Edge Based Watermarking Protection for Sequence Detectors

Jasmine Saini* *and* Sagar Chauhan
Department of Electronics and Communication Engineering
Jaypee Institute of Information Technology, Noida, India
*jasmine.saini@jiit.ac.in

Abstract

In recent years, much attention is paid on the protection of Intellectual property due to which the IP owners are forced to incorporate some kind of protection equipments like Signature or Watermark in their designs which assures them that their creations will not be redistributed among others consumers illegally. This paper proposes a new technique for watermarking IP designs in which ownership proof is embedded as part of the IP design. We have used this technique on the 8-bit Sequence Detectors with 8-bit Signatures that helps to detect the piracy. This technique utilizes coinciding edge transitions, un-used edge transitions as well as new edge transitions that were introduced in the state transition graph of the Sequence detector design. The proposed approach increases the robustness of the watermark and allows a secure implementation.

Keywords-FSM, IPP, Sequence Detector, Signature, STG.

Introduction

As there is advancement in Digital media, growth can also be seen in the Digital media piracy problems. These piracy problems can be classified into the following three categories: 1) Illegal Access, where the infringer uses the network site as a medium to obtain a Digital product without permission; 2) Intentional Tampering, where the infringer, in order to extract/insert features for malicious reasons, modifies a digital product and then proceeds to its retransmission. In this case, authenticity of the original product is lost; and 3) Copyright Violation, where the infringer receives a product and resells it without getting the permission to do so from the copyright owner. The same applies to Intellectual Property (IP) blocks used in Digital Electronic Design. The business model is such that the IP core owner gets royalty on every unit sold. Thus, it is extremely important for IP owners to protect their designs from unauthorized use. The best way of achieving this is by embedding a watermark in the design so that it assists the IP owner to prove in a court of law his ownership in case of suspected piracy of the design (Abdel-Hamid et al., 2006). A Sequence detector is a Finite state machine (FSM) which detects a particular Sequence. A Sequence detector is categorized into two types: 1) Overlapping Sequence detector; 2) Non Overlapping Sequence detector.

In this paper we have proposed a technique in which we have created two Overlapping Sequence detector machines which detects 8-bit input sequence and then gives output as '1'.Then we used Edge based watermarking technique to watermark the design. To watermark the Sequence detector, a 8-bit signature was used which was then mapped with randomly taken inputs and then watermarking is carried out by embedding un-used transitions and existing transitions in the STG (State Transition Graph) of original Sequence detector. Thus the resulting watermarked Sequence detector (FSM) has more number of bits in any input than original FSM due to which it did not remained a Sequence detector. Then extra new edges were introduced in to it to make that watermarked FSM into a Sequence detector and to use it as Sequence detector we have taken the MSB(Most Significant Bit) as '0'.

Then the Simulation of VHDL code of above watermarked Sequence detector is carried out and Waveform extraction is done using Altera Model Sim Simulator to analyze it's working.

Watermarking of Sequence Detectors

Sequence detectors used in this paper were created by the authors especially for these watermarking techniques These sequence detectors are non-overlapping sequence detectors that detects 8-bit sequence and gives output as '1'. Watermarking was carried out by two algorithms as mentioned by Amr T. Abdel-Hamid, Sofiene Tahar, and El Mostapha Aboulhamid in "Finite State Machine IP Watermarking: A Tutorial". First algorithm used was Input comparison algorithm, in which sequence detector of sequence "11010011"was used. Second algorithm used was to convert the watermarked FSM which was formed by watermarking of sequence detector using Input comparison algorithm, into a Watermarked sequence detector. Third algorithm was Output mapping algorithm, in which sequence detector of sequence "10100110" was used. Fourth algorithm used was to convert the watermarked FSM, which was formed by watermarking of sequence detector using Output mapping algorithm, into a Watermarked sequence detector.

Watermarking Using Input Comparison Algorithm

Algorithm Steps [1]

1. Firstly select any random state and then compare the inputs of the randomly selected state to the generated signature to check if they are same or not.
2. If input is not used in randomly selected state, an extra transition is added directly to the STG and the next state will be chosen randomly.
3. If input is already used in the selected state, the output of such pair is compared to the output of the transition, to check if it coincides with the generated signature. The transition will be then considered as part of the added signature, and the algorithm will advance to the next state that already exists in the STG.
4. If all the outputs of the given state are already used, then an extra input bit is added to the system. This input bit will have the same logic value for already existing transitions. Then the logic bit '0' will be added to LSB of all existing edges and the logic bit '1' will be added to the LSB of watermark transition added. The next state will be chosen randomly.
5. The process will continue to go on until all the Signature Transitions are inserted.

The Original Sequence Detector used for this type of algorithm is "11010011"

Application of algorithm on Original Sequence Detector used

1. Signature bits taken for watermarking were **"10010010".**These signature bits were used for generation of Signature Edges which were further used for watermarking the Sequence Detector.
2. As Input bits were taken randomly, so the input Sequence, on which Sequence Detector gave Output as '1', was taken as "11010011".
3. Then this Input Sequence bits were mapped with the Signature bits to generate Signature edges. The Mapping [inp/Sig] gaveDesired Signature as:-

   [1/1], [1/0], [0/0], [1/1], [0/0], [0/0], [1/1], [1/0].
4. Then these Desired Signature Edges were applied to Original Sequence Detector.
5. The result of the application of Desired Signature Edges is in the form of Watermarked FSM as shown in Figure2.
6. Then according to this algorithm, this Desired Signature was then transformed into Actual

Signature which is as shown:-

[11/1], [1/0], [0/0], [11/1], [0/0], [0/0], [11/1], [11/0].

---------->

This dotted arrow is used indicate the **Introduced Edge** which the system uses during watermarking procedure.

This Darkened arrow is used to indicate the **Existing Edge** which the system uses during Watermarking procedure.

Problem faced after application of this algorithm

Since the application of the algorithm involves changes in the Transition edges i.e, there were some Transition edges with input having two bits and some with input having one bit. Also some of the transition edges of Original Sequence Detector which were used during detection of input sequence got changed after the application of the algorithm due to which the Watermarked FSM no longer behaved as Sequence Detector. So,there was need to convert it again into a Sequence Detector.

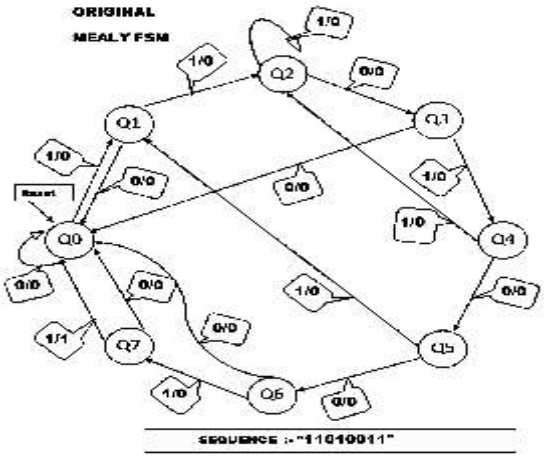

Figure1: Original Sequence Detector used in the algorithm

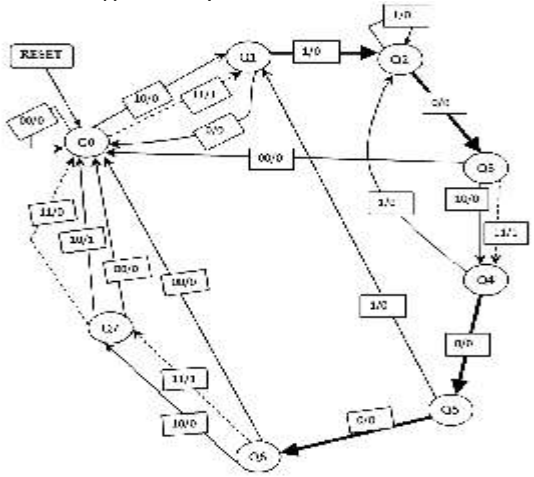

Figure 2: Edge Based Watermarked FSM

Problem faced after application of this algorithm

Since the application of the algorithm involves changes in the Transition edges i.e, there were some Transition edges with input having two bits and some with input having one bit. Also some of the transition edges of Original Sequence Detector which were used during detection of input sequence got changed after the application of the algorithm due to which the Watermarked FSM no longer behaved as Sequence Detector. So,there was need to convert it again into a Sequence Detector.

Conversion of Watermarked FSM Due To Input Comparison Algorithm Into A Watermarked Sequence Detector

Algorithm Steps

1. Firstly, all the Transition Edges in the WatermarkedFSMwhich are having input of one bit are converted into two bits input by applying **'0'** at the **MSB** due to which every state has a possibility of having **four** Input Transitions.
2. To convert the Watermarked FSM into a Sequence Detector, the **MSB** of each input is taken as **'0'**and the whole concentration is given only on the **LSB** part of the input Sequence. That is to detect the input Sequence **"11010011"** asthereintheOriginal Sequence Detector , the input Sequence of the Watermarked FSM is **"01 01 00 01 00 00 01 01"**.
3. Now, all the input Transitions are checked sequentially and if they are already present then they are used as it is otherwise the required Transitions are added externally.
4. Externally added Transitions are denoted by the arrow as " ⟶ ".
5. Due to the above the Signature Edges are also transformed thus results in the formation of **Transformed Signature** which is

   [11/1] , [01/0] , [00/0] , [11/1] , [00/0] , [00/0] , [11/1] , [11/0]

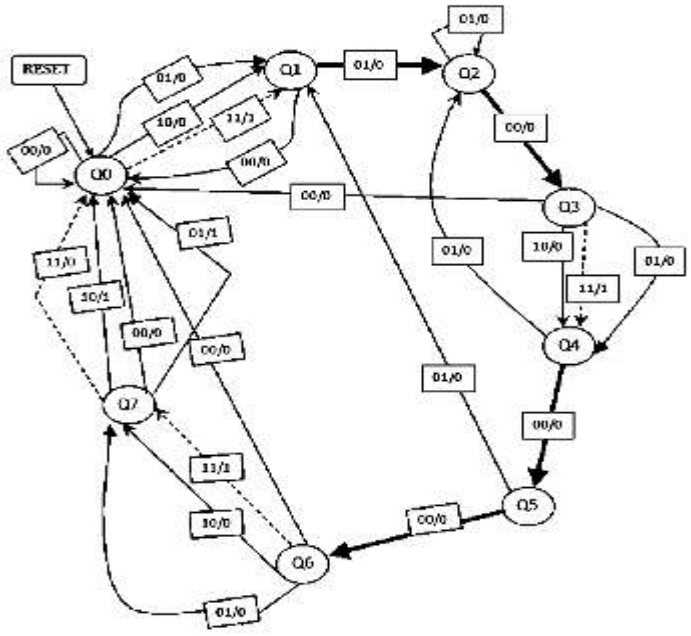

Figure 3: Edge Based Watermarked Sequence Detector

Watermarking Using Output Mapping Algorithm

This watermark insertion algorithm coincides a part of the watermark on the FSM transitions to increase the watermark robustness. Starting from any randomly chosen state ,the watermark will be added to the FSM according to the following steps[1]: -

Algorithm Steps [1]

1. Firstly select any random state and thencompare the outputs of the randomly selected state to the generated signature to check if they are same or not.
2. In case if the outputs of above *randomly selected state and the generated* signature is same then that transition will be considered as a part of Signature Transition.
3. Ifany of the outputs of above *randomly selected state and the generated* signature is not same, then the inputs of *randomly selected state will be checked to* determine if there is any free input that can be used to add an extra transition. In such a case the next state will be chosen randomly, with preference given to states with free transitions.
4. If all the inputs of the given state are already used, then an extra input bit *is added to the system*. This input bit will have the same logic value for already existing transitions. Then the logic bit '0' will be added to LSB of all existing edges and the logic bit '1' will be added to the LSB of watermark transition added. The next state will be chosen randomly.
5. The process will continue to go on until all the Signature Transitions will be inserted.

The Original Sequence Detector used for this type of algorithm is "10100110"

Ultrasonic sensing technologies based the principle that sound has a relatively constant velocity. The time for an ultrasonic sensor's beam to strike the target and return is directly proportional to the distance to the object. The sensing range of an ultrasonic sensor is the area between the minimum and the maximum sensing limits.

Application of algorithm on Original Sequence Detector used

1. Signature bits taken for generation of Signature Edges for watermarking the Sequence Detector was "**10100110**".
2. As Input bits are to be taken randomly so the input Sequence taken was "**10000100**".
3. Then this Input Sequence bits were mapped with the Signature bits to generate Signature edges. The Mapping [input/Signature] gave Desired **Signature** as **:-**

   [1/1] , [0/0] , [0/1] , [0/0] , [0/0] , [1/1] , [0/1] , [0/0]
4. Then these Desired Signature Edges were applied to Original Sequence Detector.
5. The result of this application of Desired Signature Edges was in the form of Watermarked FSM as shown in **Figure 5**

Then according to this algorithm this Desired Signature was then transformed into **Actual Signature** which is as shown:-

[11/1] , [0/0] , [11/1] , [0/0] , [0/0] , [11/1] , [11/1] , [1/0]

The Result of Application of the Algorithm in the form of Watermarked FSM

In this Watermarked FSM ,

This dotted arrow is used indicate the **Introduced Edge** which the system uses during watermarking procedure.**Figure.5 Edge Based Watermarked (Output Mapping) FSM**

This Darkened arrow is used to indicate the **Existing Edge** which the system uses during Watermarking procedure.

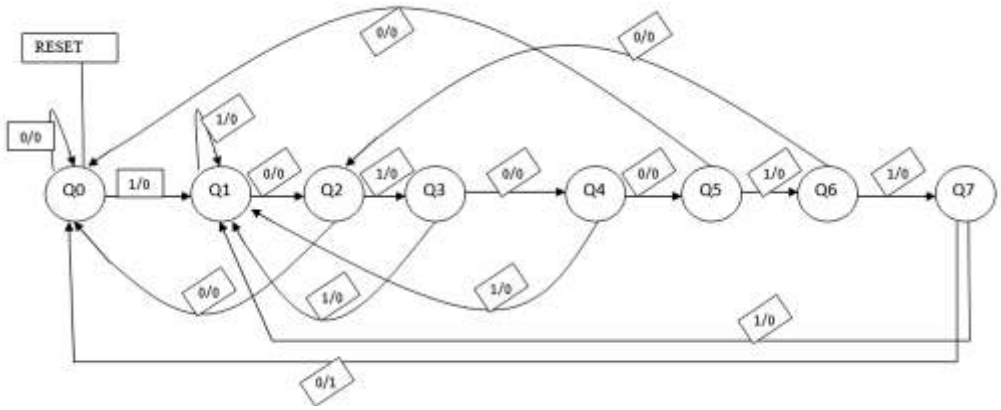

Figure 4: Original Sequence Detector used in the algorithm

Problem faced after application of this algorithm

Since the application of the algorithm involves changes in the Transition edges that is there were some Transition edges with input having two bits and some with input having one bit. Also some of the transition edges of Original Sequence Detector which were used during detection of input sequence got changed after the application of the algorithm due to which the Watermarked FSM no longer behaved as

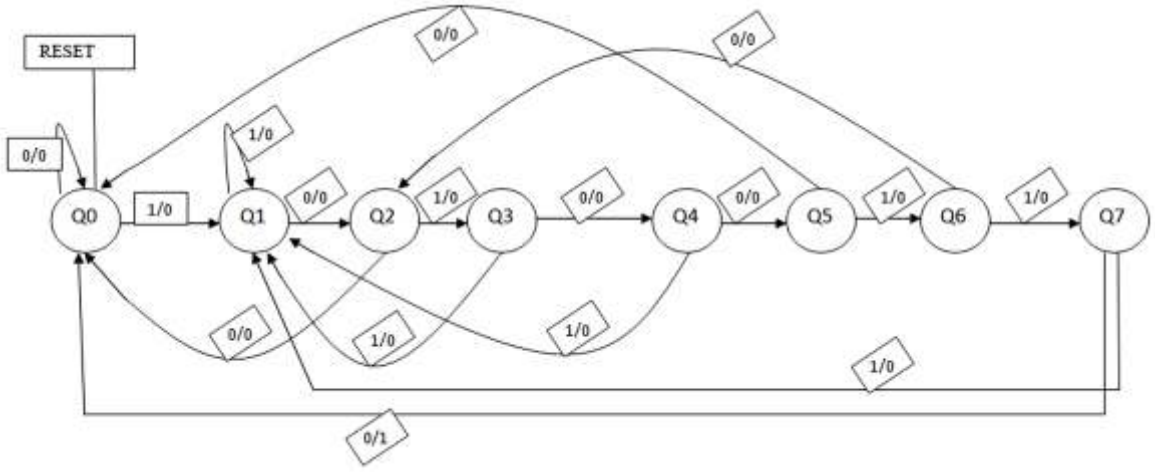

Figure 4: Original Sequence Detector used in the algorithm

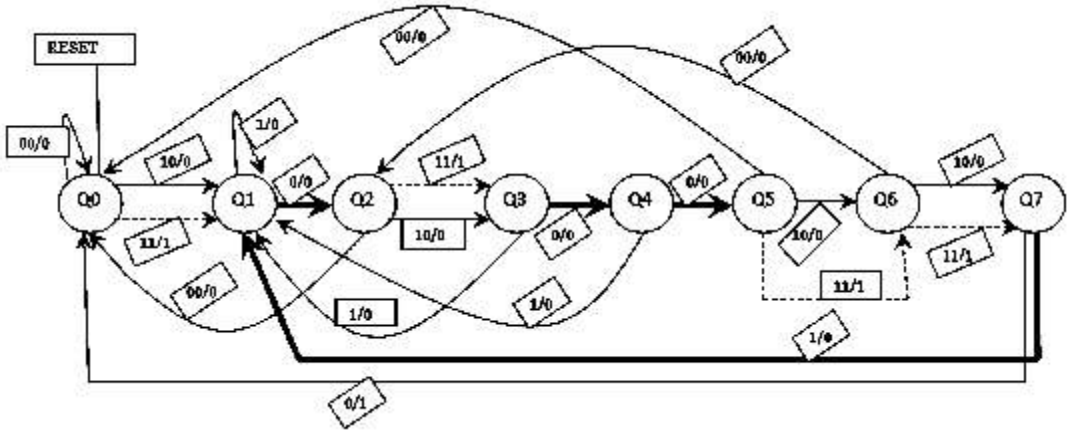

Figure 5 : Edge Based Watermarked (Output Mapping) FSM

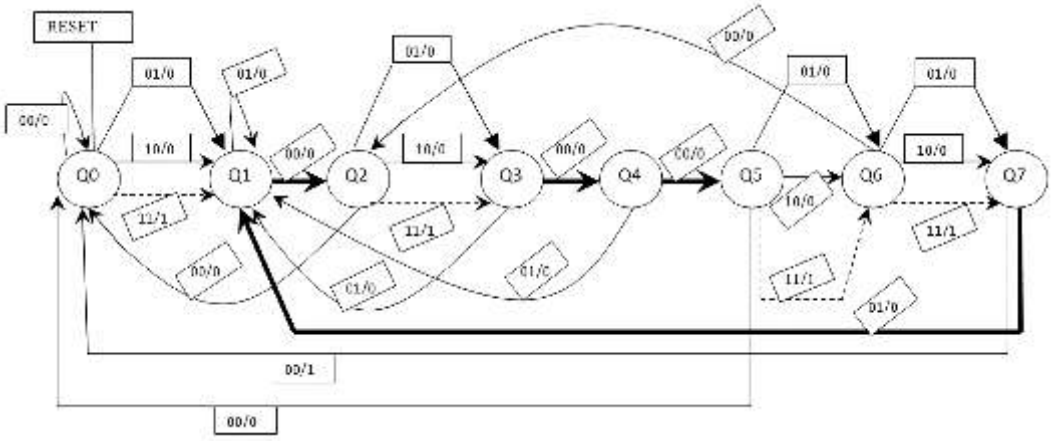

Figure 6: Edge Watermarked (Output Mapping) Sequence Detector

Results

Waveforms resulting from Input Comparison Algorithm

Waveform of Sequence Detection

The Waveform resulting from ModelSim Simulator, when two bit input sequence of "01 01 00 01 00 00 01 01 is serially inserted into the watermarked sequence detector from MSB side to LSB side. This input sequence gives output as '1'. States and output is updated at every rising clock edge of the clock signal as shown in Figure7.

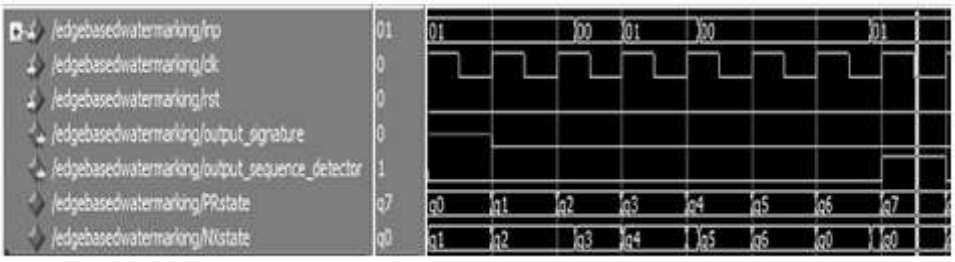

Figure 7: Waveform of Sequence Detection of Input comparison

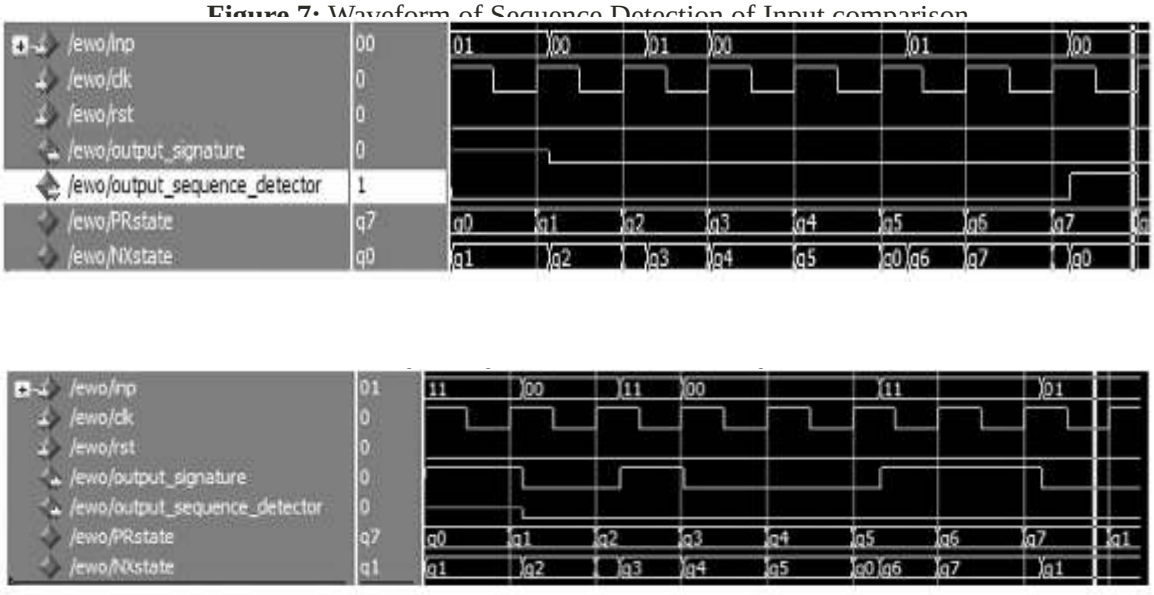

Figure10: Waveform of Signature Detection of Output mapping

Waveform of Signature Detection

The Waveform resulting from ModelSim Simulator, when input signature of "11 01 00 11 00 00 11 11" is serially inserted into the watermarked sequence detector and signature is observed as "10010010"as shown in **Figure 8**.

Waveforms resulting from *Output Mapping Algorithm*

Waveform of Sequence Detection

The Waveform resulting from ModelSim Simulator, when two bit input sequence of "01 00 01 00 00 01 01 00" is serially inserted into the watermarked sequence detector from MSB side to LSB side. This input sequence gives output as '1'.States and output is updated at every rising

clock edge of the clock signal as shown in **Figure 9**.

Waveform of Signature Detection

The Waveform resulting from ModelSim Simulator, when input signature of "11 00 11 00 00 11 11 01" is serially inserted into the watermarked sequence detector and signature is observed as "10100110"as shown in **Figure 10.**

Conclusion

First the original sequence detectors of one bit input are watermarked due to which their input bits changes in number and becomes two bit input watermarked sequence detectors. Simulation of above watermarked sequence detectors is carried out using ModelSim simulator. The results are in the form of waveform which shows that these sequence detectors behave as original sequence detectors when the MSB is considered as '0' throughout the sequence detection and only the LSB part of input is seen as input sequence at which the original sequence detector gives as output '1'. Also for signature detection, the particular input sequence as mentioned above will give the output sequence in the form of signature for

the corresponding sequence detectors which is only known to the designer of sequence detector and due to which he can prove his ownership.

References

Abdel-Hamid, A. T., Tahar, S., & Aboulhamid, E. M. 2006. Finite state machine IP watermarking: A tutorial. In *Adaptive Hardware and Systems, 2006. AHS 2006. First NASA/ESA Conference on* (pp. 457-464). IEEE.

Abdel-Hamid, A. T., Tahar, S., Aboulhamid, E. M. 2004. A tool for automatic watermarking of IP designs. In *Circuits and Systems, 2004. NEWCAS 2004. The 2nd Annual IEEE Northeast Workshop on* (pp. 381-384). IEEE.

Abdel-Hamid, A. T., Tahar, S., Aboulhamid, E. M. 2005. A public-key watermarking technique for IP designs. In *Proceedings of the conference on Design, Automation and Test in Europe-Volume 1* (pp. 330-335). IEEE Computer Society.

Bhasker, J., Bhasker, J. 1999. *A Vhdl primer*. Prentice Hall PTR.