6

Performance Evaluation for IP Protection Watermarking Techniques

Tingyuan Nie
Qingdao Technological University
China

1. Introduction

The advance of processing technology has led to a rapid increase in IC design complexity. There are now more than thousand million transistors integrated on a chip, and the increasing trend is expected to continue until 2020 or later. This creates the design productivity gap between IC design (typically 20% per year) and IC manufacturing (over 40% per year), and this gap is becoming wider and wider. To close this gap, IP (intellectual property) reuse emerged as the most significant design technology innovation in the past decades. IP companies, third-party libraries, and industry organizations such as the VSIA (Virtual Socket Interface Alliance) have created high expectations for the value and reusability of design IP.

The IP reuse in the reuse-based design methodology is rather different from other reuses such as media, devices to produce artifacts. The reuse of components, designed for a class of applications, is a method to reduce the design-effort, which is well-known from software design for a long time already. In the field of IC design, the reuse of blocks has been practiced in design houses mainly in form of an evolution of existing products. Due to shorter product cycles and rapidly increasing product complexity, many design companies will more and more refer to module cores from outside. During the process of the transfer of design blocks from the original provider to the integrator, intellectual property issues have to be seriously considered. At the same time, some essential issues for IP reuse are outlined: design quality, documentation, security, support, and integration (Thomas et al., 2001). As suggested in the "Reuse Methodology Manual for System-On-A-Chip Designs" (Keating & Bricaud, 1998), an example process of integrating IPs and doing physical chip design can be broken into the following steps:

Selecting IP blocks and preparing them for integration;
Integrating all the IP blocks into the top-level RTL;
Planning the physical design;
Synthesis and initial timing analysis;
Initial physical design and timing analysis, with iteration until timing closure;
Final physical design, timing verification, and power analysis;
Physical verification of the design.

There are many solved or unsolved issues need to be addressed for IP market: friendly interface between IP provider and IP user, design-for-manufacturing, design-for-test,

design-for-reuse, IP standardization, rules for IP exchange, and so on. IP reuse is based on information sharing and integration. Therefore piracy will also have much easier access to the IPs. The IP piracy affects the IP vendors, chip design houses as well as system manufacturers adversely by depriving their revenue and market share. As a result, recent trends of IP piracy have raised serious concerns among the IC design community.

In response to these trends, IP protection becomes crucial to both IP vendors and IP users and becomes one of the key solutions for industrial reuse-based integration. Although sometimes the lack of mechanisms for IP protection becomes barriers to increase design productivity, there have been significant advances from both industry and academic. Especially the VSIA's white paper on IP protection (VSIA, 2000a) and physical tagging standard (VSIA, 2000b) has now been widely adopted by semiconductor and EDA industry. Numerous protection techniques are proposed by researchers both from industry and academia. There exist three forms of IP protection techniques: tagging, fingerprinting, and watermarking. The idea of tagging proposed by Marsh & Kean is to provide a "security tag" for the IP core which can easily be detected off chip using an external receiver called as "wand" (Marsh & Kean, 2007). The approach is vulnerable because the tag can be easily removed by someone if he/her knows some information about the tagging. Bolotnyy & Robins use PUFs (Physically Unclonable Functions) to create aboard RFID (Radio Frequency Identification) tags to protect ICs from cloning (Bolotnyy & Robins, 2007). The security is really improved. However the PUF design is so complicated that the manufacture is hardly reachable. Majzoobi et al. proposed a "Lightweight Secure PUFs" with the new structure in low area, power, and delay overheads. The appeoach facilitates easy security versus implementation cost trade-offs (Majzoobi et al., 2008). There are also other variants in PUF researches, such as implementation of PUFs exploiting physical characteristics other than timing and delay information of silicon circuits. Ravikanth et al. Proposed an optical PUF, which uses the speckle patterns of optical medium for laser light (Ravikanth et al., 2001). Coating PUFs and acoustic PUFs measure the capacitance of a coating layer covering an IC and the acoustic reflections of a token, respectively (Skoric et al., 2005; Tuyls et al., 2005).

Among these techniques, watermarking is the most extensive mechanism implemented at multi-levels of IC design procedure. Primitive watermarking, also known as data hiding, embeds data into digital media for the purpose of identification, annotation, and copyright. The rapid development of digitized media and the internet revolution are creating a pressing need for copyright enforcement schemes to protect copyright ownership. Numerous techniques for data hiding in digital images, videos, audios, texts and other multimedia data have been developed. All these techniques take advantage of the limitation of human visual and auditory systems, and simply embed the signature to the digital data by introducing minute errors. The transparency of the signature relies on human's insensitiveness to these subtle changes. For detail survey, refers to (Gang & Potkonjak, 2003). Especially, watermarking techniques in VLSI domain protects IP cores, CAD tools as well as algorithms from illegal reuse.

CAD tools and algorithms are protected as traditional software by mechanisms such as licensing agreements and encryption. Despite the lack of enforcement of licensing agreements and the security holes of encryption protocols, these protections do not provide the ability to detect IP piracy (Lin et al., 2006). The rare technique that detects possible CAD tool and algorithm piracy is the forensic engineering approach proposed by

Kirovski et al. (Kirovski et al., 2000). It enables the identification of solutions generated by strategically different tools and algorithms. They simply check the given solution for the properties that the algorithm clustering has been performed and claim that the solution is obtained by the algorithm that has the best fit. The poor application to distinguish different algorithms as well as the requirement of candidate algorithms and computing resource is the lack of this technique. So the need for effective CAD tools and algorithms protection becomes vital and urgent. CAD tools and algorithms protection are not in the scope of this book, our work focuses on watermarking techniques for the protection of reuse IP core. We review the representative watermarking techniques and evaluate their performance for both ASIC (Application-Specific Integrated Circuit) and FPGA (Field Programmable Gate Array) designs.

Fingerprinting technology is a complementary to watermarking due to the demand of ensuring the rights of both IP provider and IP users. The main challenge of fingerprinting technique is how to create numerous IP cores with the same function for different IP users. The common approach is to acquire each IP user's signature and repeat embedding it into the entire design to create high-quality solutions from scratch within reasonable amortized design cost.

To the best of our knowledge, the first IP fingerprinting technique is published by Lach et al. (Lach et al., 1998). Their approach is based on the solution by partitioning an initial solution into a large number of parts to provide different fingerprinting realizations (a restricted FPGA mapping problem). Unfortunately, the technique cannot be applied if the design do not have natural geometric structure. Also it has relatively low resilience against collusion attacks due to the identical global structure and the time overhead for creating fingerprinted solutions is relatively high. Andrew et.al proposed a generic fingerprinting methodology that applies to arbitrary incremental optimization/synthesis problems on an watermarked initial "seed" solution to yield different but functionally identical fingerprinted IPs. The approach enhanced collusion resiliency with low runtimes but different solutions are not guaranteed (Andrew et.al, 1999). Gang and Miodrag proposed a fingerprinting technique which uses arbitrary optimization on the problem formulation superimposed additional constraints to produce numbers of distinct solutions with high quality. The run-time overhead for generating many solutions is almost zero (Gang & Miodrag, 2004).

The remainder of this section is organized as follows. We first review the related works of watermarking techniques. Analyze the representative watermarking techniques, introduce watermarking performance evaluation function, and show experimental results for watermarking techniques of ASIC. Followed give a simplified FPGA watermarking investigation and estimation. Finally we have a conclusion for overall work.

2. Watermarking performance evaluation

Referencing viewpoints by VSI Alliance (FallWorldwide Member Meeting, 1997), a state-of-art watermarking-based IPP technique should be:

1. Maintenance of functional correctness.
2. High-credible; coincidence probability, the probability a non-watermarked design might coincide by accident with a watermarked one should be low enough.
3. High-security; watermark should be in the integrity or can be extracted under attack.

4. Low embedding cost.
5. Low overhead.
6. Traceable.

According to the requirements of watermarking, a complete methodology for watermarking performance evaluation should be established. Unfortunately, limited to our knowledge, there is no comprehensive evaluation function for IP watermarking techniques so far. The only literature published for watermarking investigation is accomplished by Abdel-Hamid et al. (Abdel-Hamid et al., 2003). However, they only compared performance of the approaches from their embedding cost, overhead, probability of coincidence, and security. There was no more deeply analysis and evaluation for the watermarking techniques.

In the context, we introduce representative watermarking techniques and evaluate their performance for the two usual IC forms: ASIC and FPGA respectively.

2.1 Watermarking performance evaluation for ASIC

From watermarking construction style, there are almost two methods for watermarking ASIC IP cores. One focuses on introducing additional constraints on certain parts of the solution space of synthesis and optimization algorithms. Another is adding redundancies to the original design.

From VLSI design process, pre-processing watermarking methods and post-processing watermarking methods are discussed. Pre-processing techniques embed watermark before the synthesis tools are applied to solve the watermarked problem. Post-processing techniques firstly solve the original problem without any watermarks. The solved solution will be altered sequentially based on the watermarking constraints. According to design process, watermarking techniques at behavioural-level, structural-level, physical-level, and algorithm-level are proposed.

There may be some shortfalls or defects for a certain watermarking technique. It becomes an important work to evaluate the performance of a watermarking technique because the approaches may bring influences to the origin. So it is impending to build methodologies and functions for watermarking performance evaluation.

2.1.1 Watermarking technique review

In this section, we firstly review a few representative watermarking techniques constructed at different design levels. Then analyze them form a few essential aspects: embedding cost, coincidence probability, security, and tracing cost.

2.1.1.1 Physical-level watermarking

Kahng et al. firstly proposed the constraint-based watermarking methodologies based on the usage of available tools which solves NP-hard problems (Kahng et.al, 1998). The algorithm adds extra constraints to such solutions that would make it yield the new watermarked design. They validated the approaches in pre-processing and post-processing, respectively. The pre-processing flow provides a method that adds constraints by involving segment widths, spaces, and choice of topology. They applied the watermarking by encoding a signature as upper bounds on the wrong-way wiring used to route particular

signal nets. The post-processing flow provides a method that encodes a signature as specified parity of the cell row within which particular standard cells must be placed. Narayan et.al provided a method for embedding a watermark by modifying the number of vias or bends of the nets in a design (Narayan, et.al, 2001). There were 12~13% expense in the number of vias and wire length which is unpractical in real life. The author also proposed a post layout watermarking method which smartly changes route directions by setting obstacle and rerouting (Nie, et.al, 2005). There was no extra wire length overhead and the incremental watermarking time is acceptable. Other techniques at physical design level are also proposed (Min & Zhiqiang, 2004; Irby et.al, 2000).

For physical design watermarking, we choose the most representative technique proposed by (Kahng et.al, 1998) for evaluation instance. According to the published results, the extra routing CPU run time for watermarking is about 9.00%; increased wire-length and via number (watermarking overhead) are 0.58% and 0.55% respectively, sum is 1.13%; the coincidence probability geometrically reduced to the constraint number, from $1.1e^{-8}$ (nearly 10^{-3} for 20 constraints) to less than e^{-85} (nearly 10^{-25} for 320 constraints). From their analysis, the approaches can prevent "ghost signatures" and forging attack due to enough-long constraints and message encoding. They also showed the result from tampering with placement and routing watermark which indicates solution quality degrades much faster than signature strength. It proves that tampering does not appear to be a viable form of an attack.

2.1.1.2 Behavioral-level watermarking

Torunoglu et.al and Oliveira introduced a similar watermarking-based copyright protection technique of sequential functions at behavioral design level (Torunoglu & Charbon,2000; Oliveira, 2001). The algorithm is based on adding new input/output sequences to the finite state machines (FSM) representation of the design. It extracts the unused transitions in a state transition graph (STG) of the behavioral model. These unused transitions are inserted in the STG associated with a new defined input/output sequence, which will act as the watermark. The main advantage of this kind approach is the ability to detect the presence of the watermark at all lower design levels. Torunoglu and Charbon performed exhaustive search only in one case due to the extreme computational complexity of this method. The CPU time in this case was 1.0 second for an area of 2.33-k gates, but it increases exponentially according to their computation formula. The coincidence probability of watermarking is from 10^{-7} to 10^{-34}, averagely 10^{-11}. The watermarking overhead (Extra area of modified FSM) is from 0.2% to 143%, average is 23.77%. It will be much larger if the expected watermark becomes longer. The number of I/O pins which is used to create sequence to insert watermark is not very long, so the approach's resistance to "ghost signatures" attack is not as strong as expected. They proved the "tampering" attack will not successful under various assumptions. Unfortunately, because there is no encryption for the watermarking, the approach is weak to "forging" attack.

2.1.1.3 Structural-level watermarking

There are few watermarking works at structural-level. Kirovski et.al developed a watermarking approach to protect EDA tools and designs at the combinational logic synthesis level. The user-specific watermarking instance is soluted by imposing constraints to the original logic network, where the constraints are uniquely dependent on author's

signature (Kirovski et.al, 1998). Cui and Chip-Hong also proposed the similar approach by resynthesizing the "master design" to meet the application constraints.

We select the first appraoch as representative for structural-level watermarking performance evaluation. From their result, the runtime for the watermarking was controlled within ±5% of the program execution runtime. The average likelihood of watermarked solution coincidence is less than 10^{-13} with the overhead of 4%. Because the adopted watermark constraint length is short (5-inputs), its resistance to "ghost signatures" attack is likely low. They proved that the attacker has to perturb great deal of the obtained solution to tamper the watermark while preserving solution quality, like to develop a new optimization algorithm. For "forging" attack, it is less efficient than trying to tamper the signature in a top-down approach and it is more impossible in a bottom-up approach due to the one-way function encoding.

2.1.1.4 Algorithm-level watermarking

There are rare approaches at the algorithmic level. Chapman & Durrani proposed a Digital Signal Processing (DSP) watermarking scheme (Chapman & Durrani, 2000). The algorithm is based on the ability of designers to make minor changes in the decibel (db) requirements of filters. In this approach, the designer of a high level digital filter encodes one character (7 bits) as his/her hidden watermark data. Then the high level filter design is divided into 7 partitions where each partition is used as a modulation signal of one of the bits.

The authors did not discuss the strength of their approach or the probability *Pc* that the design might coincide with a non-watermarked design. The approach as well depends on a very low data rate, just one character (7 bits), which makes it really unpractical to be used in an industrial environment. The approach is also missing a clear way to track and extract the watermark at lower levels. Therefore, we think the approach is incipient and do not evaluate its performance.

2.1.2 Watermarking performance evaluation function

As described in the context, we consider performance evaluation of watermarking techniques from five aspects: embedding cost, coincidence probability, overhead, security, and trace cost. The components of watermarking performance are illustrated in Fig.1. We formulate watermarking technique performace P using the following function:

$$P = F(Em\_Cost, Coin\_pro, Overhead, Security, Trace\_\cos t) \quad (1)$$

Where P is a function with six variables: Em_Cost, Coin_Pro, Overhead, Security and Trace_cost. Em_Cost represents watermarking embedding cost which usually means the additional wire length or vias for watermarking represetation. Overhead represents how long EDA tools runt for watermarking process. Coin_Pro represents the probability that the watermarked design coincided with a non-watermarked one. Security represents strength of watermarking technique resists to various attacks. Trace_Cost displays the cost retrieving watermark from a protected IP design that can be considered almost the same to the embedding cost. Maintenance of functional correctness is not considered as a factor of the function because each watermarking technique in the market should at least satisfy this requirement.

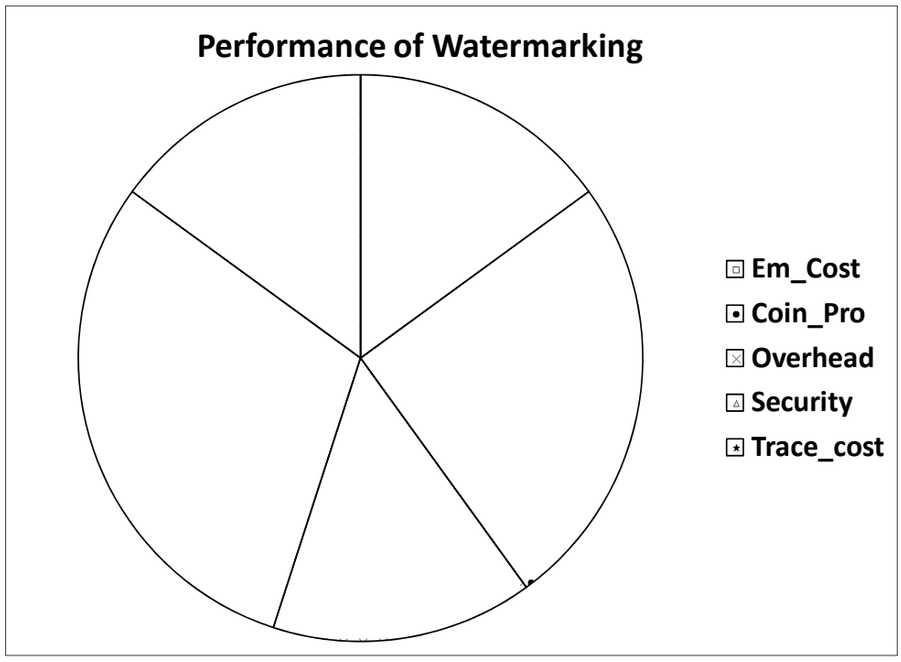


Fig. 1. Watermarking Performance Components

Obviously, lower watermarking cost leads to high-performance watermarking technique. Therefore watermarking performance is reverse to embedding cost. Similarly, watermarking performance is reverse to coincidence probability, overhead, and tracing cost. Instead watermarking performance is proportional to its security which must be concerned. We give a function f_i and a weight to each component, equation (1) can be formulated as:

$$P = \alpha \bullet f_1(Em\_Cost) + \beta \bullet f_2(Coin\_pro) + \gamma \bullet f_3(Overhead) + \lambda \bullet f_4(Security) + \mu \bullet f_5(Trace\_Cost) \quad (2)$$

In practice, watermarking tracing cost is almost equal to watermarking embedding cost, so formula (2) can be simplified as:

$$P = 2\alpha \bullet f_1(Em\_Cost) + \beta \bullet f_2(Coin\_pro) + \gamma \bullet f_3(Overhead) + \lambda \bullet f_4(Security) \quad (3)$$

Each part of formulation (3) is related to both watermark constraints size and watermarking method. Consider process of watermarking-based IP protection, we evaluate the performance of watermarking techniques in such rules:

The watermarking IP protection process is implemented by either intrusive software or an incremental implementation of EDA tool. So the additional CPU run time of the implementation is considered as the embedding cost. Generally, watermarking identification needs some extra circuits. We take the increased wire-length and (or) via number as watermarking overhead. It is considered that the security of watermarking techniques is related with its resistance to attacks. There is a brief introduction of prototypical attacks referred in (Kahng et.al, 1998). The attacks include "ghost signatures" finding, tampering, and forging. To find "ghost signatures", hacker may try a brute-force approach to find a signature that corresponds to a set of constraints that yields a convincing

proof of authorship Pc. However, this brute-force attack becomes computationally infeasible if the threshold for proof of authorship is set sufficiently low. e.g., $Pc \leq 2^{-x}$ (x is the length of constraints). So it is easy to prevent this type attack just by enlarging the length of signature (watermark). As an alternative, attackers may select re-solving every subsequent stage of the watermarking process to forge author's signature. Generally, Specific changes that attacker makes to the final solution will likely correspond to (1) local perturbations of the solution to the watermarked phase, or to (2) global-scale transformations such as those which exploit asymmetry of the design representation. It is critical that common watermarking technique has the resistance to such transformations. Tampering attacks might not be able to ruin the proof of authorship before they significantly degrade the quality of the final solution. Finally, attacker may select to forge author's signature. To finish this work, he needs a signature that he can convince others belong to author. If a signature corresponds simply to a text message, he simply chooses a text message resembling one that author would use. However, such attacks can be easily prevented by using a private key encryption system for watermark generation. We analyze the security of watermarking techniques from the above three aspects, and give a quantitative performance evaluation.

2.1.3 Watermarking analysis summary

Through the investigation and analysis, performance of various watermarking techniques is summarized in Table 1. There are total six columns each display the item of watermarking performance. The first column displays watermarking type. The second and the fifth column are the watermarking embedding cost and tracing cost which represent the increased CPU runtime corresponding to normal IC design process. The forth column of "Overhead" represents the increased percentage of wire length and via number. In the fifth column of "security", there are 3 sub-columns: G, T, and F which represent the resistance to "ghost signatures", "tampering", and "forging" separately. The value of "+" means the method has resistance to such attack, while the value of "-" means no resistance or resistance is really weak.

Watermarking	Em_Cost	Coin_Pro	Overhead	Security			Trace_Cost
				G	T	F	
Physical	9.00%	10^{-3}~10^{-25}	1.13%	+	+	+	9.00%
Behavioral	expensive	avg: 10^{-11}	23.77%	+	+	-	expensive
Structural	5.00%	< 10^{-13}	4.00%	-	+	+	5.00%

Table 1. Performance summary of watermarking techniques

2.1.4 Evaluation results

We evaluate the representative watermarking algorithms from five items: embedding cost, coincidence probability, overhead, security, and tracing cost. According to the investigated results, we calculate each sub-value in the scope (0, 1). Finally we accumulate all the value as the performance evaluation by using formula (3).

Performance of watermarking technique is related with the run time of watermarking process. The more time consumed, the more watermarking technique is ineffective. We define sub-function f_1 as:

$$f_1(Em\_Cost) = 1 - \frac{Em\_\cos t}{Design\_\cos t} \tag{4}$$

Where Design_cost displays the original design cost. If Em_cost (embedding cost) is too expensive to exceed Design_cost, the value of function f1 will be equal to 0.

If the coincidence probability of a watermarking techniques is sufficiently low (for example, less than 10^{-3}), f_2 function can be set as:

$$f_2(Coin\_pro) \equiv 1 \tag{5}$$

The overhead of watermarking (increased wire length, extra via, etc.) degrades the performance of the design. The function f_3 can be written as:

$$f_3(Overhead) = 1 - \frac{Overhead}{Total} \tag{6}$$

Where Total is the total cost for the original design.

There are three factors impact the watermarking security: the resistance to "ghost signature", "tampering", and "forging". We think that no matter which factor is satisfied, the watermarking security gets 1/3 value augment. The f_4 can be written as:

$$f_4(Security) = N \times \frac{1}{3} = \frac{N}{3} \tag{7}$$

Where N is the number of satisfied factors.

Substituting the formulations (4), (5), (6) and (7) into formulation (3) and the of set Design_cost and Total to 1, we have:

$$P = 2\alpha(1 - Em\_\cos t) + \beta + \gamma(1 - Overhead) + \lambda \bullet N / 3 \tag{8}$$

We prepare three schemes to evaluate performance of watermarking techniques: (a) **Balance evaluation** where each item weights are the same, namely α=β=γ=λ=0.2; (b) **Cost emphasis evaluation** where the weights of cost and overhead are set double to others, namely α=γ=0.25 and β=λ=0.125. (c) **Security emphasis evaluation** where security weight is set double to others, namely λ=2/6 and α=β=γ=1/6. (All the weights obey: 2α+β+γ+λ = 1)

Based on formulation (8), we calculated the concrete performance value for the several watermarking techniques. The results are shown in Table 2.

Scheme	Physical WM	Behavioral WM	Structural WM
Balance	0.9214	0.4858	0.9053
Cost_Emphasis	0.9268	0.3989	0.8867
Security_Emphasis	0.9345	0.5159	0.8656

Table 2. Performance of watermarking techniques

The first column show the evaluation schemes mentioned above: Balance evaluation, Cost emphasis evaluation, and Security emphasis evaluation. The second, the third and the forth

column show evaluated performace value of different watermarking techniques in various test schemes. From the result, we understand performace of physical watermarking representative is high, then structural watermarking representative, and behavioral watermarking representative is relatively low, no matter the scheme. From the curves in Fig.2, we can understand the comparison more intuitively.

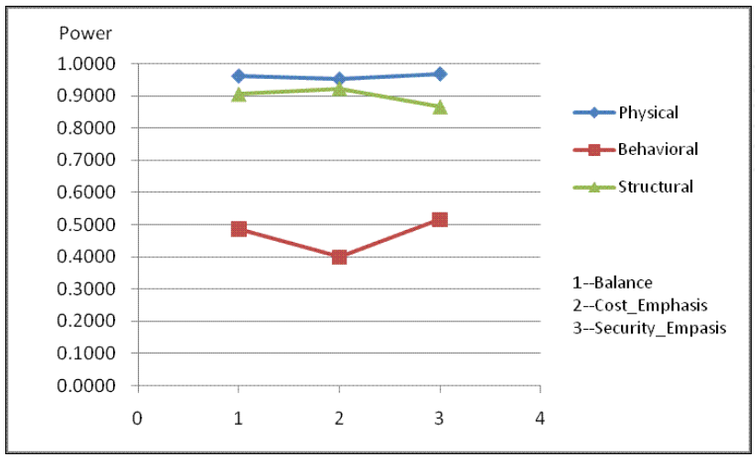


Fig. 2. Performace illustration of watermarking techniques

We introduce functions to evaluate watermarking techniques and hope this work can provide a standard candidate for researchers to evaluate their watermarking techniques. Although performace of various watermarking techniques is different, even the weak technique has its advantages. In future, researchers may develop stronger watermarking techniques by combining the advantages of different level watermarking techniques to prevent any IP piracy attempt from happening.

2.2 Watermarking performance evaluation for FPGA

Before the FPGA being watermarked, a signature should be prepared. The signature may be a short ASCII-text, which identifies the owner of the core. The string is then hashed and encrypted to generate a seed of watermark. Then the watermark is produced from the seed with a pseudo random generator like RC4.

Fig.3 gives an example of FPGA watermarking design flow. As shown in the figure, there are three types of FPGA cores: Source-cores, netlist-cores and bitfile-cores, corresponding to the design levels. Source-cores are delivered in HDL or C language. There are very flexible to synthesize for many target technologies. Netlist-cores have a medium flexibility because they have been fixed on a target technology. Bitfile-cores are very inflexible since they can be used only for a specific device.

Daniel & Jurgen had an accurate evaluation of watermarking methods for FPGA-based IP cores from functional correctness, hardware overhead, transparency, verifiability, difficulty to remove, and proof strength of authorship (Daniel & Jurgen, 2006). They divided watermarking techniques into two categories from their construction: additive methods and constraint based methods. In this chapter, we introduce recent FPGA watermarking techniques and estimate their performance under certain criteria.

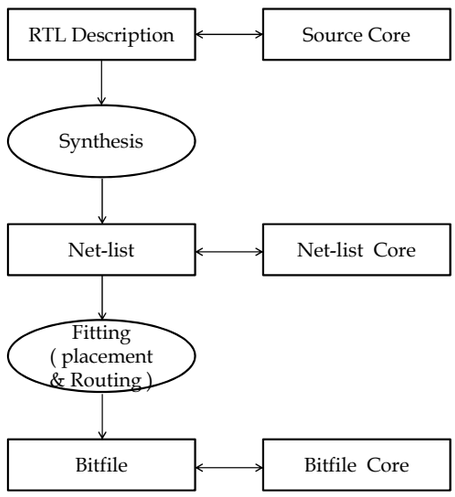


Fig. 3. FPGA watermarking design flow and IP core

2.2.1 Additive methods

Additive methods in FPGA design are watermarking procedures where a signature is added to the functional core. The watermark is not embedded into the functional core yet be masked as a part of the core.

There exist no publications about additive watermarking for source cores protection although it is possible to write an additive source component into the core. However it isn't an applicable watermark strategy because one can also remove this component easily.

Most additive watermarking methods for netlists just watermark the design by introducing redundant logic to the circuit. Moritz et.al presented a novel approach to watermark FPGA designs by converting functional LUTs (Lookup Tables) to LUT-based RAMs or shift registers prevents deletion due to optimization (Moritz et.al, 2009). The resource overhead for watermarking is tiny, generally less than 5%. The method is transparent to EDA tools because the watermarking is performed after the usual netlist generation. The suspected design can be verified only when the extracted bitfile is not encrypted. The authorship can be detected without requesting additional information from the producer. However the watermark can be easily removed by reverse-engineering and the authorship will dispear.

An approach for watermarking bitfile-core is implemeted by embedding the signature into unused look-up tables (John et.al, 1998). The signature will be hashed and coded with an error correction code (ECC) to be able to reconstruct even if some lookup tables are tampered. After the initial placement and routing, the number of unused lookup tables are determined. The ECC code is split into the size of the lookup tables and additional LUTs are added to the design. The watermarked design is obtained after being re-placed and re-routed.

The approach was improved (John et.al, 1999) by using many small watermarks whose size is the exact size of a lookup table. The small watermarks are easier to search relatively. However, the published watermark positions in verification process make the watermarking

technique easily attacked. Futhermore, Lach et.al improved the approach to a fingerprinting technology by encoding the fingerprint into the position of the mark in the tile (Lach et.al, 1998).

The watermark consumes low hardware overhead because the unused lookup tables in the original design would remain empty. The approaches provide a strong proof of authorship and are transparency to EDA tools. The methods are verifiable because it is possible to determine the position of the watermark in a tile. On the contrary, the watermark is alos easy to be remoed or overwrited.

2.2.2 Constraint based methods

The constraint based watermarking methods apply to solutions of hard optimization and constraint-satisfaction design problems. It is centered around the use of constraints to "sign" the output of a given design synthesis or optimization. The solutions of a given optimization instance that satisfy these constraints have a watermark embedded in them and provide a probabilistic proof of authorship. The less likely that randomly chosen solutions are to satisfy these constraints, the stronger the proof of authorship is. The coincidence probability Pc is given by the following formula:

$$P_c = n_w / n \tag{9}$$

where n is the number of solutions which satisfy only the original constraints and n_w is the number of solutions which satisfy both the original and the watermarking constraints. If Pc is very small, the solution provide a strong proof of the watermarking existence. A watermark's resistance to attacks is inversely proportional to an adversary's ability to manipulate it without resolving a given optimization problem from scratch.

Darko & Miodrag proposed an approach for a HDL core protection using a watermarked scan chain (Darko & Miodrag, 1998). At first all registers will be sorted to be assigned a sequential number. A pseudo random sequence is generated from author's signature to select registers according to a certain algorithm. The first K selected registers are chosen for the first register in a chain, where K is the number of used scan chains. The variation of the scan chains for different signature can be used to detect the watermark. Unfortunately, an injudicious chosen of test chain could result in more routing resources overhead. The approach is transparent to the synthesis tools because the signature is added to the HDL core. The watermark can be verified easily only when the scan chains can be accessed from outside of the chip. Some deletion of watermark results in corruption of the scan chain. In additional, a strong proof of authorship can be achieved by using a large number of registers in scan chains.

An approach to protect netlist cores is implementing by preserving certain nets in the synthesis and mapping step (Kirovski et.al, 1998). Some nets are chosen from the sorted nets of design according to a signature. These nets are prevented from elimination by the design tools by connecting to a temporary output of the core. Additional logic is inserted to connect the new outputs together to reduce the amount of the additional outputs. The design with new outputs can be seen as the result of constraint based watermarking. The additional logics for watermarking require some resource overheads. This approach is transparent to EDA tools because the choice of preserved nets for watermarking can be done before the synthesis process. The watermark can be verified by comparing the given netlist with the

original one. However, it is impossible to verify the watermark from a bitfile. The security of this approach is insufficient because the additional logic is easy to remove by re-synthesizing the design. Furthermore, although the probability of coincidence is really low, forging watermarked design is possible which results in weak authorship proof.

An incremental placement and routing or timing constraint is applied to watermark FPGA bitfile-cores.

As an alternative, a watermark can be embedded by placing configurable logic blocks (CLBs) in even or odd rows depending on the constraints (Kahng et.al, 1998). The resource overhead for watermarking is very low and even tends to zero because the placement is altered marginally. The approach is transparent because the watermarking stage is performed before placement implementation. The CLBs can be corresponded to the signature uniquely by enumerating them form the top left corner. Then the watermarked design can be verified with only the given bitfile. It is nearly impossible to remove watermark from the given bitfile because the CLBs are tightly connected with each other. This approach has a strong proof of authorship due to the large amount of CLB position candidates for watermark embedding.

Another proposed method is to add constraints to the router. The constraints make the router route a net with some unusual routing resources like "wrong way" segments, in which the net goes to a wrong direction and then back in the right direction to form a backstrap. The net can be verified as a watermark net due to its special geometry. The routing resource for watermarking is too minor to be neglected. The approach is also transparent to EDA tools because constraints are added before invoking routing. The watermarked design can be verified with the known strategy and the unique nets. It is easy to remove the mark by wrapping up the constraint nets and rerouting it again if someone knows the routing information and the watermarking algorithm. The proof of authorship is not very strong because the watermark is ambiguous and easy to remove or tamper.

A watermarking approach by setting additional timing constraints between registers is proposed in (Kahng et.al, 2001). The timing constraints for the selected paths may split into two separate constraints, each have a new constraint.

Another approach selects the uncritical paths and adds new timing constraints on them (Adarsh et.al, 2003). The last digit of the time delay is reset depending on the watermark. For example, a path has a delay of 10.64ns. If the corresponding watermark bit is '1', the new time delay of this path is set to10.61ns, if the corresponding watermark bit is '0', the delay is set to 10.60ns.

These approaches for watermarking need no resource overhead. They are transparent because additional constraints are added before invoking the routing tool. These approaches are difficult to verify so that it no use to talk about their authorship proof and attack resistance. However designers can create different bitfiles from the same design which are useful for fingerprinting.

2.2.3 FPGA watermarking validation

As mentioned in (Daniel & Jurgen, 2010), when considering a finished FPGA products, there are five potential information sources can be used for extracting a watermark: configuration bitfile, ports, power consumption, electromagnetic (EM) radiation, and temperature.

The bitfile can be extracted by wire tapping the communication between the PROM and the FPGA. Some FPGA manufactures provide an option to encrypt the bitstream which makes communication monitoring useless. However, it is possible to read out some information stored in RAMs or lookup tables to finish verification. Another approach is to employ unused ports which is limited only at top-level designs and impractical for IP cores.

The method called "Power Watermarking" can force patterns on the power consumption of an FPGA as a covert channel to transmit data to the outside. Related works shown in (Ziener & Teich, 2008) and (Ziener et.al, 2010) indicate the clock frequency and toggling logic can be used to control such a power spectrum covert channel. The resulting change in power consumption can be extracted as the signature from the FPGA's power spectrum.

With almost the same strategy it is also possible to extract signatures by raster scanning electromagnetic (EM) radiation of an FPGA with an EM sensor (Thomas & Christof, 2003). Unfortunately, it becomes unpractical since modern FPGAs are delivered in a packaged shape which decreases the EM radiation.

Finally, a watermark might be read out by monitoring the temperature radiation which is similar to power and EM-field watermarking approaches. There is only one commercial watermarking approach which reads a watermark from an FPGA taking up to 10 minutes (Kean et.al, 2008).

3. Conclusion

In this section, we first reviewed several classical IP protection methods such as tagging, fingerprinting, and watermarking. Then we investigated representative watermarking techniques of ASIC at different design levels. We proposed functions to evaluate watermarking techniques from the under aspects: embedding cost, overhead, coincidence probability, security and tracing cost. The evaluated results show that the performance of physical watermarking technique is high, structural watermarking technique is medium, and behavioral watermarking technique is low. We also summarized watermarking techniques of FPGA core protection and validation methods from three forms of FPGA: source code, netlist, and bitfile.

From this work, we hope it provides a standard candidate for researchers to evaluate their watermarking techniques. In future, researchers may develop stronger watermarking techniques by combining the advantages of different level watermarking techniques to prevent any IP piracy attempt from happening.

4. Acknowledgment

We are grateful to our work team for the contribution to this research. The Project is sponsored by SRF for ROCS, SEM. and Supported by Shandong Province Natural Science Foundation (ZR2009GL007) and A Project of Shandong Province Higher Educational Science and Technology Program (J09LG10). The author also thanks for the support of the family.

5. References

Abdel-Hamid, A.T.; Tahar, S. & El, M.A. (2003). IP watermarking techniques: survey and comparison. *Proceedings of IWSOC2003 3rd IEEE Int. Workshop on System-on-Chip for*

Real-Time Applications, pp.60–65, ISBN 0-7695-1944-X, Calgary, Alberta, Canada, June 30-July 2, 2003

Abdel-Hamid, A.T.; Tahar, S. & El Mostapha Aboulhamid. (2006). Finite state machine IP watermarking. *Proceedings of AHS 2006 1st NASA／ESA Conference on Adaptive Hardware and Systems,* pp.457-464, ISBN 0-7695-2614-4, Istanbul, Turkey, June 15-18, 2006

Adarsh, K.J. ; Lin, Y. ; Pushkin R.P. & Gang Q. (2003). Zero overhead watermarking technique for FPGA designs. *In GLSVLSI '03: Proceedings of the 13th ACM Great Lakes symposium on VLSI,* pp. 147–152, ISBN 1-58113-677-3, USA, 2003

Aijiao, C. & Chip-Hong, C. (2006). Stego-signature at logic synthesis level for digital design IP protection, *Proceedings of 2006 IEEE International Symposium on Circuits and Systems,* pp. 4611-4614, ISBN 0-7803-9389-9, Island of Kos, Greece, May, 2006

Andrew, E. C.; Hyun-Jin, C.; Andrew, B. K.; Stefanus, M.; Miodrag, P.; Gang, Q. & Jennifer, L. W. (1999). Effective Iterative Techniques for Fingerprinting Design IP, *Proceedings of the 36th annual ACM/IEEE Design Automation Conference,* pp. 208-215, ISBN 1-58113-109-7, New York, NY, USA, 1999

Bolotnyy, L. & Robins, G. (2007). Physically unclonable function-based security and privacy in RFID systems. *Proceedings of PERCOM 2007 5th IEEE International Conference on Pervasive Computing and Communications,* pp.211-220, ISBN 0-7695-2787-6, Washington, DC, USA, March 19-23, 2007

Chapman, R. & Durrani, T.S. (2000). IP protection of DSP algorithms for system on chip implementation. *IEEE Trans. on Signal Processing,* vol. 48, No. 3, (March 2000), pp. 854-86 1, ISSN 1053-587X

Daniel, Z. & Jurgen T. (2006). Evaluation of Watermarking methods for FPGA-based IP-cores. *Technical Report 01-2006,* Erlangen, Germany, Mar, 2006

Daniel, Z. & Jurgen, T. (2010). New Directions for FPGA IP Core Watermarking and Identification, *In Proceedings of Dagstuhl Seminar 10281,* 2010

Darko, K. & Miodrag, P. (1998). Intellectual property protection using watermarking partial scan chains for sequential logic test generation, *Proceedings of 1998 International Conference on Computer-Aided Design ICCAD,* 1998

FallWorldwide Member Meeting: (1997). A Year of Achievement (Guidelines Proposed by VSIA Development Working Group on Intellectual Property Protection). VSI Alliance, Santa Clara, CA, 1997

Gang, Q. & Miodrag, P. (1999). Effective iterative techniques for fingerprinting design IP, *Proceedings of Design Automation Conference,* pp. 587–592, ISSN 0278-0070, Los Angeles, CA, June, 1999

Gang, Q. & Miodrag, P. (2003). *Intellectual Property Protection in VLSI Design: Theory and Practice,* Kluwer Academic Publishers, ISBN 978-1-4020-7320-5, USA

Irby, D.L.; Newbould, R.D.; Carothers, J.D.; Rodriguez, J.J. & Holman, W.T. (2000). Low level watermarking ofVLSI designs for intellectual property protection. *Proceedings of IEEE 13th lnt ・ASlC/SOC Conferenc,* pp. 136 – 140, ISBN 0-7803-6598-4, Arlington, VA , USA, September, 2000

John L.; William H. M. & Miodrag P. (1998). Signature hiding techniques for FPGA intellectual property protection. *In proceedings of ICCAD International Conference on Computer-Aided Design,* pp. 186–189, ISBN 1-58113-008-2, California, USA, 1998

John L.; William H. M. & Miodrag P. (1999). Robust FPGA intellectual property protection through multiple small watermarks. *In proceedings of DAC99 Design Automation Conference,* pp. 831–836, ISBN 1-58113-092-9, USA, 1999

John, L.; Miodrag P. ; William, H.M. & Miodrag, P. (2001). Fingerprinting Techniques for Field-programmable Gate Array Intellectual Property Protection. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* Vol.20, No.10, (October 2001), pp. 1253-1261, ISSN 0278-0070

Kahng, A.B.; Mantik, S.; Markov, I.L.; Potkonjak, M.; Tucker, P.; Huijuan, W. & Wolfe, G. (1998). Robust IP watermarking methodologies for physical design. *Proceedings of DAC 35th Design Automation Conference,* pp.782-787, ISBN 0-89791-964-5, San Francico, California, USA, June 15-19, 1998

Kahng, A.B.; Lach, J.; Mangione-Smith, W.H.; Mantik, S.; Markov, I.L.; Potkonjak, M.; Tucker, P.; Wang, H. & Wolfe, G. (1998). Watermarking techniques for intellectual property protection. *Proceedings of DAC98 35th ACM/IEEE Design Automation Conference,* pp. 776–781, ISBN 0-89791-964-5, San Francisco, CA, USA, June 15-19, 1998

Kahng, A.B.; Lach, J.; Mangione-Smith, W.H.; Mantik, S.; Markov, I.L.; Potkonjak, M.; Tucker, P.; Wang, H.; & Wolfe, G. (2001). Constraint-based watermarking techniques for design IP protection. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* Vol.e 20, No. 10, Oct. 2001, pp. 1236-1252, ISSN 0278-0070

Kean, T.; McLaren, D. & Marsh C. (2008). Verifying the Authenticity of Chip Designs with the DesignTag System. *In Proceedings of the 2008 IEEE International Workshop on Hardware-Oriented Security and Trust,* pp. 59-64, ISBN 978-1-4244-2401-6, Washington, USA, June, 2008

Kirovski, D. ; Liu, D. ; Wong, J.L. & Potkonjak, M. (2000). Forensic Engineering Techniques for VLSI CAD Tools, *Proceedings of 37th ACM/IEEE Design Automation Conference,* pp. 581-586, ISBN 1-58113-187-9, Los Angeles, CA, June, 2000

Kirovski, D.; Yean-Yow Hwang; Potkonjak, M. & Cong, J. (1998). Intellectual property protection by watermarking combinational logic synthesis solutions. *Proceedings of ICCAD 1998 IEEE/ACM International Conference on Computer-Aided Design,* pp. l94-l98, ISBN 1-58113-008-2, San Jose, CA, USA, November 8-12, 1998

Keating, M. & Bricaud, P. (1998). *Reuse Methodology Manual for System-on-a-Chip Designs,* Kluwer Academic Publishers, ISBN 0792385586, Boston, USA, 1998

Lach, J.; Mangione-Smith, W.H. & Potkonjak, M. (1998). FPGA Fingerprinting Techniques for Protecting Intellectual Property, *Proceedings of the IEEE 1998 Custom Integrated Circuits Conference,* pp. 299-302, ISBN 0-7803-4292-5, Santa Clara, CA, May, 1998

Lin Y.; Qu, G.; Ghouti, L. & Bouridane, A. (2006). VLSI Design IP Protection: Solutions, New Challenges, and Opportunities, *In Proceedings of Adaptive Hardware and Systems 2006,* pp. 469-476, ISBN 0-7695-2614-4, NY, USA, June 15-18, 2006

Lin, Y.; Gang, Q. ; Lahouari, G. & Ahmed, B. (2006). VLSI design IP Protection: Solutions, New Challenges, and Opportunities, *Proceedings of AHS 2006 1st NASA/ESA Conference on Adaptive Hardware and Systems,* pp. 469-476, ISBN 0-7695-2614-4, Istanbul, Turkey, June 15-18, 2006

Majzoobi, M.; Koushanfar, F. & Potkonjak, M. (2008). Lightweight secure PUFs, *Proceedings of Computer-Aided Design 2008,* pp. 670-673, ISBN 978-1-4244-2819-9, San Jose, CA, 2008

Marsh, C. & Kean, T. (2007). A security tagging scheme for ASIC designs and intellectual property cores. *Proceedings of IP-SoC 2006 IP Based SoC Design Conference & Exhibition,* pp. 6-7, France, January 2007

Min, N. & Zhiqiang G. (2004). Constraint-based watermarking technique for hard IP core protection in physical layout design level. *Proceedings of IEEE 7 Int · Conf · on Solid-State and Integrated Circuits Technology,* pp.1360-1363, ISBN 0-7803-8511-X, Beijing, China, October, 2004

Moritz, S. ; Daniel, Z. & Jurgen, T. (2008). Netlist-Level IP Protection by Watermarking for LUT-Based FPGAs. *Proceedings of FPT 2008 International Conference on ICECE Technology 2008,* pp. 20 -216, ISBN 978-1-4244-3783-2, Taipei, China, Dec. 2008

Narayan, N.; Newbould, R.D.; Carothers, J.D.; Rodriguez, J.J. & Holman, W.T. (2001). IP Protection for VLSI Designs Via Watermarking of Routes. *Proceedings of 14th Annual IEEE International ASIC/SOC Conference,* pp.406-410, Washington, DC, USA, September, 2001

Nie, T.; Kisaka, T. & Toyonaga, M. (2005). A watermarking system for IP protection by a post layout incremental router. *Proceedings of DAC 42th Design Automation Conference,* pp.218-221, ISBN 1-59593-058-2, San Diego, CA, USA, June 13-17, 2005

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,* VOL. 20, NO. 9, September, 2001, pp.1101-1117, ISSN 0278-0070

Ravikanth, P.; Ben R.; Jason, T. & Neil, G. (2001). *Physical One-Way Functions,* PhD thesis, Massachusetts Institute of Technology

Skoric, B.; Tuyls, P. & Ophey, W. (2005). Robust key extraction from physical unclonable functions, *Proceedings of the Applied Cryptography and Network Security Conference 2005,* pp. 407-422, ISSN 0302-9743, berlin, 2005

Thomas, H.; Zebo, P. ; Raimund, U. ; & Manfred, G. (2001). Challenges for Future System-on-Chip Design. *Proceedings of ECCTD15th European Conference on Circuit Theory and Design,* pp.173-176, Espoo, Finland, August 28-31, 2001

Thomas W. & Christof P. (2003). How Secure Are FPGAs in Cryptographic Applications. *In Proceedings of International Conference on Field Programmable Logic and Applications (FPL 2003),* Lecture Notes in Computer Science Volume 2778, pp. 91-100, Sept. 2003

Torunoglu, I. & Charbon, E. (2000). Watermarking based copyright protection of sequential functions. *IEEE Journal of Solid-State Circuits,* vol. 35, No. 3, (May 1999), pp.434-440, ISBN 0-7803-5443-5, 2000

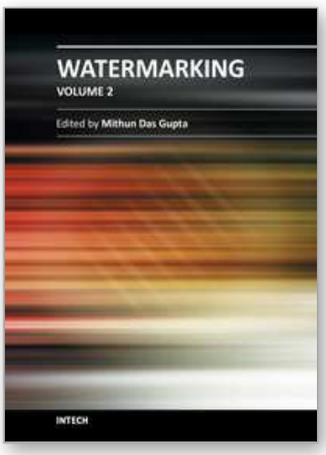
Tuyls, P.; Skoric, B. ; Stallinga, S. ; Akkermans, A. & Ophey, W. (2005). Information theoretical security analysis of physical unclonable functions. *Proceedings of Conference on Financial Cryptography and Data Security 2005,* pp. 141-155, ISSN 0302-9743, berlin, 2005

Virtual Socket Interface Alliance (2000a). Intellectual Property Protection White Paper: Schemes, Alternatives and Discussion Version 1.0. September 2000

Virtual Socket Interface Alliance (2000b). Virtual Component Identification Physical Tagging Standard (IPP 1 1.0). 2000

Ziener, D. & Teich, J. (2008). Power Signature Watermarking of IP Cores for FPGAs. *Journal of Signal Processing Systems,* VOL. 51, 2008, pp.123-136

Ziener, D.; Baueregger, F. & Teich, J. (2010). Using the Power Side Channel of FPGAs for Communication. *In Proceedings of the 18th Annual International IEEE Sympo- sium on Field-Programmable Custom Computing Machines (FCCM 2010),* pp. 237-244, ISBN 978-0-7695-4056-6, Carolina USA, May, 2010



Watermarking - Volume 2

Edited by Dr. Mithun Das Gupta

ISBN 978-953-51-0619-7
Hard cover, 276 pages
Publisher InTech
Published online 16, May, 2012
Published in print edition May, 2012

This collection of books brings some of the latest developments in the field of watermarking. Researchers from varied background and expertise propose a remarkable collection of chapters to render this work an important piece of scientific research. The chapters deal with a gamut of fields where watermarking can be used to encode copyright information. The work also presents a wide array of algorithms ranging from intelligent bit replacement to more traditional methods like ICA. The current work is split into two books. Book one is more traditional in its approach dealing mostly with image watermarking applications. Book two deals with audio watermarking and describes an array of chapters on performance analysis of algorithms.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Tingyuan Nie (2012). Performance Evaluation for IP Protection Watermarking Techniques, Watermarking - Volume 2, Dr. Mithun Das Gupta (Ed.), ISBN: 978-953-51-0619-7, InTech, Available from: http://www.intechopen.com/books/watermarking-volume-2/performance-evaluation-for-ip-protection-watermarking-techniques

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen