m$ and obtain a set of state transitions, denoted by $R(T)$.

Step3: Obtain a set of delay states $R(T)\prime$ and analyze the watermarked STG.

Step4: Extract γ watermarked state delay information with random selection rule used in watermark embedding. The last number can be extracted by analyzing state delay information and transformed into binary sequence

Step5: Recombine the binary sequence through the reverse procedure of embedding and finally get the embedded watermark digest.

Verification is implemented by comparing the extracted digest to the declared one.

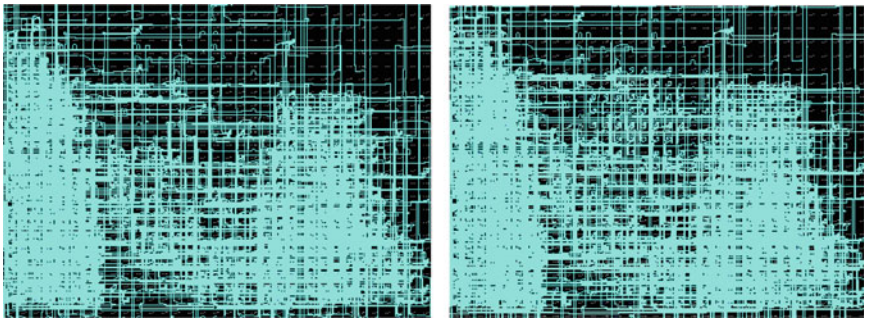
3.4 Experimental Results

The proposed method has been tested on Xilinx VirtexII device XCV600 by watermarking three public cores with 128 bits watermark: DES56, ALU, RSA. The performances in the form of timing, SNR and resources are primarily verified. The test results are shown in Table 2.

Table 2 reveals that DES core utilizes the most CLBs, while ALU the least for the three cores. The core with the maximal delay is DES occupied the most resources, followed by RSA, ALU. By comparison with methods in [26, 27], the proposed method is not the best in terms of timing performance. While the SNR and the occupied resource relative to original circuit are both lower. Therefore, the proposed method has lower impact on circuit function, better security and resource overhead.

Table 2 Performance comparison of different IP core physical layouts

Method	Core	Devices	Used slices	Timing(ns)	SNR	% Resources
[26]	DES	XCV600	972	7.706	0.432	0.786
	RSA	XCV600	668	9.103	0.503	1.899
	ALU	XCV600	481	15.122	0.422	2.591
[27]	DES	XCV600	958	8.416	0.716	0.558
	RSA	XCV600	683	9.706	0.706	2.793
	ALU	XCV600	485	16.231	0.231	2.883
Our method	DES	XCV600	947	7.802	0.602	0.367
	RSA	XCV600	656	9.901	0.491	1.707
	ALU	XCV600	479	17.998	0.368	2.165



(a) Original DES design layout (b) DES design layout with 128 bitwatermark

Fig. 10 Original DES design layout and the layout with 128 bitwatermark

Figure 10 shows the experimental results for DES core. The physical layouts reveal that, the watermarked layout in Fig. 10(b) has higher density of occupied resource, but lower impact on circuit function in comparison with the original in Fig. 10(a).

4 DFT-Based IP Watermarking Technique

Digital watermarking applied in design-for test (DFT) has been extensively concerned. Most of the DFT watermarking techniques focus on scan chains. In the methods proposed by Fan et al. [28], the watermark generation is integrated in the test module. Five possible methods for watermark hiding are presented. Since the test circuit instead of the IP core is marked independently, it is vulnerable to removal attacks. Saha et al. [29] proposed to watermark both the scan tree and single scan chain, separately embedding the signatures of the owner of physical design tool and that of the logic design tool. Cui et al. [30] proposed to insert watermark through

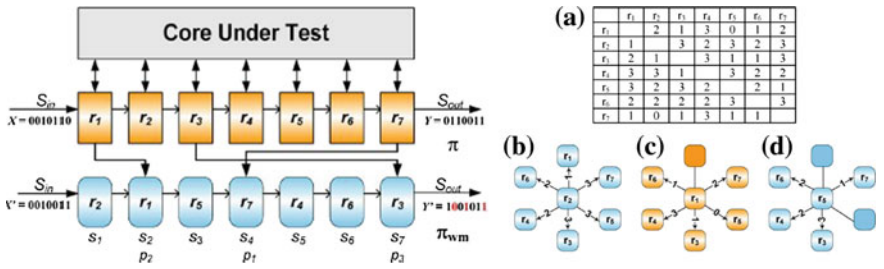


Fig. 11 IP watermark implementation by reordering scan cells

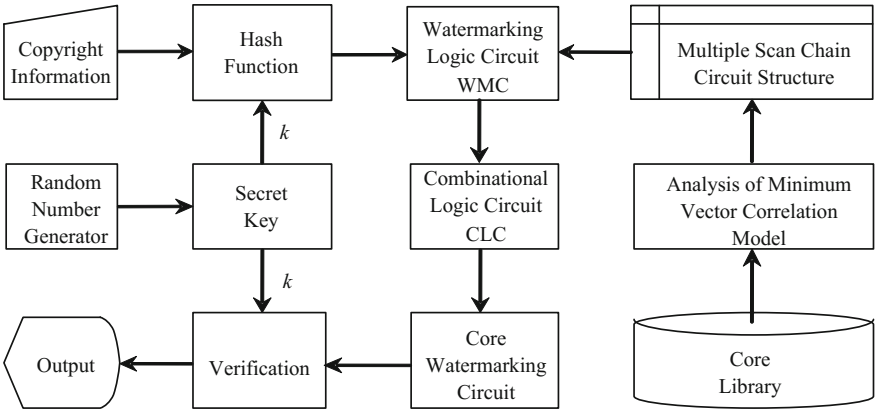


Fig. 12 Overview of multiple scan chains based IP watermarking scheme

reordering the scan cells in a single scan chain minimizing power overhead, as shown in Fig. 11.

In this section, we introduce an IP protection method by watermarking multiple scan chains in sequential circuit. This scheme adopts DFT test model in SOC design, and uses an LFSR for pseudo random test vector generation. Let the structure of multiple scan chains be M. The multiple scan chains M can be transformed into M_p with the minimum correlation $\Phi\left(M_p\right)$ after exchange operations. M_p is suitable for watermark embedding.

The overview of multiple scan chains based IP watermarking scheme in sequential circuit is described in Fig. 12. The copyright information is encrypted and transformed using hash function with private key k. On basis of the minimum correlation model and multiple scan chains, a watermarking logic circuit (WMC) is designed to change states of specific registers in multiple scan chains for watermarking the design. The watermark can be effectively detected without interference with normal function of the circuit, even after the chip is packaged.

4.1 Watermarking Architecture

A watermarking structure of multiple scan chains with minimum correlation is introduced in this section. Figure 13 shows an example for watermarking multiple scan chains. Assume that, the circuit under test consists of 6 scan cells s_i, $i = 1, 2, \ldots, 6$, these cells are organized into two scan chains $c_1 = \{s_1, s_2, s_3\}$ and $c_2 = \{s_4, s_5, s_6\}$. In the watermark circuit, one input of XOR gate is connected to one cell in multiple scan chains, and another controlled by watermark enable signal *w_en* and output of arbitration logic circuit (ALC). However, output of ALC is under the control of states in LFSR.

4.2 Watermark Embedding

The signature, representing one's identity, is encrypted and then hashed. The generated digital digest is inserted into IP core as watermark. Hash function H is a transformation using x as input and the returned value is called hash value, denoted by h, i.e. $h = H(x)$. Since hash is a one-way function, given a value h, it is computationally impossible to calculate x by using $H(x) = h$.

A signature is hashed by MD5 for a 128-bit digest ξ. In preprocessing procedure, ξ is transformed into binary sequence $< \beta_1, \beta_2, ..., \beta_i ... \beta_n >$. The chaos system generates a key sequence $\kappa_s < \kappa_{s1}, \kappa_{s2}, ..., \kappa_{sj}, ..., \kappa_{sp} >$. The sequence $< \beta_1, \beta_2, ..., \beta_i ... \beta_n >$ is mapped to a set of watermark fragments $\{< \varpi_1, \varpi_2, ..., \varpi_j, ..., \varpi_p > | \varpi_j =< \beta_{k,...r} >\}$. So, $\{\gamma(\varpi_j) | 1 < j < p\}$ is utilized to control register positions as a set of constraints. A scan chain with the minimum correlation $< s_1, s_2, ..., s_i, ..., s_\lambda >$ is selected for watermarking. The arbiter logic circuit limits constraints $\gamma(\varpi_j)$ to positions of specific scan chains. Figure 14 shows an example of multiple scan chains based watermarking scheme.

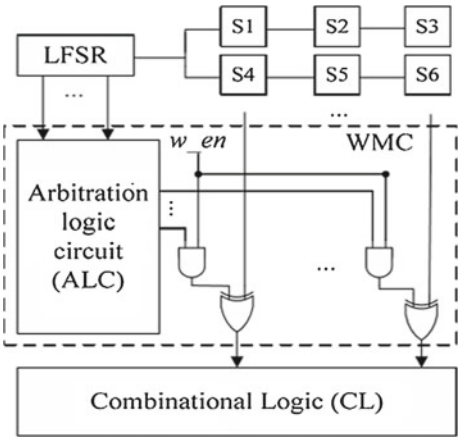


Fig. 13 The watermarking architecture of multiple scan chains

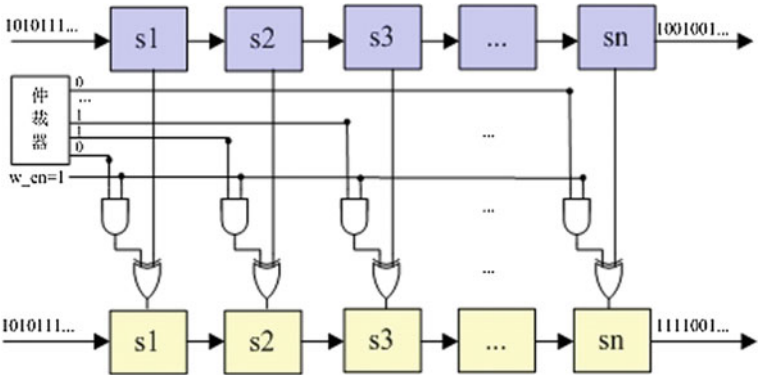


Fig. 14 An example of multiple scan chains based watermarking

There are two modes, normal model and watermark model. In the normal mode (w_en= 0), the circuit under test executes normal scan test and in watermark mode ($w\_en = 1$), a specific state shifted in ALC may cause 1, thus values of some cells in multiple scan chains will be reversed and then be output. The IP identification could be verified by comparing the output in normal mode and watermark mode for the same input vector.

4.3 Watermark Extraction

When the IP core is suspicious to be misappropriation, the author could apply to the third party for the verification of watermark by the following steps.

First of all, we read in the watermarked design and insert architecture of multiple scan chains. LFSR is used for the generation of test vectors. At present, $w\_en = 1$, the watermark circuit is active. The test vectors are shifted in multiple scan chains. The response vectors will be output through the combinational logic in the test circuit. Therefore, the watermarked responses R_m could be detected at the scan output. Then we set w_en signal as '0', now the scan results become the original response R since the watermark circuit is not active. Accordingly, given a specific input vector, by comparing the response vector R and R_m, respectively before and after watermark, the watermark positions will be found. After a series of transformations, the watermark fragments distributed in the whole design are found. Using the stored sequence $Rn(k)$, the watermark fragments can be recombined as an extracted watermark $W_m\prime$. The IP identity could be verified by comparing W_m and $W_m\prime$.

4.4 Experimental Results and Performance Analysis

The proposed scheme by watermarking multiple scan chains with the minimum correlation is implemented in VC on a 1.2 GHz Sun UltraSPARC-T1 machine. The watermarking scheme is applied on sequential circuits from ISCAS'85, ISCAS'89 and ISCAS'99 benchmark suites. The performance analysis of the proposed scheme will focus on resource overhead, resistance to attacks and comparison of experimental results.

4.4.1 Resource Overhead

It is critical to evaluate the resource overhead after watermarking. We select five circuits with the gate number over thousand for experiments. The zero delay models in [31] are used for resource evaluation, through which the transition times will be computed for reflecting the actual resource overhead. The experiment is conducted by the following steps:

(1) Generate the pseudo random vectors by using LFSR and construct the optimal scan architecture with the minimum correlation, and then output the test vectors;

(2) Load the test vectors in the circuit under test and record the transitions of internal nodes, and then calculate peak power and average power;

(3) Partition the test point in sequential circuit according to the architecture of multiple scan chains;

(4) Use LFSR for the generation of test vectors once again, and obtain the watermarked response vectors; calculate the peak power and average power after watermark by recording the transitions of internal nodes during test.

As shown in Table 3, the cells number of combinational circuit and sequential circuit are shown in column 2 and 3 respectively. The columns, "P_w", "P_f" and "ΔK" are respectively the average power, peak power and the coverage rate of the test nodes. The experimental results in Table 3 show: the average power and peak power both reduce accordingly, while the coverage rate increase slightly. It proves that the proposed scheme has the advantages of lower resource overhead and higher coverage rate without affecting the normal circuit function.

4.4.2 Comparison of Experiments

The experiments are conducted on the multiple scan chains with the minimum correlation. The comparison results of the proposed scheme with methods in [32, 33] are shown in Table 4.

Assume that, $\Phi(M_p)$ is the minimum correlation of scan chains, P_c denotes the probability of coincidence and ΔS represents the coverage rate of watermark detection. Table 4 shows that the proposed scheme has lower P_c than other methods, which verify the stronger resistance of our scheme to attacks. The coverage rate of water-

Table 3 The performance comparison of the original and watermarked circuit

Circuit	Combinational Logic N	Sequential Logic C	Original Circuit			Watermarked Circuit		
			P_w	P_f	ΔK(%)	P_w	P_f	ΔK(%)
S5378	2779	179	3797	1968	84.56	3461	1876	90.11
S9234	5597	211	6785	3622	90.12	6123	3601	98.54
S13207	7952	669	10908	6471	82.16	9875	5947	88.34
S35932	16066	1728	41235	19677	89.64	32471	17983	91.44
S38584	19354	1546	20657	14906	92.82	18195	12876	96.01

Table 4 Comparison of watermarking methods

Circuit	$\Phi(M_p)$	Proposed		[33]		[32]	
		P_c	$\Delta S(\%)$	P_c	$\Delta S(\%)$	P_c	$\Delta S(\%)$
i7	18	2.17E-21	91.02	2.91E-21	86.49	7.52E-20	90.23
i9	19	1.02E-14	90.62	3.49E-14	85.07	1.05E-13	88.48
i2	22	1.91E-23	97.83	6.38E-23	83.24	2.64E-19	79.41
i8	15	5.77e-32	94.27	1.67E-32	88.36	2.60E-31	91.86
frg2	14	1.23E-19	92.06	6.02E-18	91.68	1.91E-19	70.77
alu4	25	1.93E-41	94.82	7.14E-34	79.91	1.70E-39	86.09
apex6	20	5.77E-33	99.15	3.06E-24	95.28	8.16E-31	93.62
rot	20	4.98E-26	100.00	8.75E-25	94.76	1.41E-21	87.71
x3	18	4.66E-36	95.37	8.08E-25	89.41	6.28E-35	71.44
k2	33	2.42E-32	96.53	3.24E-32	92.09	8.64E-32	83.57

mark detection ΔS s larger. Due to the architecture of multiple scan chains we use in the scheme, the watermark has become more observable and testable. Therefore, the proposed scheme has lower probability of coincidence P_c and better coverage rate of watermark detection.

5 Self-recoverable Dual IP Watermarking Technique

Robustness is an important metric of IP watermarking technique. However, majority of existing methods cannot recover impaired watermarks after suffered from attacks, causing a failure of ownership authentication. In this section, we introduce an FPGA based dual IP watermarking technique with ability of self-recovery. It authenticates IP ownership even watermark is suffered from illegal attacks and causes lower watermark embedding overhead.

5.1 Watermark Generation

IP circuits has two signals "0" and "1". So, ownership information is firstly transformed into contents that are suitable for circuit. This section generates dual IP watermarks, respectively denoted by binary sequences $s = s_0 s_1 s_2 ... s_n$ and $s\prime = s\prime_0 s\prime_1 s\prime_2 ... s\prime_n$. A watermark indicates ownership information (signature) of IP owner and another watermark represents identity of IP user. In this case, dual IP watermarks can authenticate IP ownership and monitor the use of IPs.

5.2 Watermark Embedding

Generally, the constraints in bitfile should be modified to limit location of watermarked LUTs close to the functional LUTs. It avoids high resource occupation and delay caused by long connections after inserting watermark. The detailed process includes following steps.

(1) Breadth first search and depth first search methods are utilized to locate slices in CLBs. For Virtex II FPGA, there are two LUTs in a slice, F and G. Whether a LUT in a slice is used can be determined by values of F and G in LUT. The values "0" and "1" respectively indicate unused and used. The coordinates of unused LUTs are recorded for selection of watermark positions.

(2) The dual watermarks $s = s_0 s_1 s_2 ... s_n$ and $s = s\prime_0 s\prime_1 s\prime_2 ... s\prime_n$ are divided by 16 bits. Each group relates with a coordinate of LUT. So, an index table δ is created. Here s is the primary watermark and $s\prime$ is the secondary watermark;

(3) An ordered pair (k, m) satisfying $k \leq m$ is selected to create a polynomial $f_1(x) = a_0 + a_1 x + ... + a_{k-1} x_{k-1}$. Here the value of x can be $0, 1, 2, ..., m$ and $2 \leq k \leq m$. $a_0, a_1, a_2, ..., a_{k-1}$ is a sequence of randomly selected coefficients, $a_0 = s$, $H_k = f_1(i_k)$, $i_k \in [0, m]$. In this case, the reconfiguration information of one signature is computed, denoted by $H = \{H_k | k = 1, 2, ..., m\}$. By the same way, we get the reconfiguration information of another signature, denoted by $H\prime = \{H\prime k | k = 1, 2, ..., m\}$. The H and $H\prime$ are reserved as parameters in watermark recovery.

(4) Select four hexadecimal numbers from one signature $s = s_0 s_1 s_2 ... s_n$. The reconfiguration information of both signatures is decomposed into $A \times B + C$. C denotes the information being inserted in location (A, B). The insertion procedure is then performed, namely changing the value at corresponding position in self-constraint file of bitfile. For better security, embedded bits will be encrypted with the private key $key\prime$. The reconfigurable information corresponding to $key\prime$ is $H\prime$. With the same steps, the secondary watermarks can be also processed. Here the embedded bits are encrypted with private key key. The reconfigurable information corresponding to key is H.

(5) Each LUT implements 16 bits' watermark by configuring specific logic functions. Watermark embedding is finished until all watermark bits are inserted in redundant attribute identifiers.

5.3 Watermark Extraction

To authenticate the ownership of an IP, the embedded watermark should be also extracted in this scheme. The extracted watermark will compare to the declared one. If they are consistent, the ownership can be successfully authenticated. But if the extracted watermark has some errors, the task of watermark recovery will be activated. Dual IP watermark extraction includes following steps.

(1) Extract redundant attribute identifiers. If the watermarks are not impaired, we can find all newly inserted redundant attribute identifiers in self-constraint file of bitfile with private keys *key*, *key'*, and the reserved watermark locations.

(2) Reconfigure index table and mapping relation of redundant combinational expression. With position parameter μ of embedded redundant attribute identifiers, the index table and redundant combinational expression $A \times B + C$ can be computed. Thus, we can get the positions of LUTs corresponding to the value of C in index table.

(3) With the inverse process of watermark transformation, we can extract redundant attribute identifier in index table and compute related logic expression. The extracted information is transformed to get hashed value. If it matches that of original signature, the IP is authentic.

5.4 Watermark Recovery

In traditional IP watermarking techniques, watermarks are difficult to recover if being damaged by adversary. The ownership cannot be authenticated with an impaired watermark. To address this issue, a watermark recovery scheme is proposed to authenticate ownership after being suffered from attacks. It depends on the thought of key reconfiguration in secret sharing mechanism. When IP dispute occurs, IP owner can extract and recover impaired watermarks C_1 and C_2. Dual IP watermarks $s = s_0 s_1 s_2 ... s_n$ and $s\prime = s_0 s_1 s_2 ... s_n$ are mutually relevant. There are two cases in watermark recovery. (1) If a part of watermark C_1 is damaged, it can be recovered by $s\prime$, namely $C_1 = E^{-1}\left(F\left(x_2\right), \rho\right). F(x_2)$ is the main mapping function of $s\prime$. ρ denotes self-recovery factor. (2) If another part of watermark C_2 is damaged, s could recover C_2 by calculating $C_2 = E^{-1}\left(F\left(x_1\right), \rho\right). F(x_1)$ is the main mapping function of s.

When IP watermark is impaired after suffering from attacks, watermark recovery can be used to extract correct signature for successful IP authentication. The flow of dual IP watermark recovery is shown in Fig. 15. Relevancy stream P is extracted from watermark stream S' and encoded as $P' = \{f(x_i) | i = 1, 2, ..., k\}$. Finally, P' is utilized as sub-key for reconstructing original signature. Watermark M' can be restored by reconfiguring $f(x)$ and transformed into original watermark finally.

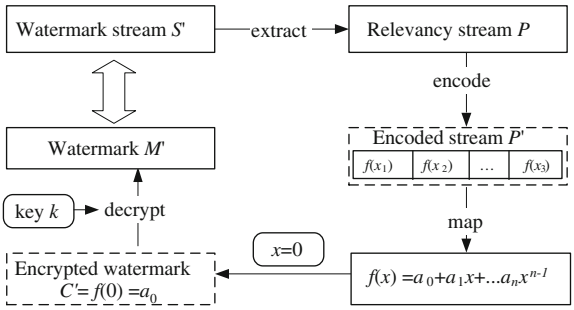


Fig. 15 Flow of dual IP watermark recovery

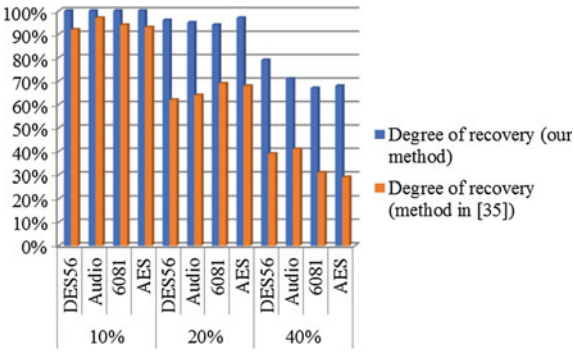


Fig. 16 Evaluation and comparison of watermark recovery

5.5 *Performance Evaluation*

We conduct experiment to evaluate the resistance against removal attack. The length of embedded watermarks is 512 bits. The results with impaired watermarks of 10, 20 and 40% are compared to method in [34]. The comparison is shown in Fig. 16.

After suffering from removal attack, successful recovery of 70% watermarks is regarded as criterion of acceptability. In Fig. 16, with the increase of impaired watermarks, watermark recovery leads to increase of resource and path delay. But if there are 20% impaired watermarks, method in [34] cannot achieve the recovery criterion. The more embedded watermarks are, the more occupation of LUTs is. If impaired watermarks reach 40%, the proposed method has a high percentage to recover impaired watermarks. But method in [34] cannot realize watermark recovery in this case. So, the resistance against removal attack is encourage in the proposed method.

6 Conclusion

IC chip is the basic equipment in IOT environment. IP reuse technique brings convenience, but also cause risk of copyright being infringed. Many watermarking schemes are proposed to address IP protection problems. Reasonable IP watermark embedding and extraction scheme provide protection at various design levels of IP designs. This chapter introduces several types of IP watermarking techniques. It is focused on the intellectual protection problem of the very large integration circuit and a novel algorithm which is suitable for the IP protection of integration circuit has been proposed. These techniques realize improvements on previous work and have great significance to protect reused IPs in IC designs. They succeeded in reducing the power consuming as well as largely increasing the watermark information concealment of the safety modal. Thus, it indeed improved the resistance ability of the watermark algorithm against the illegal attacks

Although the intellectual property core watermark technique has provided many effective watermark algorithms for the research area of integration circuit secure design in recent years, these achievements are still far away from maturation in industrial application. Thus, more research and exploration is still required to find the solution which has a high recognition by both academic and industrial fields.

Acknowledgements This work is supported by the National Science Foundation of China (61572188), the Research Project supported by Xiamen University of Technology (YKJ15019R, YSK15003R), Xiamen Science and Technology Foundation (3502Z20173035).

References

1. Koushanfar F, Fazzari S, McCants C, et al. 2012. Can EDA combat the rise of electronic counterfeiting? In *Proceedings of 2012 49th ACM/EDAC/IEEE design automation conference (DAC)*, 133–138.
2. Majzoobi M, Koushanfar F, Devadas S. 2010. FPGA PUF using Programmable Delay Lines. In *Proceedings of information forensics and security (WIFS)*, 51–65.
3. Guajardo J, Guneysu T, Kumar S S, et al. 2009. Secure IP-block distribution for hardware devices. In *IEEE international workshop on hardware-oriented security and trust*, 82–89.
4. Kirovski D, Potkonjak M. Local watermarks: Methodology and application to behavioral synthesis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 1277–1283.
5. Marsh C, Kean T. 2007. A security tagging scheme for ASIC designs and intellectual property cores. *Design & Reuse*, 57–64.
6. Goren S, Ugurdag H F, Yildiz A ,Ozkurt O. 2010. FPGA design security with time division multiplexed PUFs. In *Proceedings of international conference on high performance computing and simulation (HPCS)*, 608–614.
7. Lach J, Mangione W H, Potkonjak M. 2001. Fingerprinting techniques for field-programmable gate array intellectual property protection. *IEEE transactions on computer-aided design of integrated circuits and systems*, 1253–1261
8. Guneysu T, Moller B, Paar C. 2007. Dynamic intellectual property protection for reconfigurable devices. In *Proceedings of the 15th annual IEEE symposium on FPT*, 287–288

9. Li, D., W. Zheng, and M. Zhang. 2007. Development of IP watermarking techniques. *Journal of Circuit and Systems* 12(4): 84–92.
10. Roy J A, Koushanfar F, Markov I L. 2008. EPIC: Ending piracy of integrated circuits. In *Proceedings of the conference on design, Europe*, 1069–1074.
11. Yip K, Ng T. 2000. Partial-encryption technique for intellectual property protection of FPGA-based products. *IEEE Transactions on Consumer Electronics*, 183–190.
12. Nie T, Liu H, Zhou L. 2012. A time-constrained watermarking technique on FPGA. In *Proceedings of 2012 international conference on industrial control and electronics engineering (ICICEE)*, 795–798.
13. Khan M and Tragoudas S. 2005. Rewiring for watermarking digital circuit netlists. *IEEE transactions on computer-aided design of integrated circuits and systems*, 1132–1137.
14. Liang, W., X. Sun, Z. Xia, and J. Long. 2011. A chaotic IP watermarking in physical layout level based on FPGA. *Radioengineering* 20(1): 118–125.
15. Liang, W., K. Wu, H. Zhou, and Y. Xie. 2015. TDCM: An IP watermarking algorithm based on two-dimensional chaotic mapping. *Computer Science and Information Systems* 12(2): 823–841.
16. Liang W, Long J, Chen X, Xiao W. 2016. Publicly verifiable blind detection for intellectual property watermarks through zero-knowledge protocol. *International Journal of System Assurance Engineering and Management*, 738–981.
17. Xu J B, Long J, Liang W. 2011. A DFA-based distributed IP watermarking method using data compression technique. *Journal of Convergence Information Technology*, 152–160.
18. Raj N, Josprakash, et al. 2011. Behavioral level watermarking techniques for IP identification based on testing in SOC design. In *Proceedings of international conference on information technology and mobile communication*, 485–488.
19. Castillo E, Meyer-Baese U, García A. 2007. IPP@HDL: Efficient intellectual property protection scheme for IP cores. *IEEE Transactions on VLSI Systems*, 578–591.
20. Sun, X., M. Zhang, and H. Zhang. 2013. *Two-Dimension Chaotic-Multivariate Signature System* 10(1). 1694–0814.
21. Basu, A., D.B. Roy, and D. Banerjee. 2011. FPGA implementation of IP protection through visual information hiding. *International Journal of Engineering Science and Technology* 3(5): 4191–4199.
22. Torunoglu I, Charbon E. 2000. Watermarking-based copyright protection of sequential functions. *IEEE Journal of Solid-State Circuits*, 434–440.
23. Oliveira A L. 2001. Techniques for the creation of digital watermarks in sequential circuit designs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 1101–1117.
24. Abdel-Hamid A T, Tahar S. 2008. Fragile IP watermarking techniques. In *Proceedings of NASA/ESA conference on adaptive hardware and systems*. Noordwijk, 513–519.
25. 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*, 1565–1570.
26. Yuan L and Qu G. 2004. Information hiding in finite state machine. In *Information hiding workshop*, 340–354.
27. Abdel-Hamid A T, Tahar S, and Aboulhamid E M. 2006. Finite state machine IP watermarking: A tutorial. In *Proceedings of the first NASA/ESA conference on adaptive hardware and systems (AHS'06)*, 457–464.
28. Fan Y. 2008. Testing-based watermarking techniques for intellectual-property identification in SOC design. *IEEE Transactions on Instrumentation and Measurement*, 467–479.
29. Saha D, Sur-Kolay S. 2010. A unified approach for IP protection across design phases in a packaged chip. In *Proceedings of 23rd international conference on VLSI design*, 105–110.
30. Cui A, Chang C H. 2012. A post-processing scan-chain watermarking scheme for VLSI intellectual property protection. In *Proceedings of 2012 IEEE Asia pacific conference on circuits and systems (APCCAS)*, 412–415.

31. Khan, M., and S. Tragoudas. 2005. Rewiring for watermarking digital circuit netlists. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 24(7): 1132–1137.
32. Cui, A., Chang, C. H. 2008. Intellectual property authentication by watermarking scan chain in design-for-testability flow. In *Proceedings of International Symposium on CAS*, 2645–2648.
33. Kirovski, D., Y.Y. Hwang, et al. 2006. Protecting combinational logic synthesis solutions. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 25(12): 2687–2696.
34. Xu, J., Y. Sheng, W. Liang, L. Peng, and J. Long. 2016. A high polymeric mutual mapping IP watermarking algorithm for FPGA design. *Journal of Computational and Theoretical Nanoscience* 13(1): 186–193.