Research Article Impact Factor: 0.621 ISSN: 2319-507X
Flavia Leitao, IJPRET, 2014; Volume 2 (9): 807-815 IJPRET

INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

CRYPTOGRAPHIC SHA-1 HASH IMPLEMENTATION FOR INTELLECTUAL PROPERTY (IP) PROTECTION

FLAVIA LEITAO, DR. AMEETA G. SINAI AMONKAR

1. ME (Microelectronics), Electronics and Telecommunication Engineering, Goa College of Engineering, Farmagudi, Goa
2. Head of the department, Electronics and Telecommunication Engineering, Goa College of Engineering, Farmagudi, Goa

Accepted Date: 27/02/2014 ; Published Date: 01/05/2014

Abstract: In this paper, the detection approach of Intellectual property protection (IPP) called Watermarking is presented. The Watermark or the digital signature is obtained using cryptographic SHA-1 hash function, thus allowing the prevention of forging attacks. The approach used here is to embed the signature at the HDL-level so that it goes down to all levels thus securing all levels and use some cryptographic hash function to prevent it from forging.

Keywords: Intellectual Property, (SHA) Secure Hash Algorithm, Hash value.

PAPER-QR CODE

Corresponding Author: MS. FLAVIA LEITAO

Access Online On:

www.ijpret.com

How to Cite This Article:

Flavia Leitao, IJPRET, 2014; Volume 2 (9): 807-815

INTRODUCTION

Intellectual property (IP) cores are pre-designed, pre-tested, reusable unit of logic, cell, or chip layout design that can be easily used in embedded applications.

As REUSE-BASED design methodology has taken hold, the very large scale integration (VLSI) design industry is confronted with the increasing threat of intellectual property (IP) infringement. IP providers are in pressing need of a convenient means to track the illegal redistribution of the sold IPs. An active approach to protect a VLSI design against IP infringement is by embedding a signature that can only be uniquely generated by the IP author into the design during the process of its creation. When a forgery is suspected, the signature can be recovered from the misappropriated IP to serve as undeniable authorship proof in front of a court.

Such a copyright protection method is widely known as watermarking.

Approaches to secure IPs

1. Deterrent approach: where the owner uses legal means trying to stop attempts for illegal distribution i.e. using patents, copyrights and trade secrets. This method does not provide any physical protection to the IP.
2. Protective approach: where the owner tries to prevent the unauthorized use of the IP physical by license agreements and encryption.
3. Detection approach: where the owner detects and traces both legal and illegal usages of the designs as in watermarking and fingerprinting.

We will be focusing on the Detection approach in this paper.

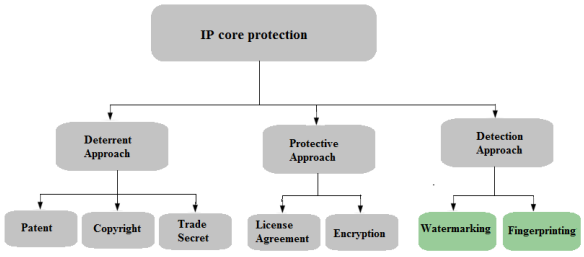


Fig.1. Approaches to secure IPs

PRINCIPLE OF OPERATION

Watermarking Technique:

An active approach to protect a VLSI design against IP infringement is by embedding a signature that can only be uniquely generated by the IP author into the design during the process of its creation. When a forgery is suspected, the signature can be recovered from the misappropriated IP to serve as undeniable authorship proof in front of a court.

Such a copyright protection method is widely known as watermarking.

Fingerprinting Technique

An improvement to the watermarking approach is the fingerprinting technology that enables the owner to see which customer has given the core away

- This is achieved by dividing the FPGA into tiles.
- In each tile one lookup table is reserved for the watermark.
- The position of the mark in the tile encodes the fingerprint.

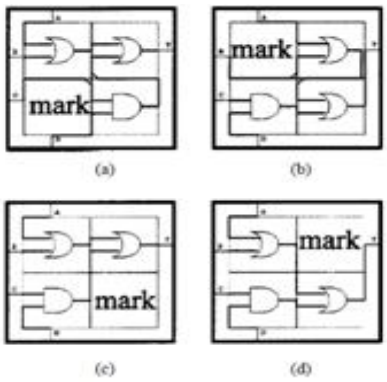


Fig.2. Fingerprinting technique

SYSTEM BLOCK DESCRIPTION

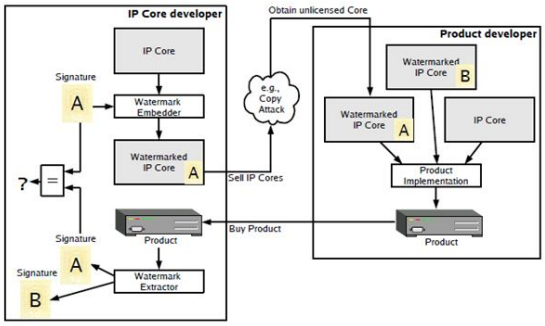


Fig.3. Watermarking technique block diagram

- An IP core developer embeds a signature inside his core and sells the protected IP core.
- A third-party company may obtain an unlicensed copy of the protected IP core and use it in one of their products.
- If the IP core developer becomes suspicious that his core might have been used in a certain product without proper licensing, he can simply acquire the product and check for the presence of his signature.
- If this attempt is successful and his signature presents a strong enough proof of authorship, the original core developer may decide to accuse the product manufacturer of IP fraud and press legal charges.

Watermark Evaluation criteria

- Functional correctness: If the watermark process destroys the functional correctness, it is useless to distribute the core.
- Resources overhead: Many watermark algorithms need some extra resources. Some for the watermark itself, some because of the degradation of the optimization results from the design tools.
- Transparency: The watermark procedure should be transparent to the design tools. It should be easy to integrate the watermarking step into the design flow, without altering the common design tools.
- Verifiability: The watermark should be embedded in such a way that simplifies the verification of the authorship. It should be possible to read out the watermark only with the given product without any further information from the design flow, which must be ordered from the accused company.

- Difficult to remove: The watermark should be resistant against removal. The effort to remove the watermark should be greater than an effort needed to develop a new core or removal of watermark should cause corruptness of the functionality of the core.

Watermarks which are embedded into the function of the core are more robust against removal than additive watermarks.

- Strong proof of authorship: The watermark should identify the author with a strong proof. It should be impossible that other persons can claim the ownership of the core. The watermark procedure must be resistant against tampering.

SHA-1 CRYPTOGRAPHIC HASH FUNCTION OVERVIEW

SHA-1 is a cryptographic hash function designed by the United States National Security Agency (NSA). It produces a 160-bit (20-byte) hash value, which is typically expressed as a hexadecimal number, 40 digits long.

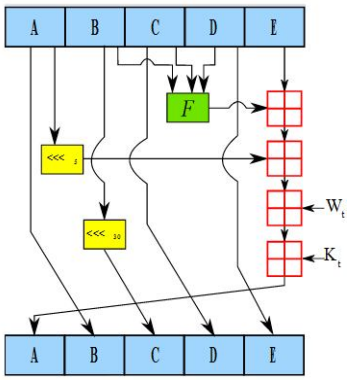


Fig.4. One iteration within the SHA-1 compression function

One iteration within the SHA-1 compression function:

A, B, C, D and E are 32-bit words of the state (160 bits);

F is a nonlinear function that varies;

<<<n denotes a left bit rotation by n places;

n varies for each operation;

Wt is the expanded message word of round t;

Kt is the round constant of round t; denotes addition modulo 232.

Flowchart of the SHA-1 Algorithm

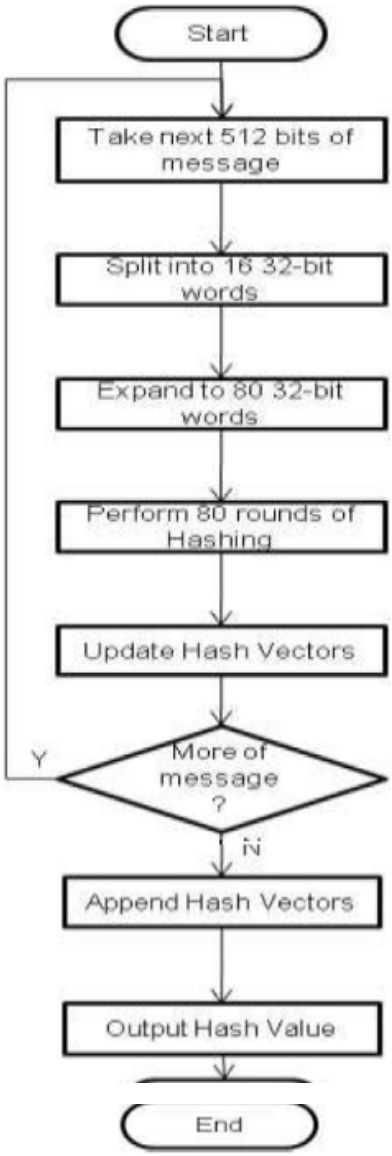


Fig.5. Flowchart of SHA-1 algorithm

SHA1 Cryptographic hash function algorithm

- Padding
  - Pad the message with a single one followed by zeroes until the final block has 448 bits (Mod 512).
  - Append the size of the original message as an unsigned 64 bit integer.

- Initialize the 5 hash blocks (h0,h1,h2,h3,h4) to the specific constants defined in the SHA1 standard.
- Hash (for each 512bit Block)

- Allocate an 80 word array for the message schedule
- Set the first 16 words to be the 512bit block split into 16 words.
- The rest of the words are generated using the following algorithm
  - ✓ Word[i3]XOR word[i8]XOR word[i14]XOR word[i16] then rotated 1 bit to the left.
- Loop 80 times doing the following.
  - ✓ Calculate SHAfunction() and the constant K (these are based on the current round number.
  - ✓ e=d
  - ✓ d=c
  - ✓ c=b (rotated left 30)
  - ✓ b=a
  - ✓ a = a (rotated left 5) + SHAfunction() + e + k + word[i]
    - Add a,b,c,d and e to the hash output.
    - Output the concatenation (h0,h1,h2,h3,h4) which is the message digest.

Digital Signature

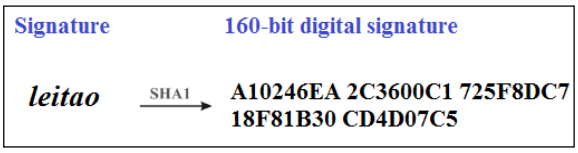
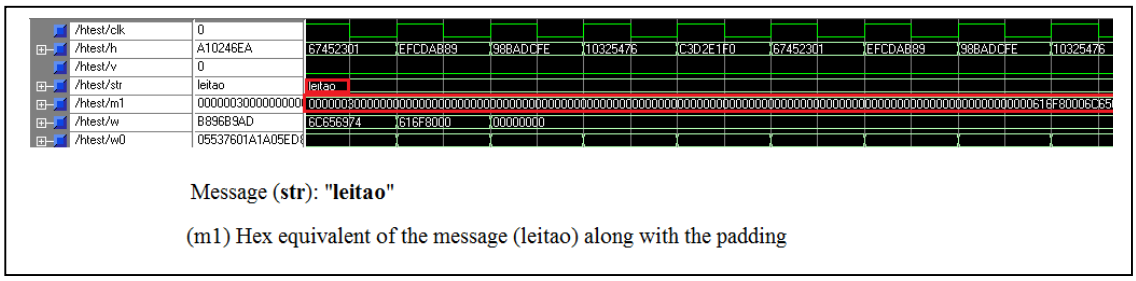


Fig.6 SHA-1 digital signature

SIMULATION RESULTS

The digital signature has been created using the SHA-1 cryptographic hash function. The SHA-1 code has been written in VHDL and simulated using Mentor Graphics Model-Sim5.7g Simulator.

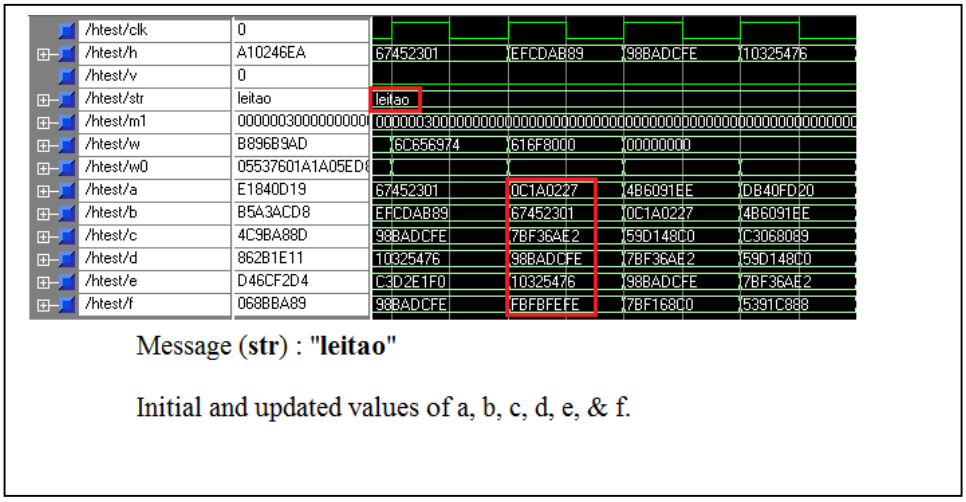
The Simulation results observed are as shown in Fig.7, Fig.8 and Fig.9.



Message (**str**): "**leitao**"

(m1) Hex equivalent of the message (leitao) along with the padding

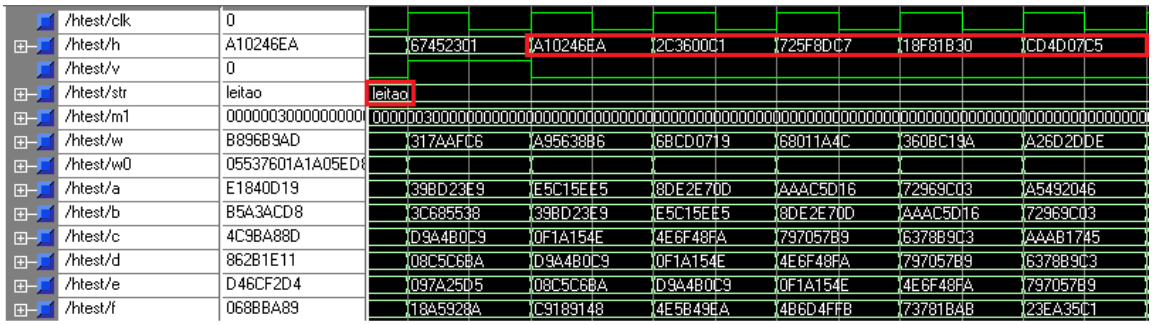
Fig.7 Hex equivalent of the message (leitao) along with the adding.



Message (**str**) : "**leitao**"

Initial and updated values of a, b, c, d, e, & f.

Fig.8 Initial and updated values of a, b, c, d, e & f.



Message (**str**): "**leitao**"

Hash Value (**h**): "**A10246EA 2C3600C1 725F8DC7 18F81B30 CD4D07C5** "

Fig.9 Hash value of the signature "leitao"

CONCLUSION

The digital signature or the watermark using SHA-1 cryptographic hash function has been successfully generated.

Since a cryptographic hash function is applied to the signature or the watermark it is impossible to forge the signature thus making the watermarking technique of IP protection full proof.

ACKNOWLEDGMENT

I am deeply indebted to Dr. Ameeta G. Sinai Amonkar, my project guide and Head of the Department-ETC, Goa college of Engineering whose expertise, understanding, and patience added considerably to my research experience.

I am thankful to Prof. V. N. Shet, Principal, Goa college of Engineering, for his constant support and encouragement.

REFERENCES

1. Daniel Ziener and J¨urgen Teich "Evaluation of Watermarking methods for FPGA-based IP-cores" University of Erlangen-Nuremberg, Co-Design-Report 01-2005

2. Encarnacion Castillo, Antonio Garcia, Luis Parrilla and Antonio Lioris, 2007. "IPP @ HDL: Efficient Intellectual Property Protection Scheme for IP Cores", IEEE Transaction on Very Large Scale Integration Systems, Vol .15, No.5 pp. 578-591.

3. M .Meenakumari & G. Athisha , 2013. " A Survey on Protection of FPGA Based IP Designs ".ISSN (Print) : 2278-8948, Volume-2, Issue-2, 2013

4. http://m.metamorphosite.com/one-way-hash-encryption-sha1-data-software

5. http://en.wikipedia.org/wiki/SHA-1

6. SHA1 TeamFlux.pdf

7. James Docherty and Albert Koelmans, 2011." Hardware Implementation of SHA-1 and SHA-2 Hash Functions", Technical Report Series NCL-EECE-MSD-TR-2011-170.

8. Daniel Ziener, J urgen Teich. "New Directions for FPGA IP Core

9. Watermarking and Identication", Hardware/Software Co-Design

10. Department of Computer Science University of Erlangen-Nuremberg, Germany Am Weichselgarten 3 91058 Erlangen, Germany.