A practical authentication method of FPGA designs suitable for mass production

Orhun Süzer[1], Gürkan Kaya[1], Faruk Dönmez[3], Gökay Saldamlı[2], Müştak E. Yalçın[1]

[1]Istanbul Technical University, Turkey
{suzero,kayagu,mustak.yalcin}@itu.edu.tr
[2]Boğaziçi University, Turkey
gokay.saldamli@boun.edu.tr
[3]Vestek Electronics
faruk.donmez@vestel.com.tr

Abstract

Traditionally, IC companies were able to protect their IP assets by simply keeping these in safe. However, the new trends such as outsourcing and fabless IC development make the silicon processing and IC development more accessible. Therefore, IC authentication and IP (intellectual property) protection have become real world problems that industry eagerly seeks for efficient solutions. Most of the considerable proposals to these intense problems involve complicated cryptographic schemes and procedures that bring extra burden on system design. Moreover, if the target platform is a constraint environment, this burden is amplified and even the most efficient solutions become infeasible. Therefore, designers tend to use the ad-hoc methods that possibly have serious security risks. In this study, we seek practical solutions for the FPGAs (Field Programmable Gate Array) which represent a relatively small but important subset of hardware IP utilization.

1. Introduction

The forgery or counterfeiting activities in the world is increasing day by day. Regardless of any particular sector or area in question, these activities are possible be appeared everywhere. Textiles, accessories, electronics goods, everything from pharmaceuticals to brand names could easily be copied or imitated. The cost of all the fraudulent activities is roughly estimated more than 500 billion dollars.

According to the OECD (Organization for Economic Co-operation and Development) reports [1], about 5-7% of the whole world trade are made through illegal counterfeiting activities. Moreover, this stake is increasing 10% every year. One of the most affected industries is the electronics industry, particularly consumer electronics, integrated circuits and all kinds of game consoles (IC) are often encountered or counterfeiting. In the most recent example of Intel's new processor i7-920 package counterfeited chips are offered for sale to the market and Intel confirms this event by giving an explanation about these chips.

Naturally, counterfeit products and counterfeit producers having no R&D investment has a much shorter time to market cycles. Moreover, these parties do not need to work on product creation-planning through design and revenue generation efforts. Hence, industries such as electronics, suffering from significant R&D efforts and high investment costs face the loss of market claims. In addition, the result of counterfeit and fake products could be cause losses in these companies's brand name, value and prestige.

Although counterfeiting does not harm only the electronics industry, because of its critical position and importance of know-how counterfeiting should be addressed specifically for this industry. In particular, as more and more companies are increasingly outsourcing or moving to the Asian countries while Asian countries (excluded Middle East) are responsible around 70 % for all counterfeit products [1]. In other words if this issue is not considered seriously, the losses would likely to increase.

Because of its re-usability, shorter time to market, flexibility, and many other desirable features, FPGA technology has attracted considerable interest in the electronics industry. On the other hand, just because of some side effects, the same features may cause the theft of IPs which poses a major threat to IC companies. In order to address these pitfalls, Xilinx Inc. introduces a new solution for Spartan 3A, 3AN and Spartan 6 family FPGAs called as the device DNA –- i.e. is an identifier value that consists of 57 bits ---. This embedded, nonvolatile and temper resistant value is programmed into the FPGA during the manufacturing process [2] and it is an attribute to the DNA structure in the organisms. This value can be easily obtained externally through JTAG port or internally through primitive and dedicated shift register that the DNA is stocked in. Mainly Device DNA can be used to distinguish every single device from others and by this way it is aimed at protecting the devices and designs from overbuilding or copying. The protection method using Device DNA is based on a security algorithm, previously created and stored check value in a nonvolatile area and comparator module to compare stored check value and active value on every usage attempt to the desired device [3].

On the other hand, since Spartan 3 FPGA family, Xilinx offers a capability called “multiboot” which means switching between different design files that reside in an external memory(ies) under the control of FPGA [2]. To do that, after the desired design files are generated, these files should be integrated and a memory file (.mcs) should be generated. After generation of memory file, it is programmed to the external memory and it starts to work. On the fly, FPGA “current design” triggers a multiboot event on a desired moment and FPGA reconfigures itself towards the next memory address that includes the design to be downloaded as “next design”.

Our contribution: Even the existence of Device DNA, it is not practical to use this DNA value for high amount of production. It is not possible to read the Device DNA of every single device, to encrypt or hash it through a specific security algorithm and then to write it to a specified area of a nonvolatile memory as it is offered by Xilinx [3]. In this work we propose a practical, hardware cost conscious and also suitable (for mass production) security precaution against copying and overbuilding, using Device DNA and multiboot capabilities together. Moreover, in the work, thanks to a basic self destruction mechanism, the main authentication value generating system is also corrupted after being used only once. Hereby a security structure is also added to the system.

2. The use of device DNA

The idea is based on two different designs and one data array located in three different areas of an external memory. The memory that used to boot FPGA is separated into 3 main parts. In the first part of the memory, the "Controller_Design" design resides. The residing design in the second part is "One_Time_Design" and the third part is used to store encrypted values as "stored check value". In other words the first and second parts of the memory include two designs that actually work and third part is only used to write, store or read by these designs.

One of the two main hardware design sections, first section called "Controller_Design" consists of "data segment controller", "multiboot trigger", "active value obtainer" and "comparison" parts. On the other hand, the second main hardware section called "One_Time_Design" consists of "DNA" and "memory" parts. Respectively, the first hardware section is in charge of controlling and deciding according to the relationship between data segment value and active value. The second hardware section is in charge of constituting the check value that will be reserved in third part of the memory and used as reference value to authenticate the device or design.

In short, OTD* is responsible for constituting of reference values and CD** is responsible for comparison of these reference values with active values, and deciding the permit of authentication.

Case 1: First Turn-on after programming – Start Up
When the complete design is used for the first time after programming the memory, both the first and the second parts of the memory include the significant boot files (CD and OTD) for FPGA core. The third part of the memory is blank at this time; because of the reference check values are not generated yet. When the device is energized after programming, it is always booted from CD. In the normal operating of the CD, "data segment controller" part checks the data segment address and it reads the needed bits of the data segment. Due to the OTD has not worked yet and data segment is blank at this time, a multiboot is triggered by the CD to run the OTD. After multiboot event is triggered, the device downloads OTD and it configures itself as it is declared in OTD. When OTD design starts, first it reads the Device DNA value of the device and dopes it with zeros until it becomes 64 bits. Afterwards, under the DNA part, 64 bits of doped DNA value is transferred to the encryption submodule. In the encryption submodule, TEA (Tiny Encryption Algorithm) which consists of Feistel rounds is used as a small, simple and strong cryptographic algorithm [4]. The 64 bit received value becomes the input value for the encryption algorithm and as an output; encryptor submodule generates an encrypted value. This value should be kept to be used as reference check value for the next attempts of usage. That's why, just after DNA part ends, encrypted 64 bits value is transferred to the memory module. On this step, it is written to the data segment in the memory by the writer submodule. When the writer submodule finishes write operation to the data segment, this value becomes "stored check value" and writer submodule activates the erase submodule to erase the start page of OTD in the memory. Due to the fact of it is a kind of self destruction, after erase submodule is executed, re-starting of OTD is prevented. Hereby, another authentication or production of reference check value for another system is also prevented. After erase operation, the device should be restarted or a multiboot event should be triggered to jump to the first design again.

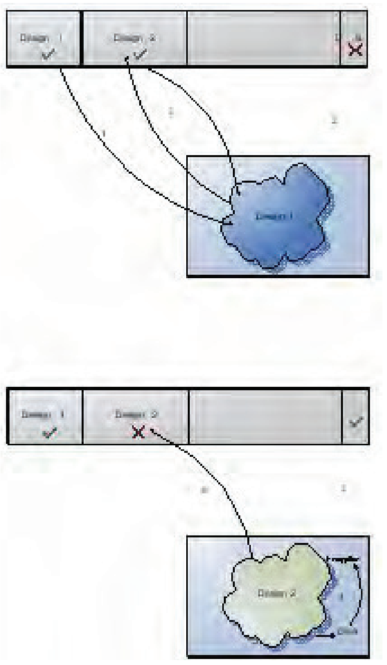

Figure 1. Just after mass production processes, first time energizing, start up

1: Downloading and starting Design 1
2: Checking Data Segment
3: Creating Multiboot event due to the blank Data Segment and jumping to Design 2
4: Obtaining and encrypting DNA value
5: Writing the value to the Data Segment
6: Erasing Design 2

Case 2: Second or more Turn-on – Regular Usage
After the restart or multiboot event towards CD, FPGA starts as regular usage, due to the fact that it has already energized once and this time is for second time or more. It is also the reason for considering this usage as "Regular Usage" of the device. After device is turned on in this scenario FPGA downloads the CD which resides in first part of the non-volatile memory (stated as next address). Then data segment controller reads the data segment again and gets stored_check_value. Due

to the Data Segment is not blank, it is considered as sensible data (stored_check_value) and Design_1 does not create any multiboot event. After data segment controller reads and decides not to create any multiboot event, it sends the stored check value to the comparison part. In parallel, active value obtainer part, works in the same way as in OTD and it reads the device DNA and then encrypts it through encryption part and gets the "active value" as 64 bits. Afterwards this value is also sent to the comparison part. Comparator is in charge of comparing obtained "active_value" with read "stored check value" and deciding whether the device should be authenticated or not. If there is a requested relation between "active value" and "stored check value", it means the device should be considered as authenticated or vice versa. If the values provide this relation, then CD activates the design that realizes the real act. If values do not provide the relation, outputs can be disabled, limited functionality or active defense can be applied too. [3]

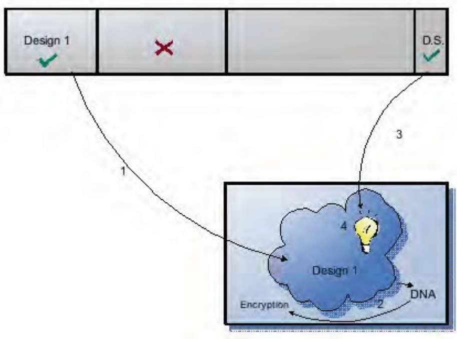

Figure 2.Regular user attempt to use the design

1: Downloading Design 1
2: Obtaining and encrypting DNA Value
3: Reading stored check value
4: Comparing and taking a decision whether to authenticate or not

3. Implementation

In the work, an implementation of the concept is also realized. The chosen platform for the work is Spartan 3A Starter Kit, equipped with a XC3S700A (Spartan 3A) device and the memory to use as multiboot memory is chosen as Atmel AT45DB161D (16Mbit Atmel memory) which is already on the board. As the concept is explained in a detailed way above, there are two different designs and three main parts of the memory block. The blank part between first and second design parts is an obligation in the situations that Digital Clock Manager [2] is used in the designs of multiboot. The design files are indicated in a visual way in Figure 3 and Figure 4. The placement, start and end addresses of the design segments, blank segments and data segment are also declared in Figures 1 and 2.

As a security algorithm, the 128 bit key, 64 bit plaintext and 32 rounded version of TEA is used. (Key produce) The design which is requested to be protected is also instantiated into the Design_1 and it receives the activation signals directly from the inside of the control design. Under these choices and conditions, except of the real design, the whole cost of the designs are below.

Table 1. Implementation results

	Design 1	Design 2
FF	539 (4%)	367 (3%)
LUT	689 (5%)	642 (5%)
Slice	540 (9%)	390 (6%)
ICAP	1 (100%)	1 (100%)
DNA	1 (100%)	1 (100%)

Due to the second design is used only once and after that only first design which is associated with the real design is going to be used, the values in the first column should be considered. As it is also shown in the columns, the extra cost of the design is less then 10% of the sources in the gate array. Every FPGA configuration file (.bit) occupies 318Kbyte space in the case of any of the data compression capabilities are not used. On the other hand, when it is converted to a .mcs file, it occupies 918Kbyte of the non-volatile memory. The multiboot memory file (.mcs) which includes both of Design_1 and Design_2 occupies approximately 1.9Mbyte space of the non-volatile external memory. However just after the first time usage, due to the destruction of second configuration file, more than half of the 2Mbyte memory will be free to use again.

4. Conclusion

The authentication through Device DNA is until now only applicable by reading, encrypting and checking the values manually. However this method is not applicable in mass production mechanisms. The production line is needed to be changed to provide steps for every steps of authentication like reading DNA, encrypting and then writing it in the non-volatile memory. However, by this work due to all processes are done automatically the idea is suitable for mass production. It does not change any step of mass production. The only changed part is the memory file to be programmed. Moreover, the design is hardware cost effective and practical. Even at the worst case, less than 10% of sources of FPGA are used. Furthermore, due to the self destruction of the second design, it is secure. In short, a security precaution to protect the time and source consuming R&D efforts and IP's in an easy and logic way is offered by this work.

On the other hand, the design can be much more robust by implementing a few improvements. First of all, the number of the bits of DNA can be increased by connecting the DNA output port to DNA input port or combining these values with secure memory ID values. It increases the needed time for the brute attack exponentially. Furthermore, the usage of the secure registers (OTP Registers in memories) is also another option to increase the system robustness. Secondly, the write operation on the memory can be more complicated. For instance, instead of writing only needed bits, the whole page can be fulfilled with senseless content. The sensible content that is used as "previously generated reference value" can be hidden in this senseless content. Thirdly, even the partly erase operation of the second design prevents design to work again, erasing of whole design is more precise move. By erasing whole locations that Design 2 resides, it is possible to increase the precision. On the other hand, even TEA is a tiny and strong algorithm; it has some lack points like equivalent key problem [5]. The increase of the number of rounds, enhancement of algorithm (XTEA or

XXTEA) or replacing the algorithm are also other options to increase the robustness of the system..

7. References

[1] The Economic Impact of Counterfeitin, OECD Pres, 1998. Available online http://www.oecd.org/dataoecd/11/11/2090589.pdf

[2] Spartan 3 Generation Configuration User Guide UG332, Xilinx Inc., 2009

[3] Spartan 3 Security Solutions WP 266, Xilinx Inc., 2008

[4] D. J. Wheeler, R. M. Needham, "TEA, a tiny encryption algorithm". *Lecture Notes in Computer Science* (Leuven, Belgium: Fast Software Encryption: Second International Workshop) 1008: pp. 363–366, 1994.

[5] R. M. Needham and D. J. Wheeler, "Tea extensions." Technical report, Computer Laboratory, University of Cambridge, October 1997.

[6] D. J. Wheeler and R. M. Needham, "Correction to XTEA". Computer Laboratory, Cambridge University, England, October 1998.

[7] Y. Su, J. Holleman, and B. Otis, “A digital 1.6 pj/bit chip identification circuit using process variations,” Solid-State Circuits, IEEE Journal of, vol. 43, no. 1, pp. 69 –77, jan.2008.

[8] Saar Drimer, Volatile FPGA Design Security, 2008 Available online http://www.cl.cam.ac.uk/~sd410/papers/fpga_security.pdf

[9] Design Security for High Volume Applications, Xilinx Inc., 2008