Proceedings of the 5th WSEAS Int. Conf. on Instrumentation, Measurement, Circuits and Systems, Hangzhou, China, April 16-18, 2006 (pp262-267)

An FPGA Intellectual Property Authentication Scheme through Watermarking Techniques

Chang N. Zhang[1] , Xiang Xiao and Fajiang Yu
Department of Computer Science
University of Regina
Telecommunication Research Labs (TRlabs)
Regina, SK, S4S0A2
Canada

Abstract: - This paper presents an intellectual property (IP) protection authentication scheme for field programmable gate array (FPGA) with embedded watermark. The use of digital watermarks to provide ownership (watermarking) and recipient (fingerprinting) identification for IP has become widespread. However, it could not prevent the unauthorized reuses when a person resells the watermarked FPGA design with signature to others. The authentication process in the proposed scheme is based on creating a watermark by encrypting the unique FPGA board ID with the signature of the original IP creator, which guarantees that a watermarked FPGA design can only be used on an authorized FPGA board. In addition, two feasible implementations of the proposed IP authentication scheme are presented.

Key-Words: - Security, Intellectual Property, FPGA, Watermarking, Authentication

1 Introduction

The need to reuse a design IP has grown due to the continuous increase in digital IC system complexity. Design partitioning allows complex systems to be assembled from smaller modules which include parameterized memory systems, I/O channels, ALUs, and complete processor cores. Design reuse has led to the rise of Intellectual Property Protection (IPP) concerns [8]. IP modules are often designed by one company (e.g. Lexra, Altera, Xilinx, VA Research) and sold in a non-physical form (e.g. HDL, netlist, layout) to others, and therefore do not have a natural physical manifestation. The IP blocks are modular and designed to be integrated within other systems, usually on the same chip. As a result of the flexible, intangible nature of these modules, IP copyright protection has become a problem. A thief needs only resell or reuse an IP module, without even reverse engineering the design.

Many current IP protection techniques are based on encrypting the source files. For example, encrypted HDL modules disguise the form and structure from IP users. This allows the IP users the ability to incorporate soft modules and high performance simulation models into their design using a CAD tool provided with the decryption key. However, this approach has been routinely and successfully attacked, often by directly attacking the CAD tool.

Existing FPGA design watermarking [4] and fingerprinting [5] techniques attempt to deter such direct theft and misappropriation of FPGA IP. Digital marks are embedded in a design layout which identify the design origin and specific design instance recipient. These marks allow the IP owner to verify the physical layout as their property and establish the source of misappropriation, in a way that is more compelling than the existing option of verifying the design against a registered database or design history documentation.

These existing FPGA IP protection techniques focused mainly on embedding the mark and keeping it secure [4,5,11]. However, these FPGA watermarking techniques cannot prevent the following wrongdoing: a person can buy the FPGA design with watermarking from an authorized user and then resell it to others without permission of the original creator.

In this paper, we extend the FPGA Intellectual Property Protection Scheme based on watermarking techniques [17] with an authentication process by encrypting the unique FPGA board (chip) ID with the signature of the original IP creator as the key, which guarantees that a watermarked design can only be used on the authorized FPGA chip. This could prevent the unauthorized reuses when certain users

[1] ~~Corresponding author. Professor of~~ Department of Computer Science, University of Regina, Canada. E-mail: zhang@cs.uregina.ca.

violate the copyright protection law and resell the design and signature to others.

In addition, two implementations of the proposed authentication process are presented. The first one is to implement the authentication process in the Embedded Development Kit (EDK), which is a tool suite for intuitive hardware system creation. Another option is to execute the authentication process by making full use of the spare units on FPGA (e.g. MicroBlaze soft processor on Xilinx FPGA board [16]) without burdening the host PC.

2 The Proposed Authentication Scheme

Digital watermarking is a method of embedding secure information on a digital format for the purposes of authentication and security. Any effective watermarking scheme should achieve the following goals [21].

1) The mark must be difficult to remove.
2) It must be difficult to add a mark to a released design
3) The mark must be transparent.
4) The mark must have low area and timing overhead and require little added design effort.

The existing FPGA design watermarking embeds a digital mark in unused lookup tables (LUTs) throughout the design [1,14,15]. These LUTs are incorporated into the design with unused interconnect and neighboring logic block "don't care" inputs, further hiding the signature [5,8,9,10]. The benefits of Properties 1 and 2 can be achieved by integrating the mark into the FPGA design. This integration makes the mark more difficult to detect and remove, as there is no clear distinction between the mark and the functional portions of the design. Similarly, this integration makes it more difficult to add marks. Any attempts to remove the mark or add another incur a great, nearly certain risk of changing the design functionality.

Scanning a large set of diverse FPGA designs for potentially stolen copies can be a laborious process if comparisons are based on functional similarities or widely dispersed watermarks. Fortunately, existing FPGA IP protection techniques restrict mark placement to logic block LUTs, and it can be quickly established where LUTs are located in an FPGA bitstream. Therefore, LUTs can be scanned for ownership marks reasonably efficient [17].

Recently, work has been proposed for the protection of digital circuit IP through watermarking at various levels in the design hierarchy using the superimposition of additional constraints in conjunction with those specified by the user [19,20,21]. Cryptography is used for selecting a subset of FPGA physical design constraints for mark embedding, as it provides probabilistic randomization and therefore protection from added constraints. For this task, the standard cryptography tools from the PGP-cryptography suite, the secure hash function MD5, and the RSA/MIT stream cipher RC4 [13,22] are used.

The scheme introduced in this paper presents a new approach to the authentication function for FPGA board with embedded watermark. Unauthorized users could not bypass the authentication process even if they obtained the design and signatures from the authorized users as different FPGA boards have different board IDs.

In this scheme, the IP user is required to provide the unique board ID of target FPGA board and signature of the original IP creator. An IP creator need to create watermark by encrypting the unique FPGA board ID of authenticated IP user with the signature. This watermark will be embedded during the digital IP design. Meanwhile, the authentication process on the user's side will make sure that the IP design only can be used on the authorized FPGA board. We will first analyze the FPGA design (with embedded watermarks) flow on the IP owner's side and the authentication process on the user's side, which can be summarized by the following steps.

(For IP designers)
1) Create initial non-watermarked design.
2) Extract timing and area information.
3) Get a unique identification number of the FPGA board (such as serial number or product number) from the user of the FPGA design. Also a signature from the original IP creator is required.
4) Encrypt the ID users provide with the signature to produce marks.
5) Establish mark location.
6) Modify netlist for mark locations.
7) Execute vendor place-and-route tools on modified netlist.
8) Embed marks.
9) Incorporate unused logic blocks into design.

Steps 1 and 2 are a part of any digital design flow. The original netlist is mapped by the place-and-route tools, and subjected to timing and area analysis. This timing and area information is later used for

evaluating the overhead incurred due to watermarking.

Mark preparation is then performed in Step 3 and Step 4 by encrypting the unique FPGA board ID with the signature from the original IP creator. For example, each Xilinx Virtex-II Pro FPGA board contains a unique 48-bit serial number that can be used as the MAC address for the board. A reference design is provided to allow the user to obtain the unique serial number from the one-wire serial bus [16]. Therefore, each recipient receives a different instance of the design after embedding different watermarks for each recipient into the same design. This mark uniquely identifies both authorized FPGA board and original design creator, yet is difficult to detect and/or remove.

The RC4 encryption algorithm is used for encryption. In RC4, the keystream is completely independent of the plaintext. A 256 byte array called S-state (S_0 - S_{255}), each of the entries is a permutation of the numbers 0 to 255, and the permutation is a function of the variable length key. There are two counters i, and j, both initialized to 0 used in the algorithm.

Initially S-state is assigned by filling S_1 to S_{255} linearly (i.e. $S_0 = 0$; $S_1 = 1$ ... $S_{255} = 255$).

The S-state is then filled with the key (exacted from the signature of the original IP creator), the key is repeated as necessary to fill the entire array.

The index j is then set to 0
for (i = 0 to i = 255)
j = (j + Si + ki) MOD 256
Swap Si and Sj

The following is used to generate a random byte:
i = (i + 1) MOD 256
j = (j + Si) MOD 256
Swap Si and Sj
t = (Si + Sj) MOD 256
K = St

K is the XORed with the unique FPGA board ID to produce the watermark, or the watermark to produce the board ID.

Mark locations are defined by a design seed for certain user in Step 5. The physical constraints based on the established mark locations are input to the netlist and CAD tool constraints file in Step 5, allowing the modified design to be re-mapped and reprocessed by the place-and-route tool in Step 7. Steps 8 and 9 embed the marks in the appropriate LUTs which are incorporated into the rest of the design by receiving dummy inputs and outputting to neighboring “don’t care” inputs, further hiding the marks.

(For IP users)
1) Provide the unique identification number of the FPGA board.
2) Get the watermarked design for the authorized FPGA board and signature from the IP creator.
3) Implement the authentication process with signature.

Once the authentication process integrated in the EDK (software solution) or FPGA (soft processor solution) is passed, the design tool will allow the users to use the design or incorporate it into other designs.

3 Authentication Process Implementations

Two feasible implementations of the proposed IP authentication scheme have been conducted.

3.1 Software Solution

The software solution is based on the integration of an authentication process in the FPGA design tools. Once the authentication process is passed, the design tool will allow the users to use the design or incorporate it into other designs.

The Xilinx embedded development kit is a complete embedded development solution that includes a library of peripheral IP cores, the award-winning Xilinx Platform Studio tool suite for intuitive hardware system creation, a Built-On Eclipse software development environment, GNU compiler, debugger and more.

We added an authentication function in the menu of EDK, which enables the kit to test the ownership of the design. The user is requested to input the signature, which will be encrypted with the unique serial number of FPGA using RC4 [22]. RC4 is a stream cipher symmetric key algorithm which satisfies the flexible length of signature from the original IP creator. The state table is used for subsequent generation of pseudo-random bytes and then to generate a pseudo-random stream, which is XOR-ed with the unique FPGA board ID to produce the watermark.

3.2 Processor Solution

Many FPGAs provide a soft processor, which unleashes the full potential of embedded FPGA designs. The soft processor on the FPGA can be used to execute the authentication process. One advantage

of this solution is that it makes full use of the resources on the FPGA and achieves robust IP security without increasing CAD tool effort and user design effort. Also, it could avoid the risk of authentication process being compromised by a direct hostile attack to the design tools.

The Xilinx Virtex-II Pro XC2VP4 Multimedia system board was chosen as the example device. The MicroBlaze and Multimedia Development Board is designed to be used as a compact platform for developing multimedia applications. The board supports PAL and NTSC television input and output, true color SVGA output, and an audio CODEC with power amplifier, as well as Ethernet and RS-232 interfaces. Several push button and DIP switches are available for user interaction with the system. The embedded SystemACE controller allows for high-speed FPGA configuration from CompactFlash storage devices.

The Virtex-II Pro XC2VP4 development kit is an ideal solution for FPGA and system designers who need a low cost, easy to use prototype platform. The system board includes four 2.5 Gbps Rocket IO ports, an 8M x 32 SDRAM memory, P160 expansion slot for 110 user I/Os, four clock sources, RS-232 port, and additional user support circuits. Its targeted applications include Embedded Microprocessor, DSP, Telecom/Datacom, ASIC Prototyping and Digital Video.

Figure 1 shows the diagram of the authentication process on the Xilinx Virtex-II Pro XC2VP4 FPGA board.

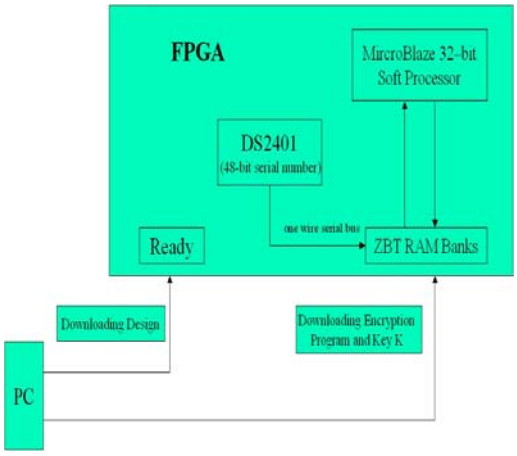

Figure.1 Diagram of Authentication Process on the FPGA Board

Xilinx unleashes the full potential of embedded FPGA designs with the MicroBlaze soft processor solution. The MicroBlaze core is a 32-bit Harvard RISC architecture with a rich instruction set optimized for embedded applications. The processor is a soft core, meaning that it is implemented using general logic primitives rather than a hard, dedicated logic block in the FPGA. The MicroBlaze soft processor is supported in the Xilinx Spartan and Virtex series of FPGAs.

The MicroBlaze solution is designed to be flexible, giving the user control of a number of features such as the cache sizes, interfaces, and execution units. The configurability allows the user to trade off features for size, in order to achieve the necessary performance for the target application at the lowest possible cost.

Each board contains a unique 48-bit serial number that can be used as the MAC address for the board. This serial number is contained in a Dallas Semiconductor "1-wire" Silicon serial number DS2401. A reference design is provided to allow the user to obtain the unique serial number from the one wire serial bus [16]. We can use this unique 48-bit serial number as the FPGA board ID.

The board contains five fully-independent banks of 8M x32 ZBT RAM with a maximum clock rate of 130 MHz. In our scheme, the ZBT RAM is used for buffer to store RC4 encryption program, authentication process and results.

The detailed description of processor solution is as follow.

1) Download the FPGA design to the target FPGA board from the Xilinx EDK.
2) Download the RC4 [22] Encryption Program and signature K to the RAM on the FPGA board after initialization.
3) The encryption program will obtain the serial number (SN) of the FPGA board from the one-wire serial bus and encrypt the unique SN with K as the key using MircroBlaze 32–bit Soft Processor.
4) Store the encrypted result S in the predefined address of the RAM on FPGA board, where the authentication process can reach S.
5) The authentication process will compare S with the watermark embedded in the FPGA design.

Figure 2 explains the detailed flow of authentication process. Once the user provides the correct signature and the authentication process retrieves the correct serial number from the target FPGA board, the authentication procedure will take place. The design will not be functional on the other FPGA board because the serial number of each board is unique.

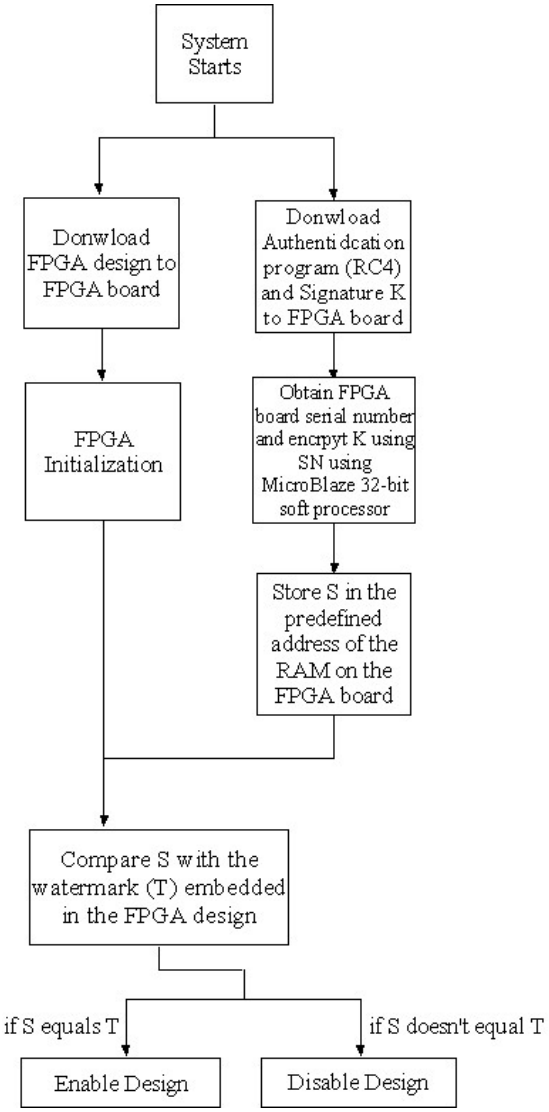

Figure 2. Detailed Explanation of the Authentication Process

4 Conclusion

A new FPGA intellectual property protection authentication scheme through watermarking techniques is proposed. In this scheme, watermarking implements IP protection by making untraceable unauthorized reuse as difficult as recreating IP from small pieces of scratch. The authentication process in the proposed scheme is based on creating a watermark by encrypting the unique FPGA ID with the signature of the original IP creator in order to prevent unauthorized reuse or resell.

Detailed descriptions of two implementations of the authentication process implementation are presented. The first one is software solution and the second one is processor solution. A theoretical analysis shows that the proposed mechanism can protect the rights of IP producers and owners and effectively and efficiently.

References:

[1] W. Bender et al., "Techniques for Data Hiding," IBM Systems Journal, vol. 35, no 3-4, 1996, 313-336.

[2] L. Boney et al., "Digital Watermarks for Audio Signals," International Conference on Multimedia Computing and Systems, 1996.

[3] E. Charbon, "Hierarchical Watermarking in IC Design," Custom Integrated Circuits Conference, 1998.

[4] I.J. Cox et al., "Secure Spread Spectrum Watermarking for Images, Audio, and Video," Internationa Conference on Image Processing, 1996.

[5] S. Craver et al., "Can Invisible Watermarks Resolve Rightful Ownership?" Storage and Retrieval for Image and Video Databases, Proceedings of the SPIE, vol. 3022, 1997, 310-321.

[6] W. Diffie and M. Hellman, "New Directions on Cryptography," IEEE Transactions on Information Theory, vol. IT-22, no. 6, Nov. 1976, 644-654.

[7] S. Furber, ARM System Architecture, Menlo Park: Addison-Wesley, 1996, 329.

[8] R. Goering, "IP98 Forum Exposes Struggling Industry – Undefined Business Models, Unstable Core Prices Cited," EE Times, Issue 1000, March 30, 1998.

[9] F. Hartung and B. Girod, "Copyright Protection in Video Delivery Networks by Watermarking of Pre-Compressed Video," ECMAST'97, Springer Lecture Notes in Computer Science, vol. 1242, 1997, 423-436.

[10] I. Hong and M. Potkonjak, "Behavioral Synthesis Techniques for Intellectual Property Protection," Design Automation Conference, 1999.

[11] B. Hutchings et al., BYUcore: A MIPS R2000 Processor for FPGAs, 1997.

[12] A.B. Kahng et al., "Robust IP Watermarking Methodologies for Physical Design," Design Automation Conference, 1998, 782-787.

[13] B. Schneier, 1963- Applied Cryptography: Protocols, Algorithms, and Source Code in C, New York: John Wiley & Sons, 1996.

[14] J. Lach, W. H. Mangione-Smith, and M. Potkonjak, "Fingerprinting Digital Circuits on Programmable Hardware," International Workshop on Information Hiding, 1998, 16-31.

[15] J. Lach, W. H. Mangione-Smith, and M. Potkonjak, “Signature Hiding Techniques for FPGA Intellectual Property Protection,” International Conference on Computer-Aided Design, 1998.
[16] Xilinx, The Programmable Logic Data Book, San Jose, CA, 1996.
[17] J. Lach, W. H. Mangione-Smith, and M. Potkonjak, “Enhanced Intellectual Property Protection for Digital Circuits on Programmable Hardware”, 1998.
[18] Xilinx, MicroBlaze and Multimedia Development Board User Guide, 2002.
[19] E. Charbon, “Hierarchical watermarking in IC design,” in Proc. Custom Integrated Circuits Conf., May 1998, pp. 295–298.
[20] I. Hong and M. Potkonjak, “Behavioral synthesis techniques for intellectual property protection,” in Proc. Design Automation Conf., June 1999, pp. 849–854.
[21] A. B. Kahng et al., “Watermarking techniques for intellectual property protection,” in Proc. Design Automation Conf., June 1998, pp. 776–781.
[22] G. A. Spanos and T. B. Maples, “Performance study of a selective encryption scheme for the security of networked, real-time video,” in Proc. Int. Conf. Computer Communications and Networks, Sept. 1995, pp. 2–10.