WUJNS
Wuhan University Journal of Natural Sciences

Vol.11 No.6 2006 1679-1682

Article ID:1007-1202(2006)06-1679-04

# Using Combinational Logic Function to Watermark IP Based on FPGA

☐ MIAO Sheng, DAI Guanzhong†,
MU Dejun, LI Meifeng

College of Automation, Northwestern Polytechnical University, Xi'an 710072, Shaanxi, China

**Abstract**: In the field of digital circuit design, the extensive applications of reusable intellectual property (IP) simplify the design procedure based on very large scale field programmable gate array (FPGA), and shorten the time to market (TTM). However, the flexibility of reusable IP makes itself easy to be stolen and illegally distributed by intruders. The protection method proposed in this paper maps IP owner's signature to combinational logic functions, and then implements these functions into unused lookup tables (LUTs) in the design based on FPGA, which can be used as a strong proof of IPs ownership. The related experiment results show that this protection method has favorable characteristics such as low overhead, few effects on performance, and high security.

**Key words**: field programmable gate array; watermark; intellectual property

**CLC number**: TP 309

**Received date**: 2006-05-20

**Foundation item**: Supported by the National Defense Basic Scientific Research of China (C2720061361)

**Biography**: MIAO Sheng (1977-), male, Ph. D. candidate, research direction: embedded system design and network security. E-mail: minisheng@sohu.com

† To whom correspondence should be addressed. E-mail: daigz@nwpu.edu.cn

## 0 Introduction

In the field of digital circuit design, field programmable gate array(FPGA) is widely used. With the great improvements in capacity, functionality and credibility, FPGA is also used to solve the problem of high investment in non recurring engineering (NRE) and long time to market(TTM)[1].

With the appearance of very large scale FPGA, design method based on Intellectual Property(IP) reuse has been accepted more and more, meanwhile how to protect IP from being illegally distributed is becoming a serious problem[2]. This article proposed a new method, which maps IP owner's signature to combinational logic function and embeds the function into FPGA, to validate IP's ownership.

## 1 SRAM FPGA

It was reported that the Static Random Access Memory (SRAM) FPGAs have a market share higher than 60%[3], therefore the watermark method proposed in this paper aims to protect IP's ownership of such FPGAs.

As shown in Fig. 1, most SRAM FPGA is composed of configurable logic block (CLB), Input/Output block (IOB) and configurable routing resource (CRR)[4]. The CLB consists of flip flop and LUT.

Each $n$-input LUT can be considered as a $2^n \times 1$ bits RAM, and used to implement an arbitrary combinational logical function with $n$ variables.

## 2 Watermark Generating or Embedding and Extraction

The procedures to generate watermarked IP by using combinational logic function and to extract information from watermarked IP are described in Fig. 1 and Fig. 2.

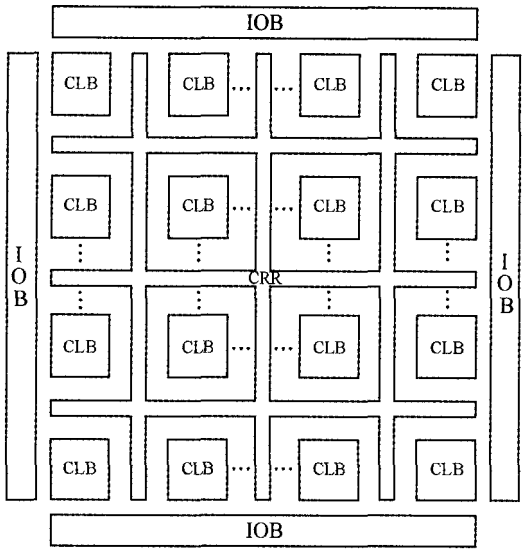



**Fig. 1 The structure of SRAM based FPGA**

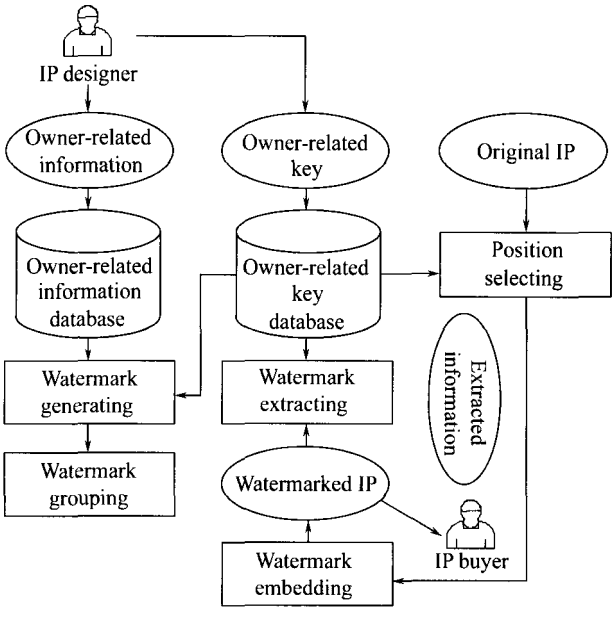



**Fig. 2 Watermark generating, embedding and extracting**

### 2.1 Watermark Generating

Watermark generating procedure is shown in Fig. 3.

Owner-related information is padding with "0" to meet the length requirement of encryption algorithm.

Watermark (WM) is generated by encrypting padded information in Cipher Block Chaining (CBC) mode, with KEY as encryption key, IV as an Initial Vector.

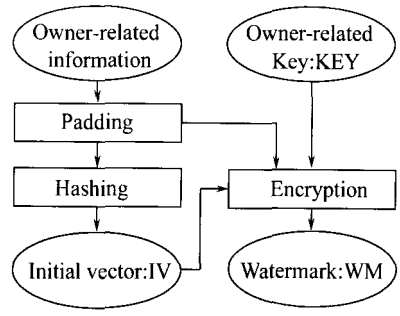



**Fig. 3 Watermark generating**

### 2.2 Watermark Grouping

In the traditional embedding method[5,6], therefore each 4-input LUT always embeds 16 b watermark. In order to improve the camouflage, the WM is divided into groups. Each group has different number of watermark bits, which can be mapped to a combinational logic function and implemented with a 4-input LUT. The pseudo code of watermark grouping is shown as follows:

WM= $wm_1\ wm_2\ wm_{number\_w}$;//in hexadecimal form
$grp_1 = wm_1$; $j=1$;//group index
for ($i=2$; $i<=$number_$w$; $i++$){
if($wm_{i-1} >= wm_i$){$j++$; $grp_j$ =null;}
$grp_j = grp_j$ & $wm_i$;}
number_$g=j$; GRP=$grp_1\ grp_2\ grp_{number\_g}$;

### 2.3 Position Selecting

Embedding position is selected within unused slices. Each slice has 2 LUT with 4 inputs. Therefore the number of position should be:

number_$p$=1+[number_$g$/2]

KEY is used as a seed to select embedding position randomly within unused slices. The pseudo code of position selection is shown as follows:

SLICE={$slice_1$, $slice_2$, …, $slice_{number\_s}$};//unused slices
number_$p$=1+[number_$g$/2];//number of position
if(number_$s$<number_$p$) exit for insufficient slices;
srand(KEY);// set a random starting point
for($i=1$; $i<=$number_$p$; $i++$){
$rnd_i = 1+$rand( )% number_$p$; $pos_i = slice_{rnd_i}$;
for($j=1$; $j<i$; $j++$) if($rnd_i == rnd_j$) $i--$;}
POS={$pos_1$, $pos_2$, …, $pos_{number\_p}$};

### 2.4 Watermark Embedding

Before embedding, each group in Section 2.2 should be mapped to a combinational logic function in terms of minterms, in which one hexadecimal bit is mapped to a corresponding minterm, Then the function will be implemented in the corresponding position.

The pseudo code of embedding $grp_i$ is shown as follows:

$grp_i = wm_{i1}\, wm_{i2}\, wm_{ij}$ ;//the group

$val\_ij =$ HEX2DEC ($wm_{ij}$);//the index of minterm

$fun_i = m_{val\_i1} + m_{val\_i2} + \cdots + m_{val\_ij}$ ;//the logic function

implement $fun_i$ in $pos_{1+[i/2]}$;

Because there are few conjunctions between embedding positions and original design positions, intruders are likely to find out the embedding positions easily. Although the embedded watermark can not be deduced only from the positions, intruder can mangle the embedded watermark without affecting the original design's function. In order to hide embedding position, additional routing constrains should be randomly added without affecting original routes and layouts. With the help of extra constrains, embedding positions resemble a part of original design, which make it more difficult for intruders to detect.

### 2.5 Watermark Extracting

In order to validate IP's ownership, embedded watermark should be extracted as follows.

① Finding out the embedding position with the help of KEY: $POS = \{pos_1, pos_2, \cdots, pos_{number\_p}\}$;

② Extracting logical function embedded in each LUT: $FUN = \{fun_1, fun_2, \cdots, fun_{number\_g}\}$;

③ Writing $fun_i$ in terms of minterms with ascending order; and then mapping $fun_i$ to $grp_i$ in hexadecimal form, in which each minterm is mapped to a corresponding hexadecimal bit;

④ The watermark is: $WM = grp_1\, grp_2 \cdots grp_{number\_g}$;

⑤ Reverse watermark generating process to get extracted information.

## 3 Experimental Results and Analysis

The following properties can be use to evaluate the performance of an IP protection method [7,8].

### 3.1 Overhead

Overhead is the cost induced by watermark embedding, which can be categorized as physical cost and design cost.

In this paper, physical cost is the additional slices used to embed watermark. Fig. 4(a),(b) are the growth ratio of slice respectively caused by embedding different length of watermark in b05 of International Test Conference 1999 (ITC99) [9,10], and by embedding same length of watermark in different designs of ITC99.

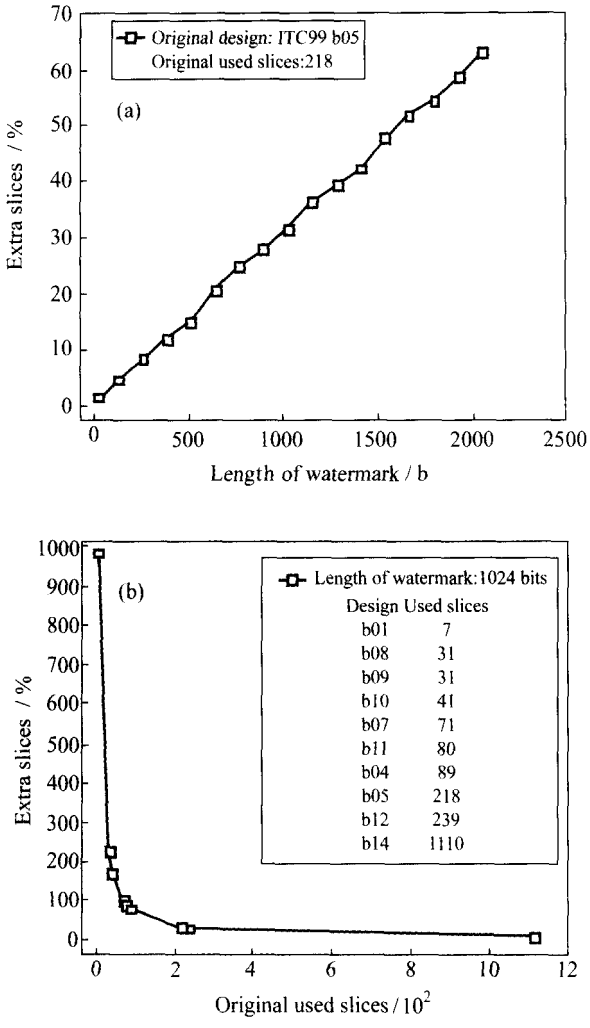



**Fig. 4 Physical cost caused by watermark embedding**

(a) Different watermarks in same design;

(b) Same watermark in different designs

The conclusion can be summarized as: the growth ratio of slice is in direct ratio to the watermark length, while is in reverse ratio to the number of slices needed by original design.

Design cost is the additional design time caused by watermark embedding. According to the embedding procedure, design cost should be less. The experimental result shown in Fig. 5 can be used as a proof.

### 3.2 Effect on Performance

This property is used to describe the performance degradation caused by watermark embedding (Fig. 6).

As mentioned before, the watermark embedding procedure will not change the original design's route and layout; therefore, there should no effects on the performance of original design.

Related experimental results are shown in Table 1 and Table 2.

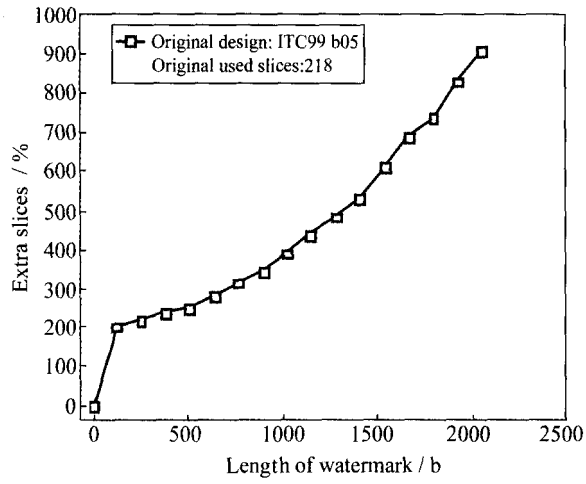
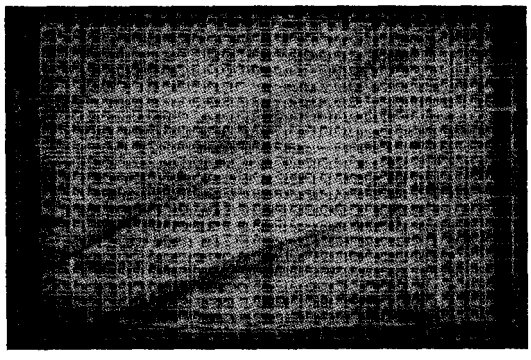
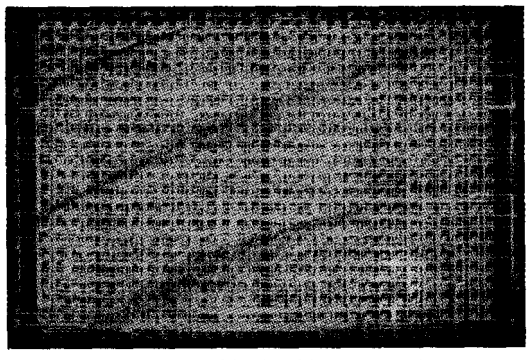



**Fig. 5 Design cost caused by watermark embedding**



(a) Original layout



(b) Layout with 1024 bits watermark

**Fig. 6 Layout of b14 of ITC99**

**Table 1 Different watermarks in same design b05**

| Mark size/b | $F_{max}$ /MHz |
|---|---|
| 0 | 57.2 |
| 128 | 57.2 |
| 256 | 57.2 |
| 512 | 57.2 |
| 768 | 57.2 |
| 1024 | 57.2 |
| 1280 | 57.2 |
| 1536 | 57.2 |
| 1792 | 57.2 |
| 2048 | 57.2 |

**Table 2 Same watermark in different designs** MHz

| Design | $f_{max}$ | |
|---|---|---|
| | Before embedding | Aftter embedding |
| b01 | 128.0 | 128.0 |
| b04 | 62.2 | 62.2 |
| b05 | 57.2 | 57.2 |
| b07 | 94.6 | 94.6 |
| b08 | 104.2 | 104.2 |
| b09 | 105.1 | 105.1 |
| b10 | 119.8 | 119.8 |
| b11 | 75.4 | 75.4 |
| b12 | 73.8 | 73.8 |
| b14 | 25.1 | 25.1 |

### 3.3 Security

Security is the ability of watermark to resist different hostile attacks. During the watermark embedding procedure, extra routing constrains have been added between watermark positions and original design positions Fig. 6. Those constrains will make it more difficult for intruder to locate the embedding positions, so as to damage or extract watermarks.

## 4 Conclusion

In the field of IP design, IP owners pay more attentions to protecting their designs from being illegal distributed. The method proposed in this paper can do such protection by embedding watermark into unused LUTs, and has been proven to have good characteristics such as low overhead and few effects on performance.

## References


[1] EE Times. 2005 EDA Survey Report [EB/OL]. [2006-03-21]. *http://www.eet.com/edasurvey.*

[2] VSIA. IPP: Schemes, Alternatives and Discussion [EB/OL]. [2001-01-08]. *http://www.vsi.org/documents/index.htm.*

[3] Bossuet L, Gogniat G, Burleson W. Dynamically Configurable Security for SRAM FPGA Bitstreams [C]//18*th International Parallel & Distributed Processing Symposium*. Santa Fe, New Mexico, Apr. 26-30,2004:175.

[4] Xilinx Inc. Spartan-II Data Sheet [EB/OL]. [2005-08-15]. *http://www.xilinx.com/bvdocs/publications/ds001.pdf.*

[5] Lach J, Mangione-Smith W H, Potkonjak M. Enhanced Intellectual Property Protection for Digital Circuits on Programmable Hardware [C]// *International Workshop on Information Hiding*. Dresden, Sep. 29-Oct 1, 1999:65-70.

[6] Lach J, Mangione-Smith W H, Potkonjak M. *Fingerprinting Digital Circuits on Programmable Hardware* [M]. Berlin:Springer-Verlag, 1998:16-31.

[7] Ingemar J C, Matthew L M, Jeffrey A B. *Digital Watermark* [M]. Beijing: Electronics Industry Press, 2003 (Ch).

[8] Jain A, Yuan L, Pari P, *et al*. Zero Overhead Watermarking Technique for FPGA Designs [C]// *Proceedings of the 13th ACM Great Lakes Symposium on VLSI*. New York: ACM Press, 2003: 147-152.

[9] Corno F, Sonza R M, Squillero G. RT-Level ITC 99 Benchmarks and First ATPG Results [J]. *IEEE Design & Test of Computers*, 2000,**17**(3):44-53.

[10] Scott D. ITC99 Benchmark [EB/OL]. [2005-10-20]. *http://www.cerc.utexas.edu/itc99-benchmarks/bench.html.*


□