Watermarking Technique for Intellectual Property Protection in FPGA Design

[1]Dr.A.Amutha, [2]A.Keerthana Priyaa, [3]S.A.Sivakumar
[1]Associate Professor, [2]PG Scholar, [3]Assistant Professor
[1,2,3] Department of Electronics and communication Engineering,
[1,2,3] Info Institute of Engineering, Coimbatore, Tamilnadu

Abstract : The Watermarking is an intellectual property (IP) protection technique. It can protect field-programmable gate array (FPGA) IPs from infringement. IP protection of hardware designs is the most important requirement for many FPGA intellectual property vendors. Digital watermarking has become an innovative technology for IP protection in recent years. This paper proposes the publicly verifiable watermarking for intellectual property protection in FPGA design. The chaos-based zero knowledge verification protocol is used in this watermarking detection technique. The time stamping is also used and it can resiliently resist the sensitive information leakage and embedding attacks, and is thus robust to the cheating from the prover, verifier, or third party. The zero-knowledge protocol proposed in this paper is implemented by MATLAB R2014a in which C programming language is used in it and ModelSim 10.5b in which VHDL coding is used in it, are running on a PC. The synthesis tool Xilinx ISE 14.5 is also used to verify and implement the watermarking scheme.

IndexTerms - Intellectual property (IP) protection, field-programmable gate array (FPGA), publicly verifiable watermarking, zero-knowledge protocol.

I. INTRODUCTION

With the prevalence of reusable design methodology in the IC design field, intellectual property (IP) infringement becomes increasingly serious. A modular designed IP cores are easy to be copied or sold by third parties without reverse engineering. In which it results in huge economic losses to IP owners and reduces the market share of their products. Therefore, how to prevent the IP infringement effectively has become a huge challenge for field-programmable gate array (FPGA) vendors and IC designers. However, the existing watermarking techniques may give away sensitive information during the public verification, which enables malicious verifiers or third parties to remove the embedded watermark and resell the design. Various watermarking verification schemes can address the sensitive information leakage issue but are vulnerable to embedding attacks, which makes them ineffective in preventing the infringement denying of un-trusted buyers (verifiers).

In this paper, the new publicly verifiable watermarking detection scheme based on chaotic sequences is proposed to address the issues that the FPGA watermarking technique may leak the sensitive information and the existing zero-knowledge FPGA watermarking detection schemes are vulnerable to embedding attacks. In this scheme first, a watermark is generated with the signature information and then the watermark is embedded into the benchmark circuits based on the embedding algorithm. Next the watermarking overhead and the robustness of position permutation of zero-knowledge protocol are analyzed. The verification scheme proposed in this paper can not only prove that the watermark does exist in IP without revealing its content and position, but any verifier (including un-trusted verifier) can verify the legitimacy of the watermark and resist embedding attacks against the cheating from the prover, verifier, or third party effectively. The experimental results show that the random permutation algorithm has a higher robustness.

II. RELATED WORKS

Since the digital images are very susceptible to manipulations and alterations, a variety of security problems are introduced. For example, the security centre may wish to authenticate the data received from sensors spread across a facility it is supposed to protect. Another common application is resolving the ownership disputes when copyrighted material is distributed illegally. Those problems and the needs can be treated by embedding a secret invisible watermark (WM) in images. A WM is the additional identifying message, covered under the more significant image raw data, without perceptually changing it. By adding the transparent WM to an image it can be made possible to detect alterations inflicted upon the image such as cropping, scaling, covering, blurring and many more. The WM can be added on either a software platform or hardware platform, each having some benefits and some drawbacks. Although the WM implementation on a hardware platform suffers from a limited processing power compared to the software implementation, it features real time capabilities and compact implementations. The advantages of the hardware WM implementations are especially enhanced in CMOS imagers, where it is possible to integrate WM embedded monolithically with the sensor array on the same die.

In the existing work, to prevent the leakage of sensitive information and to enhance the robustness of watermarking are done by using a large number of small watermarks instead of one large watermark. However, this method will leak a part of the set of

watermarking positions after the public verification. When infringement occurs repeatedly, more and more watermarking positions will be given away, which facilitates the attacker to remove more watermarks. The existing publicly detectable VLSI watermarking techniques embeds an independent public watermark for public verification. However, this method is not suitable for FPGA designs because publicly watermarking positions will be leaked after public verification, hence attackers can tamper, remove, or cover the public watermark in the bit-stream of FPGA design, which would result in the wrong verification of IP. The purpose of the public verification is to reduce or eliminate the dependence of one party on the reliability of the other parties, reduce the constraint of the verifier in the protocol, and improve the security of the entire scheme. When a dispute occurs among the parties involved in the protocol, public verification is convenient for the arbitration of a dispute.

III. PUBLICLY VERIFIABLE WATERMARKING SCHEME

In this paper, a new publicly verifiable watermarking detection scheme is proposed and this scheme comprises the following.

1) The watermark is hidden in the unused lookup table (LUT) of the used Slice, and the watermarking content of LUT of ˆ I is encrypted to prevent the leakage of watermarking content.

2) Since the chaotic sequences have high randomness and low cross correlation, we can generate a real number chaotic sequence in each round of verification. The chaotic sequence is binarized into ρ, which is used as an input to the position mapping algorithm π (ρ) to control the location permutation of the LUTs in FPGA bit-stream, i.e., π (ρ) is applied to ˆ I to get the scrambled design ξ. Then ρ and the position of the watermark in ξ are used to interact between the prover and the verifier. With the zero-knowledge protocol, the prover makes the verifier to believe the watermark existing in IP without leaking the position information.

3) Timestamp is introduced to resist the embedding attack to prevent dishonest IP buyers from denying.

3.1Watermarking Generation and Embedding

The process of watermarking generation and embedding are as follows.

Step 1: Watermarking generation - First, the signature S is encrypted with an encryption algorithm. Second, the encrypted S is imputed into a one-way Hash function (such as SHA-2) to generate an abstract Swith fixed length. Finally, the watermark W is obtained by scrambling S with hashed chaotic sequence (the initial value of the chaos is used as the key K1).

Step 2: Locating watermark positions - Using a pseudorandom number generator (such as chaos K2 as the key) to generate a pseudorandom sequence as the watermark embedding positions.

Step 3: The watermark is grouped according to the maximum value of the watermark in an LUT and then embedded into unused LUT of used Slice.

Step 4: The input and output of watermarked ILUTs are connected with the “do not care” inputs of the original circuit in order to disguise the embedded watermark. This is shown in fig.1.

3.2 Chaos-Based Zero-Knowledge Verification Protocol

1) Protocol Overview: Zero-knowledge public verification is to prove that the watermark of IP owner exists in the bit-stream of FPGA design without revealing the watermarking content and position. Assume the prover is Alice and the verifier is Bob. According to the watermark generation and embedding algorithm mentioned, Alice gets W based on S. Then W is embedded into I (FPGA design) to get Iˆ. Alice wants to prove the existence of W without leaking its position. The process of zero-knowledge verification protocol comprises the following.

2) Protocol Implementation: In the zero-knowledge watermarking detection protocol, the random permutation of the FPGA bit-stream is an important component of the zero-knowledge protocol. Random permutation must meet two requirements: (i) the number of random permutations should be enough and (ii) the correlation between random permutations should be extremely low. The implementation of protocol is described in this method by the above requirements. This is shown in fig.2.

3.3 Zero-Knowledge Proof

Zero-knowledge proof or protocol is a method in which a party A can prove that given statement X is certainly true to party B without revealing any additional information. Let say the Alice and Bob want to communicate over shared network. The Alice initiates the communication and sends secret to Bob. Bob verifies the secret so it can be certain that it is communicating with Alice. Once it verifies the secret it sends conformation. In the above scenario, Bob must know Alice secret so it can verify Alice identity but now Bob can impersonate Alice. Zero-knowledge protocol allows Alice to prove Bob that it knows the secret without revealing the secret. In this protocol, verification is performed for many executions and each time Alice needs to pass the verification. Zero-knowledge protocol is three pass identification protocol. The first message is commitment or witness sent from Alice to Bob, a second message is challenge sent from Bob to Alice and a final message is response sent from Alice to Bob. If the statement is true, the honest verifier will be convinced by the honest prover.

The zero-knowledge protocol has three properties. If the statement is true, the honest verifier will be convinced by honest prover and it is called as Completeness. If the statement is false, Trudy cannot convince the verifier that it is true, except with some small probability and it is called as Soundness. If the statement is true no cheating verifier learns anything other than this fact and it is called as zero-knowledge property. The Randomness is also an important property of Zero knowledge protocol. Randomness in the commitment and challenge message are used to hide the secret information. The pros of the Zero-Knowledge Protocol includes,

Secured (not requiring the revelation of one's secret and Simple (does not involve complex encryption methods). The Zero-knowledge proof introduces "The Knowledge Complexity of Interactive Proof-Systems". This method proposes the IP hierarchy of interactive proof systems and conceived the concept of knowledge complexity, a measurement of the amount of knowledge about the proof transferred from the prover to the verifier. It also gives the first zero-knowledge and a way to eliminate the necessity of one way functions. One way with multi-prover interactive proof systems which have multiple independent provers instead of only one, allowing the verifier to "cross-examine" the provers in isolation to avoid being misled.

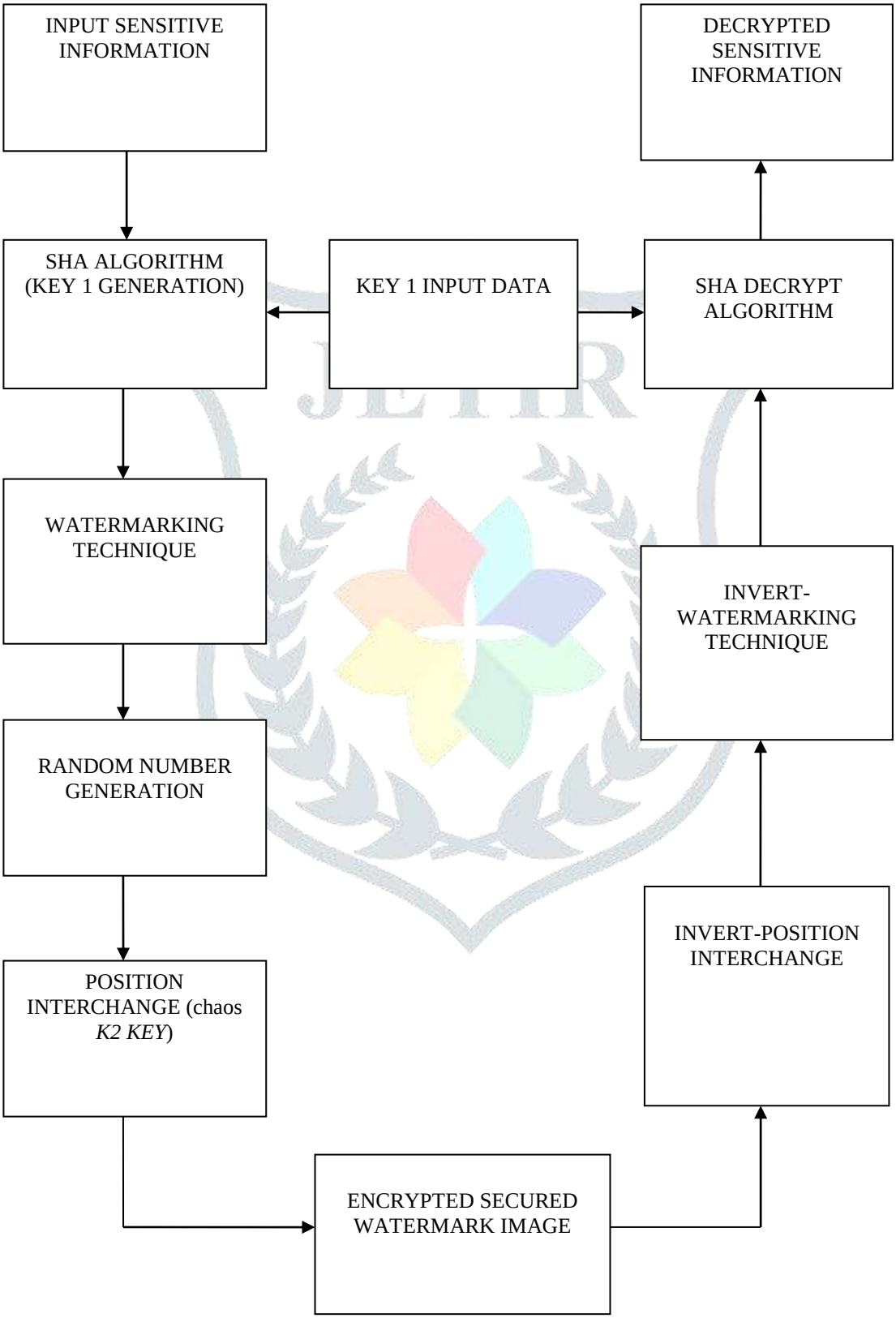


Fig.1 Watermarking technique using SHA algorithm (Zero-knowledge protocol)

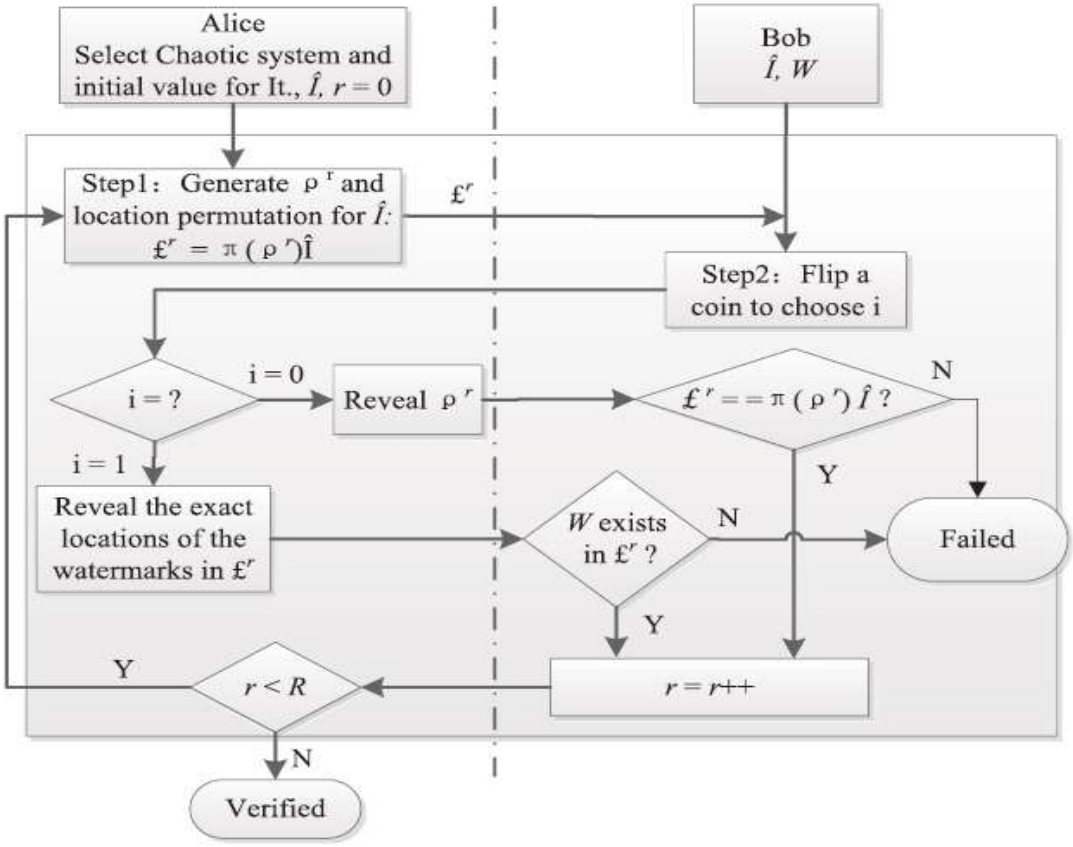


Fig.2 Zero-knowledge verification protocol between prover Alice and verifier Bob

IV. PROTOCOL ANALYSIS

4.1 Analysis of Embedding Attacks

A dishonest IP buyer (verifier) uses an unauthorized IP. An honest IP owner (prover) wants to prove that the IP contains his watermark. In the actual public verification process, the verifier is often un-trusted because the verifier will strive to make an honest IP owner unable to prove his illegal use of IP, i.e., even though the verifier uses the IP illegally, the IP owner is unable to prove it. Since FPGA IP is essentially a bit-stream file, a malicious attacker is able to embed the watermark into the file. Therefore, the existing FPGA zero-knowledge watermarking detection systems are vulnerable to embedding attacks. In order to prevent embedding attacks and denial of infringement, we not only need to watermark the FPGA bit-stream, but also ensure the existence of the watermark before certain time.

We address the problem using the linking or distributed trust time-stamping scheme. The two time-stamping schemes can guarantee that no matter how unscrupulous the time-stamping service (TSS) is, the times it certifies will always be the correct ones, and that it will be unable to issue incorrect time-stamps. The distributed trust time-stamping scheme even could be implemented without the need for a centralized TSS at all. When a copyright dispute occurs, the time that an attacker copies the IP illegally and launches the embedding attack to embed the forged watermark would lag in the genuine time. In the Analysis of Protocol Properties, the zero-knowledge protocol should satisfy three properties - completeness, soundness and zero-knowledge.

4.2 Watermarking Overhead

The resource and time overhead are measured by the used Slice and minimum clock period. The method proposed in this paper does not affect the minimum clock cycle of the design. This is because we embed the watermark in the physical layout of the circuit the routing modification for original design is very small. The routing influence on the design will cause the design to partly change ("hold to clock clk," "setup to clock clk," and "clock clk to pad"), but the minimum clock period keeps unchanged. Therefore, the overhead is almost 0 for our proposed watermarking method, which is an obvious advantage compared with the previous watermarking methods.

Table 1: Resource Overhead Comparison

FPGA	Slices in original IP core	Existing method (previous watermarking techniques)		Proposed method (Zero-knowledge verification protocol)	
		Slices in watermarked IP core	Resource overhead	Slices in watermarked IP core	Resource overhead
XC2V250-6cs144	1231	1263	2.600%	1231	0%
XC2V1000-6bg575	2385	2417	1.342%	2385	0%
XC2V1000-6bg575	2538	2570	1.261%	2538	0%
XC2V1000-6bg575	3100	3132	0.946%	3100	0%
XC6VCX75t-2ff484	3320	3352	0.865%	3320	0%

Table 2: Timing Overhead Comparison

FPGA	Minimum clock period in original IP core	Existing method (previous watermarking techniques)		Proposed method (Zero-knowledge verification protocol)	
		Minimum clock period in watermarked IP core	Timing overhead	Minimum clock period in watermarked IP core	Timing overhead
XC2V250-6cs144	17.123ns	15.210ns	-11.17%	17.123ns	0%
XC2V1000-6bg575	17.019ns	20.297ns	19.26%	17.019ns	0%
XC2V1000-6bg575	16.523ns	15.696ns	-5.01%	16.523ns	0%
XC2V1000-6bg575	16.310ns	16.838ns	4.63%	16.310ns	0%
XC6VCX75t-2ff484	13.578ns	14.106ns	3.24%	13.578ns	0%

Table 3: Experimental data of Watermarking Nova

	Original Nova	Watermarked Nova	Watermarking overhead
Slices	3320	3320	0%
Minimum clock period	13.578ns	13.578ns	0%

4.3 Robustness of Position Permutation

Position permutation is an important metric to evaluate the security of the verification protocol. We can also use the average Manhattan distance (μN(δL)), Manhattan standard deviation (σN(δL)), and the correlation coefficient (ρN(l, δL)) between the LUT position and Manhattan distance to measure the robustness of the position permutation. The larger the value of σN (δL) and the smaller the value of ρN(l,δL) are the higher the robustness of the position permutation. (xk ,yk) and (xk , yk) are the positions of the Kth LUT before and after position permutation, respectively.

In this zero-knowledge protocol watermarking technique the simulation results are obtained by the process of giving the input image and input text first and then it undergoes the encryption process by which the encrypted output image is obtained, finally decryption process takes places and then the decrypted output is obtained.

V. ANALYSIS AND COMPARISON OF WATERMARKING TECHNIQUES

The existing watermarking methods have several drawbacks when compared with the method proposed in this paper. The disadvantages of existing methods includes, low robustness, low accuracy and performance level, high complexity level. The analysis of some watermarking techniques, are as follows.

The publicly detectable watermarking for IP authentication in VLSI design includes a publicly detectable VLSI watermarking technique that embeds an independent public watermark for public verification and the watermark is publicly detected without losing its strength and security. The idea is to create a cryptographically strong pseudo-random watermark embed it into the original problem as a special constraint and make it public. The Fingerprinting techniques, for field-programmable gate array intellectual property protection includes the technique that leverages the unique characteristics of FPGA to protect commercial investment in IP through fingerprinting. The hidden encrypted mark is embedded into the physical layout of a digital circuit when it

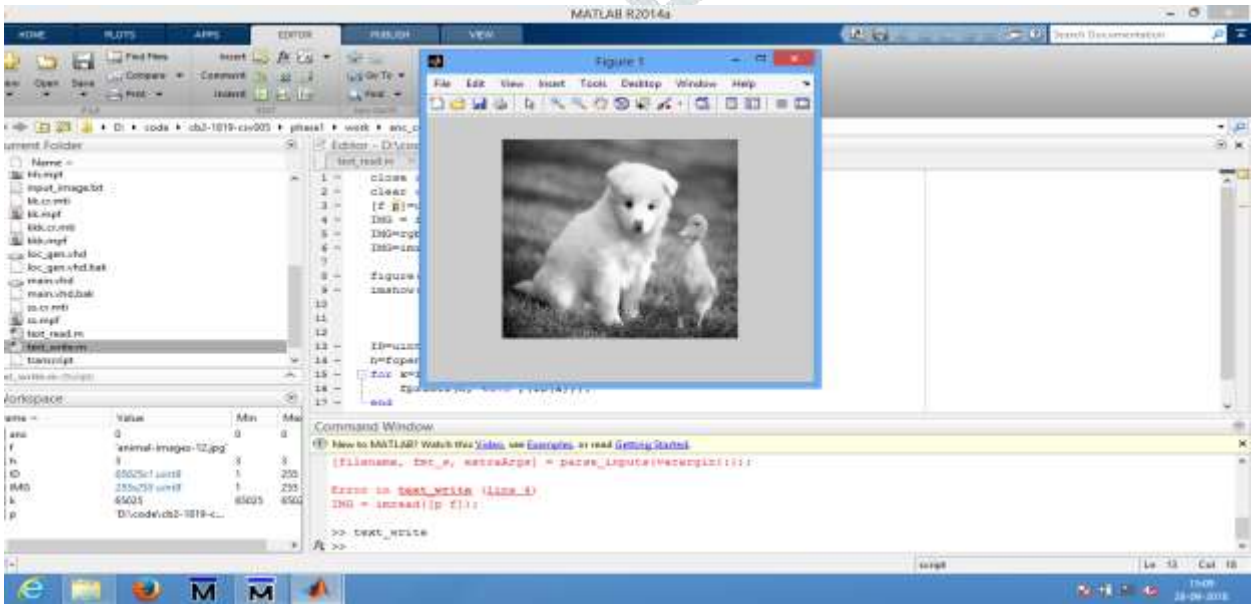
is placed and routed onto the FPGA. This mark uniquely identifies both the circuit origin and the original circuit recipient. The secure public verification of IP marks in FPGA designs includes the zero-knowledge protocol in which it is an interactive two-person game between the prover and the verifier. This protocol satisfies zero-knowledge property and introduces statistical metrics to measure its robustness. The protocol used in this is fast, incurs no additional design overhead and needs no centralized signature database.

The ultra-low overhead dynamic watermarking on scan design for hard IP protection includes ultra-low overhead watermarking scheme to protect hard IPs, the dominating form of commercial IPs. An optimized scan design uses two complementary connections between two adjacent scan cells and such scan design flexibility in the section of local connection styles provides a vehicle to embed watermarking constraints. It can be conveniently implemented by local rewiring or by introducing dummy scan cells. A blind dynamic fingerprinting technique for sequential circuit IP protection includes the first dynamic fingerprinting technique on sequential circuit IPs to enable both the owner and legal buyers of an IP embedded in a chip to be readily identified in the field and the fingerprint in this is an oblivious ownership watermark independently endorsed by each user through a blind signature protocol. Thus the authorship can also be proved through the detection of different users fingerprints without the need to separately embed an identical IP owner's signature in all fingerprinted instances.

Table 4: Comparative Analysis of some Existing Watermarking Techniques

ALGORITHM	ADVANTAGE	DISADVANTAGE
The combine data-integrity techniques are used in which it is compatible and resulting public-private watermark maintains the strength of watermark.	Easy detectability and high credibility.	Low robustness is obtained.
The technique of cryptographically encoded marks to FPGA digital designs is used.	Capable of encoding long messages.	Performance and area impacts are minimal.
The fingerprinting based FPGA digital signature verification scheme is used.	Good robustness and overhead can be achieved.	Vulnerable to embedding attacks.
The ultra-low overhead watermarking scheme is used in-order to protect hard IPs.	Easy detectability.	Low performance and vulnerable to embedding attacks.
The blind signature protocol is used for sequential circuit IP protection.	Applicable to both ASIC and FPGA IPs.	The robustness is low and low accuracy.

VI. SIMULATION RESULTS



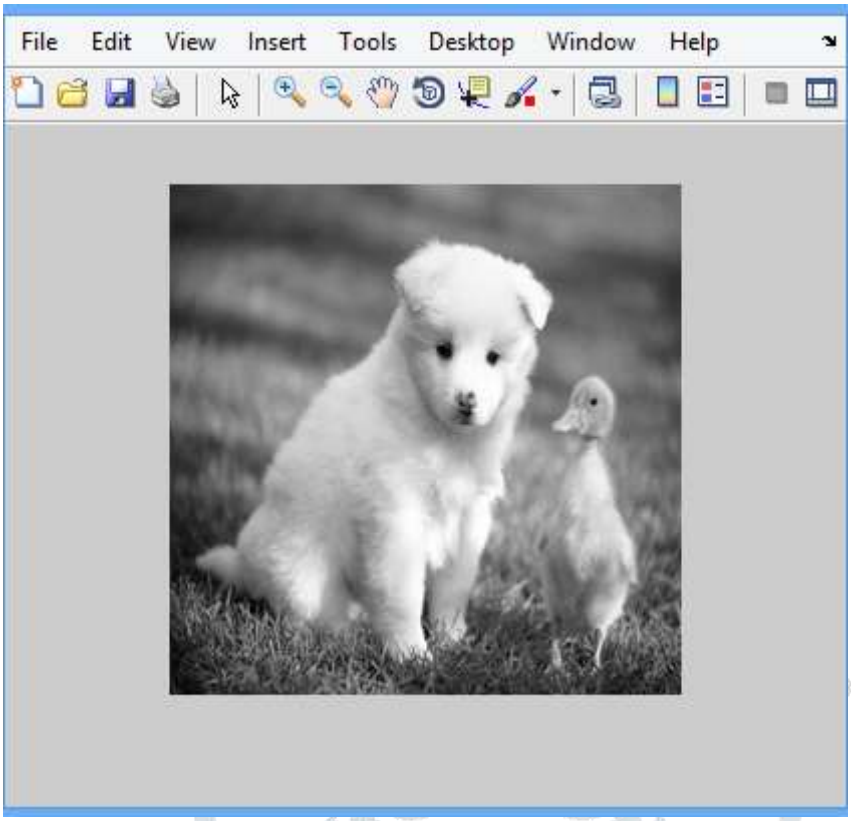
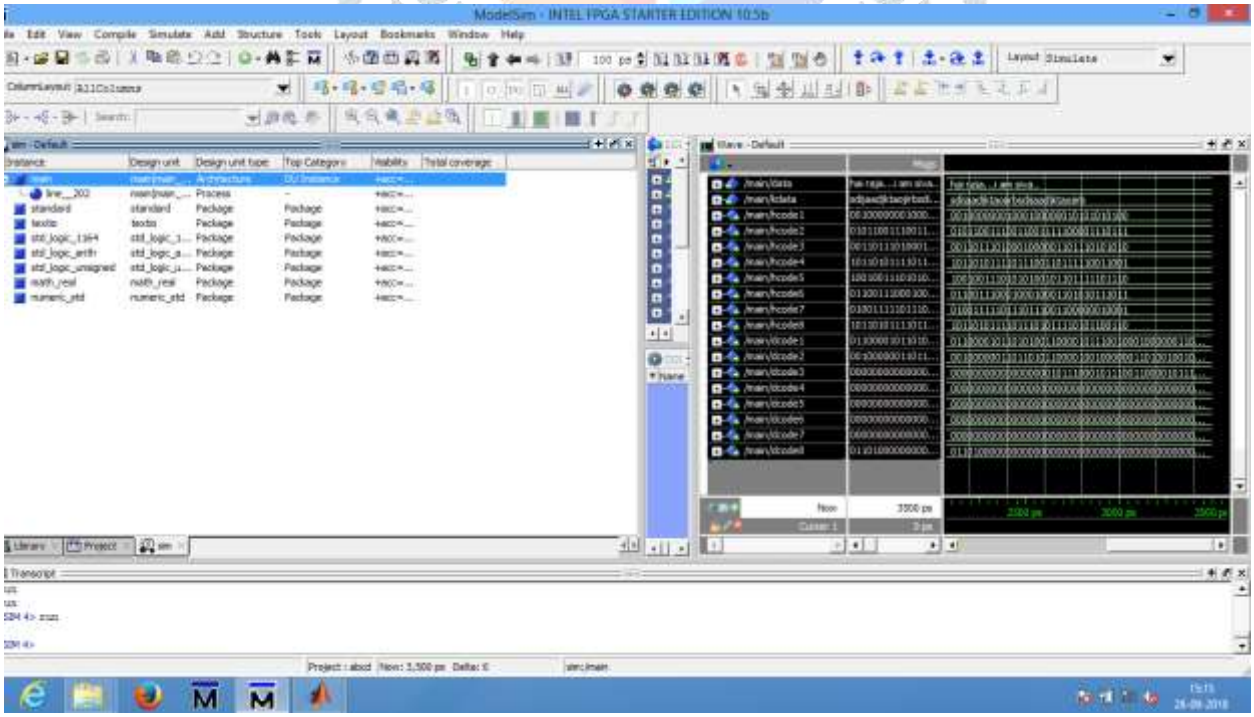


Fig.3 Input-Image



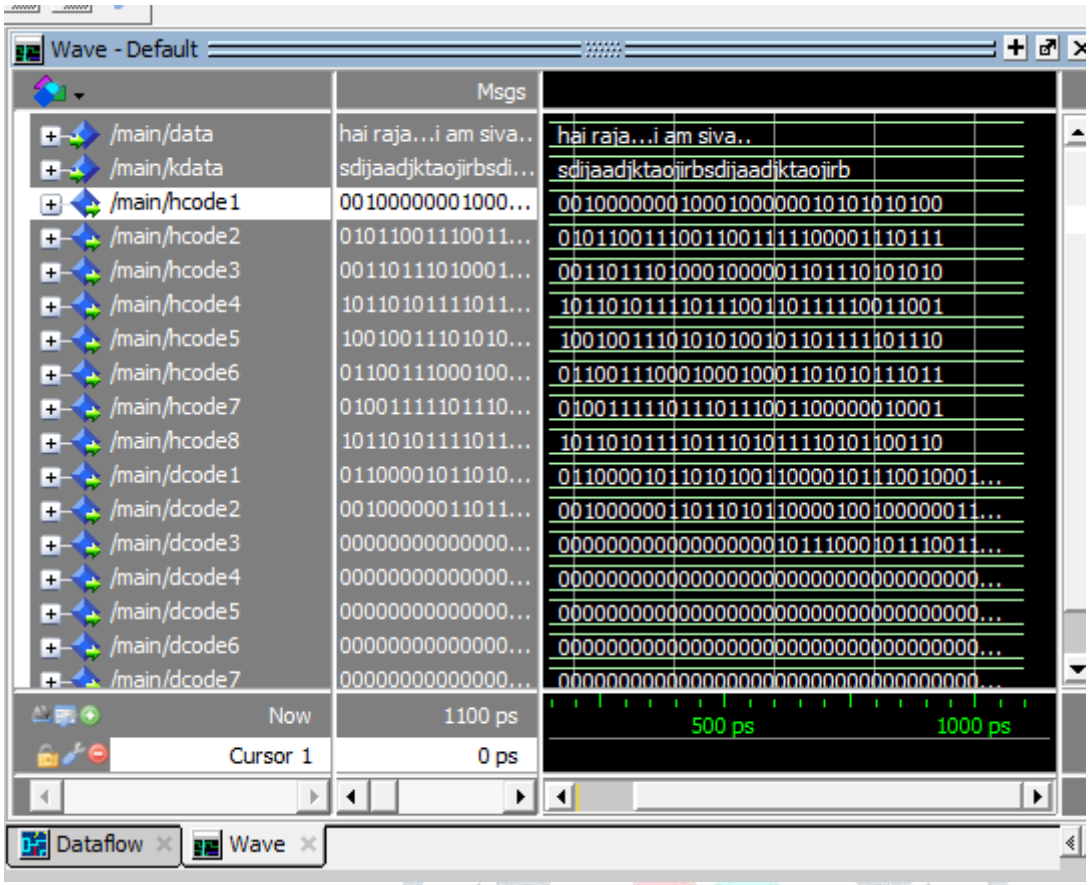
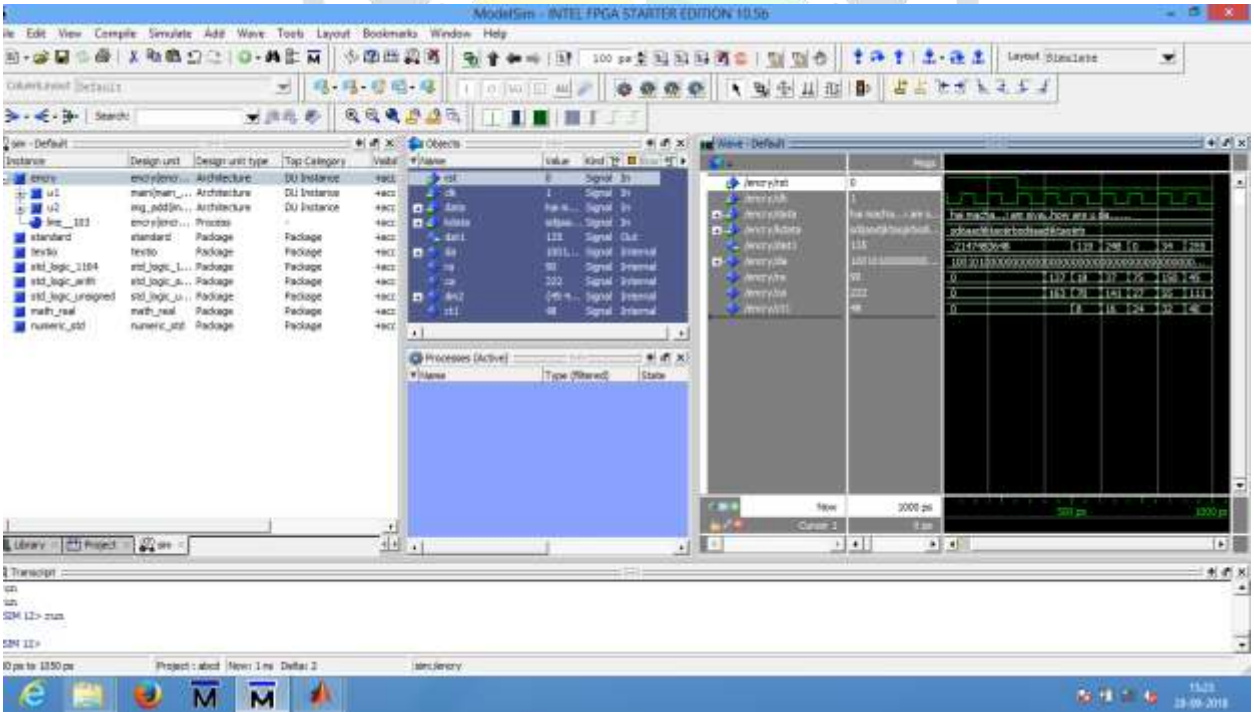


Fig.4 Input-Text



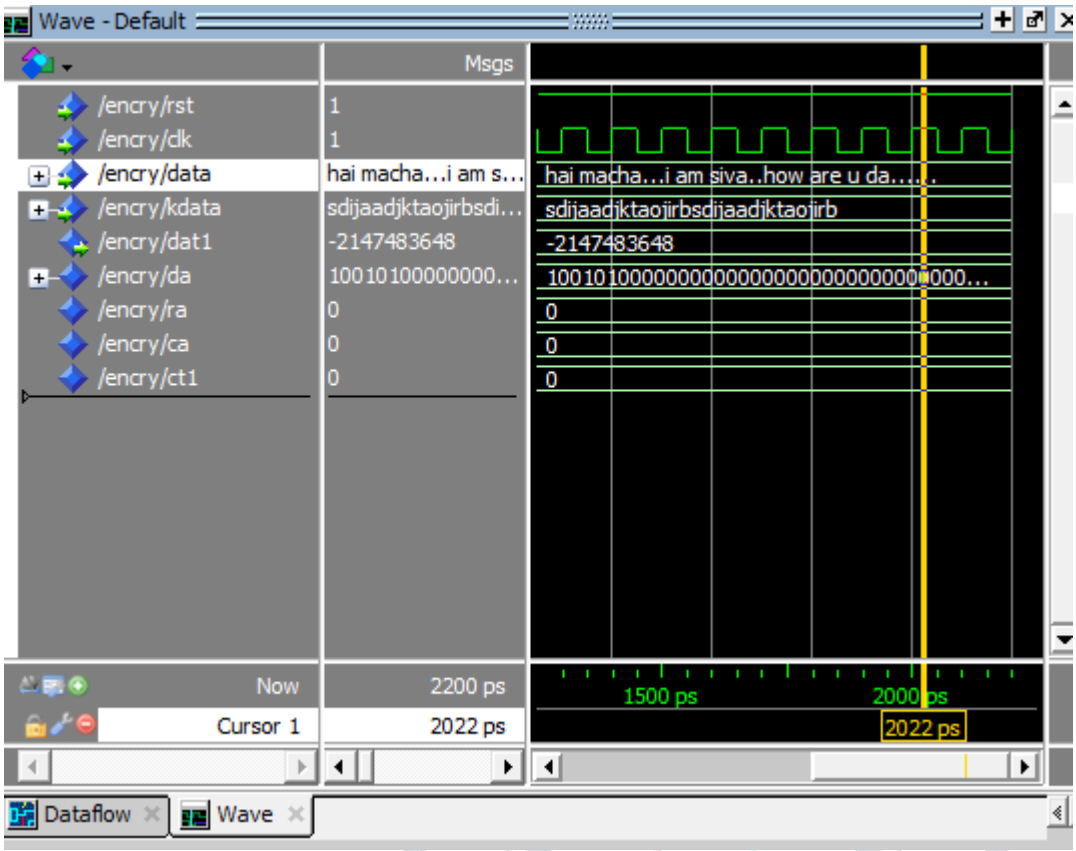
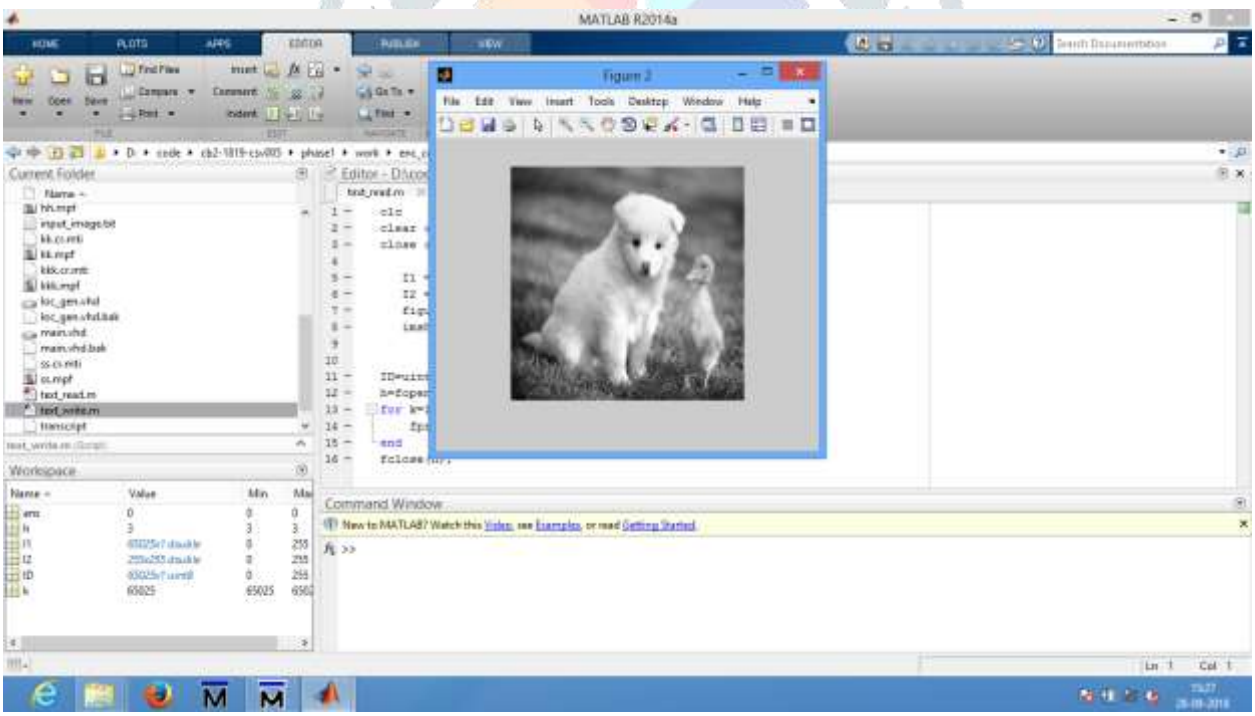


Fig.5 Encryption



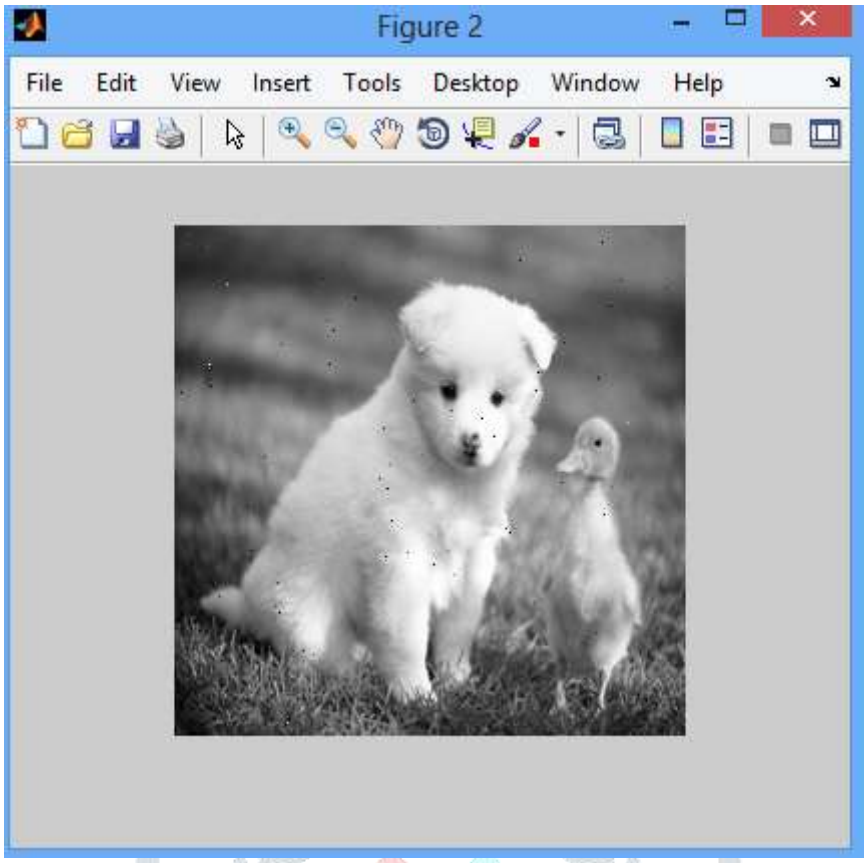
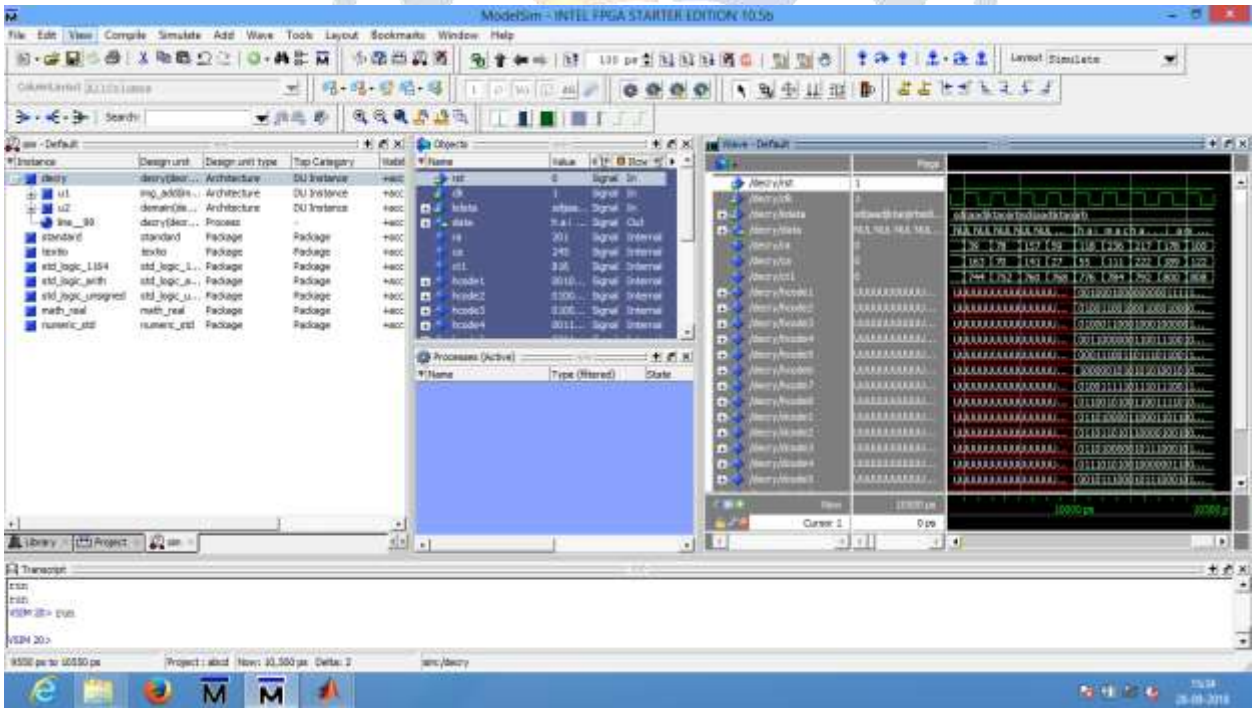


Fig.6 Encrypted Output Image



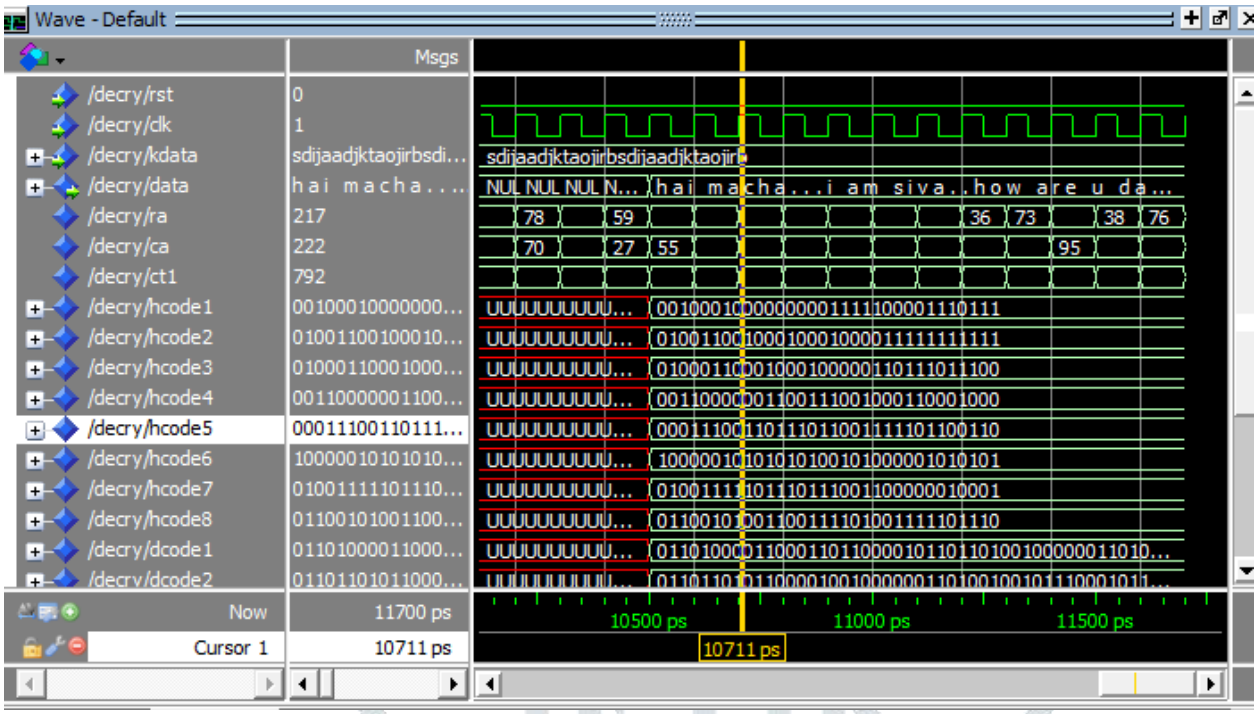


Fig.7 Decryption and Decrypted Output

VII. CONCLUSION

The chaos-based zero knowledge publicly verifiable watermarking detection scheme, proposed in this paper will not give away sensitive information such as the content and the position of embedded watermarks. In addition, the linking or distributed trust time-stamping mechanism is used to address the issue that the existing FPGA watermarking detection schemes are vulnerable to embedding attacks. Since the inherent advantages of the chaotic system exactly meet the special requirements of random position permutation in the zero-knowledge protocol, the proposed scheme has high position permutation robustness. In this proposed method zero percentage resource, timing and watermarking overhead can be achieved. The experimental results also show that the proposed watermarking scheme incurs almost zero overhead and the analysis show that the proposed method has better robustness than the previous watermarking techniques. This method also provides improved accuracy and performance level and it also reduces the complexity level. Thus this paper presents the work on the use of watermarking technique for the IP protection in FPGA design.

VIII. ACKNOWLEDGEMENT

Our sincere thanks to the Management of Info Institute of Engineering for providing the Research Laboratory for our work.

REFERENCES

[1] G. Qu, "Publicly detectable watermarking for intellectual property authentication in VLSI design", IEEE Trans. Comput.-Aided Des.Integr. Circuits Syst., vol. 21, no. 11, pp. 1363–1368, Nov. 2002.

[2] J. Lach, W.H. Mangione-Smith, and M. Potkonjak, "Fingerprinting techniques for field-programmable gate array intellectual property protection", IEEE Trans. Comput.-Aided Des.Integr. Circuits Syst., vol. 20, no. 10, pp. 1253–1261, Oct. 2001.

[3] D. Saha and S. Sur-Kolay, "Secure public verification of IP marks in FPGA design through a zero-knowledge protocol", IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 20, no. 10, pp. 1749–1757, Oct. 2012.

[4] A. Cui, G. Qu, and Y. Zhang, "Ultra-low overhead dynamic watermarking on scan design for hard IP protection", IEEE Trans. Inf. Forensics Security, vol. 10, no. 11, pp. 2298–2313, Nov. 2015.

[5] C.H. Chang, and L. Zhang, "A blind dynamic fingerprinting technique for sequential circuit intellectual property protection", IEEE Trans. Comput.-Aided Des.Integr. Circuits Syst., vol. 33, no. 1, pp. 76–89, Jan. 2014.

[6] J. Lach, W.H. Mangione-Smith, and M. Potkonjak, "Robust FPGA intellectual property protection through multiple small watermarks", in Proc. 36th Annu.ACM/IEEE Design Autom. Conf., Jun. 1999, pp.831-836.

[7] J. Zhang, Y. Lin, Q. Wu, and W. Che, "Watermarking FPGA bit-file for intellectual property protection", Radio-engineering, vol. 21, no. 2, pp.764-771, Jun.2012.

[8] A. Adelsbach and A.R. Sadeghi, "Zero-knowledge watermark detection and proof of ownership", in Proc. 4th Int. Workshop Inf. Hiding, Apr. 2001, pp. 273–288.

[9] W. Liang, K. Wu, Y. Xie, and J. Duan, "TDCM: An IP watermarking algorithm based on two-dimensional chaotic mapping", Comput. Sci. Inf. Syst., vol. 12, no. 2, pp. 823–841, 2015.

[10] Q. Liu, W. Ji, Q. Chen, and T. Mak, "IP protection of mesh NoCs using square spiral routing", IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 24, no. 4, pp. 1560–1573, Apr. 2016.