Hindawi
International Journal of Reconfigurable Computing
Volume 2023, Article ID 4550758, 13 pages
https://doi.org/10.1155/2023/4550758

Research Article

Hardware Obfuscation Based Watermarking Technique for IPR Ownership Identification

Priyanka Bagul[1] **and Vandana Inamdar**[2]

[1]*Department of Electronics & Telecommunication, M. E. S. College of Engineering, Pune, Maharashtra, India*
[2]*Department of Computer Engineering, Government College of Engineering and Research Center, Awasari Khurd, Ambegaon, Maharashtra, India*

Correspondence should be addressed to Priyanka Bagul; priyanka.awasare@mescoepune.org

Received 3 February 2023; Revised 4 June 2023; Accepted 6 June 2023; Published 3 July 2023

Academic Editor: Vamseekrishna Allam

As the reuse of IP cores or the development of frequently used hardware modules is gaining more attention in the semiconductor industry, the misappropriation of the owner's identity is a rising concern. Therefore, imprinting the owner's identity in the form of a watermark or signature on the IP core is essential to avoid intellectual property right (IPR) infringement. In view of this, a watermarking technique is proposed in the present manuscript. A constraint-based dynamic watermarking method to generate the owner's signature is proposed in conjunction with the logic encryption-based hardware obfuscation method. The method formulated in this manuscript consciously makes use of a basic switching component for embedding a watermark with IP core and hardware obfuscation, to achieve a lower overhead budget. Through the switching mechanism, the embedded watermark can be made detectable to legitimate end users off chip via test pin. The logic encryption-based method is set for accessing the watermark. Furthermore, an encrypted functionality is set as the signature generator module for generating owner's signature. This provides hardware obfuscation and two-stage authentication mechanism for the generation of owner's signature, and as a result of this, double-layer protection is achieved. Furthermore, a novel method to configure input key for signature generation module and to formulate owner's signature is proposed. The viability of the present watermark technique for real-life application is checked on the ground of transparency, security, reliability, performance overhead, and robustness. Since the watermark in the proposed method is embedded outside the IP core, it does not cause any latency for the IP core functionality. Thus, even with significantly lower area overhead (~<1.4%), the proposed method is able to provide higher robustness in terms of lower probability of coincidence ($P_C = 4.68\ e-97$).

1. Introduction

Development of intellectual property (IP) cores and its reuse is very common in the very large-scale integration (VLSI) field. These IP cores are often categorized as a soft IP core, hard IP core, and firmware IP core. The soft IP cores are formulated using high-level description language and offered as synthesizable register transistor level (RTL) in hardware description language (HDL) such as system verilog or system C and VHDL, whereas hard IP cores are analog or digital which are realized on silicon real estate, and this is the bottommost level of abstraction in IP cores. Firmware IPs are in the netlist format [1, 2].

Reuse of IP cores is a usual practice to accelerate the design of system-on-chip (SoC) products [3]. This is becoming a standard practice because it makes assembly of complex system easier by allowing integration of smaller components and thereby reducing system built-up complexity through enabling resource optimization [4, 5]. Thus, it also reduces development time and cost. Therefore, design and development communities are looking for the best possible gathering of IP cores of basic and frequently used modules to form libraries.

However, this also brings in the potential risk related to various design security issues such as copying the design illegally, reverse engineering the design, and its overuses

[6, 7]. To overcome these threats, ownership labeling on these IP cores has emerged as a possible solution [8]. Therefore, to reduce IP core infringement while building VLSI and other application-specific integrated circuits (ASIC), several IP protection mechanisms have been proposed. Despite of numerous challenges, IP protection techniques can potentially reduce the ownership misappropriation. In view of this, the electronic design automation industry has established several protecting methods including patents, copyrights, mask works, or trade secrets [9]. However, for intellectual property protection of reusable cores, those methods proved to be insufficient and/or inapplicable [3]. Recently, the watermarking methodology has been put forward to prove its effective candidature for claiming the IP core ownership [10, 11]. Watermarking is the method in which the asset is marked with some known signature structure and due to which the protection against theft or overuse can be provided.

Several different approaches to create a watermark are introduced which are based on the implementation of some algorithmic constraints to incorporate the owner's signature at various stages of logic or physical synthesis, without tampering the original functionality of the IP designs. High-level synthesis (HLS) is one of the important stages of IP core designing. In the present paper, a novel method is presented to embed a watermark at the register allocation stage during the conversion of HLS to GDS II format. However, choosing the method for embedding the watermark is a complex and nontrivial task, and therefore, a number of competitive design solutions are offered by many researchers in the design space [12]. This is due to the well-known fact that each watermark integration method exhibits specific latency and area overhead, and therefore, selecting an appropriate method for embedding the watermark plays a crucial role. A variety of high-level synthesis-based techniques for IP protection is well described in the literature [13–18]. For reusable IP cores, the techniques based on single-phase, triple-phase watermarking, digital signature-based watermarking, binary encoding-based watermarking, and in-synthesis-based watermarking are described. A group of researchers used the encrypted-hashing method to embed digital signature for IP core protection and achieved stronger robustness [19]. Some researchers suggested the implementation of hardware security constraints with the help of multilevel encryption and steganographic constraints using a high-level synthesis framework [20]. A method to achieve a strong ability of obfuscation is introduced by researchers wherein multikey-based structural obfuscation is integrated with tamper-tolerant physical level watermarking [21]. A group of researchers proposed a key-based multiplexer design and incorporated structural and functional obfuscation in the DSP circuit to prevent reverse engineering [22]. A method of quadruple-phase watermarking is introduced to secure hardware IP cores. The use of graph partitioning, encoding tree, and eightfold mapping is demonstrated to accomplish high tamper tolerance [23].

In the present paper, logic encryption-based hardware obfuscation technique is used for embedding watermark. Similarly, some basic switching component is used deliberately for hardware obfuscation and thereby embedding watermark with the IP core to achieve a lower overhead budget. Moreover, the main aim of embedding the owner's identity indelibly into the primitives of the design is to discourage IP theft. Therefore, one must be able to demonstrate the ownership in the court in case of theft. The present work utilizes a dynamic watermarking technique. In this method, typically, the owner's signature is encrypted and embedded as some set of constraints that can be retrieved by running the protected IP with some specific input sequences [24]. In the present paper, a novel method to configure the input key for the signature generation module and to formulate the owner's signature is proposed.

However, at this stage, it is equally important to check its viability of being useful in actual applications, for which the developed identification module is tested on the ground of a few more essential characteristics such as transparency, security, reliability, and performance overhead. Integration of all these key points is considered while designing the present signature generation module (SGM), which is capable of providing fairly good protection to IP cores and simultaneously can fulfil all the characteristics requirements of the practically implementable watermark.

2. Motivation

From the authorship verification viewpoint, watermarking methods are categorised as static watermarking and dynamic watermarking [25]. In the case of the static watermarking method, retrieval of the watermark requires reverse engineering up to the embedding point, which makes this method expensive and intrusive. On the other hand, the dynamic watermarking method provides the ace of watermark verification at the output level by running the protected design with a specific code sequence. For the implementation of dynamic watermarking, numerous methods are demonstrated by researchers. Typically, dynamic watermarking is implemented in the state transition graph (STG) of finite state machine (FSM) [26–29], in the architectural level of digital signal processors [30, 31], or at the design-for-testability stage [32–34]. However, it is important to note here that later developments show the evolution of many watermarking techniques which can be implemented at various design levels such as system design, behaviour design, logic design, and physical design [27, 31, 32, 34–43]. Amongst them, few researchers also reported the use of two different approaches in conjunction with each other [14, 17, 18, 44–53].

Various methods are invented for hardware watermarking. The hardware IP watermarking usually includes embedding and concealing of the owner's signature in the description of a circuit whereas cryptography-based hardware IP protection methods are often incorporated as a part of the design flow of field programmable gate array (FPGA). However, in the case of hardware obfuscation techniques, modifications in the description or the structure of electronic hardware are intentionally made in order to conceal its functionality, due to which comprehending the actual functionality of a design becomes a more complex task for the adversary.

Hardware obfuscation techniques are classified mainly as the passive or active techniques. The hardware obfuscation techniques are classified as passive when the techniques are implemented by modifying the description of the circuit in the soft form. This makes understanding the circuit functionality difficult. It is implemented by string substitution or changing circuit description on HDL level.

In contrast to this, the hardware obfuscation techniques are classified as active when logic-based circuitry is added for obfuscation of the design at the access point of IP core. Frequently, the normal functionality of the obfuscated IP core is enabled only upon the successful insertion of the single predetermined key at the input, which is often set as a combination of some sequence and acts as a secret key. Failure of correct key insertion leads to exhibit incorrect functionality or locking of the IP core. This particular approach is referred as FSM-based hardware obfuscation techniques [17].

One of the most noticed approaches involves the integration of FSM at gate-level design for authentication [49]. This FSM is designed to authenticate a series of input patterns, and for each input, a specific transition state is assigned. This mechanism is set to unlock the IP core functionality, and failure of this authentication leads to faulty output generation. This faulty output then triggers the gate-level design to obfuscate the functionality of the locked chip.

However, implementation of these techniques brings in the high overhead in terms of area, power, and delay. Therefore, in such a scenario, performance trade-off is very common. This makes the choice of the watermarking method even more nontrivial since every watermarking technique impacts the latency and area in a different way, and therefore, selecting a low-cost solution for embedding watermark becomes a more challenging issue. After considering all these important aspects, herein, a watermark implementation method is designed and presented [54]. Instead of using any resources of IP core, it is isolated to keep IP core functionality unaltered, and outside resources are utilized. To achieve this, a dynamic watermarking method to generate the owner's signature is proposed in conjunction with the logic encryption-based hardware obfuscation method.

Considering the fact that multiplexer (MUX) and demultiplexer (DEMUX) exhibit minimal hardware overhead compare to other switching devices, it is used in the present method for hardware obfuscation-based integration of watermark with IP core. As a usual hardware obfuscation practice, extra gates used for logic encryption are controlled using predetermined keys, which, upon insertion of correct key input, enables the IP core and allows networks to produce correct output through it. On contrary to this, the present method utilizes the extra gates to enable a constraint-based watermarking circuit that produces the owner's signature. Herein, for accessing the watermarking circuit, the correct predetermined enable key is needed as an input at this logic encrypted gate while no such key is needed to activate the original functionality of the IP core, and it runs as a default state of the watermarked IP core. Thus, through a switching mechanism, the embedded watermark can be made functional and accessible to legitimate end users off chip via test pin. Furthermore, a novel method to configure the input key for the signature generation module and to formulate the owner's signature is proposed, which includes the use of clock ticks in combination with data bit sequence. The main distinguishable attempts made in the present manuscripts are as follows:

(1) Instead of using any resources of IP core, it is isolated, and outside resources are utilized to keep IP core functionality unaltered

(2) Logic encryption-based hardware obfuscation method is employed for selective running of either IP core or SGM (watermark)

(3) A unique two-stage watermark verification system is designed as SGM to generate owner's signature

(4) The method selected for embedding the watermarked solution with IP exerts minimal hardware overhead in terms of switching devices (multiplexer)

(5) The present SGM is designed in such a way that it can be directly integrated with IP core and recognized as a mark or identity of owner

(6) Bit sequence and clock tick delays are combined to formulate an owner's signature

2.1. Threat Model. The proposed watermarking method intends to secure the underlying IP core and also the signature generator module to preserve the ownership claim. It also targets the possibility of unauthorized signature implanting attack for false claiming and authorized signature removal. Similarly, the logic encryption-based hardware obfuscation method is used for embedding the watermarks, due to which reverse engineering and identifying correct functionality becomes difficult. Two-stage verification is set for generating correct signature of the owners, due to which guessing the correct key at both stages is difficult, and this enhances the layer of protection.

3. Proposed Methodology

In this section, the proposed methodology is explained. At first, a brief overview of the proposed approach is described followed by which detailed information of each stage is explained.

Figure 1 represents the overview of the proposed methodology. The method is established as a two-stage mechanism to generate ownership signature or claim, which is capable of giving double layer protection. The first stage is hardware obfuscation which is designed using multiplexers and demultiplexer, and similarly, a secret-key-based activation mechanism is set at this stage. In order to access the second stage, it is necessary to follow the correct access process during the first stage and fed the correct key. The second stage is SGM, which is designed to validate given secret owner's key and generate owner's signature. It is mainly consisting of signature check block, time scale

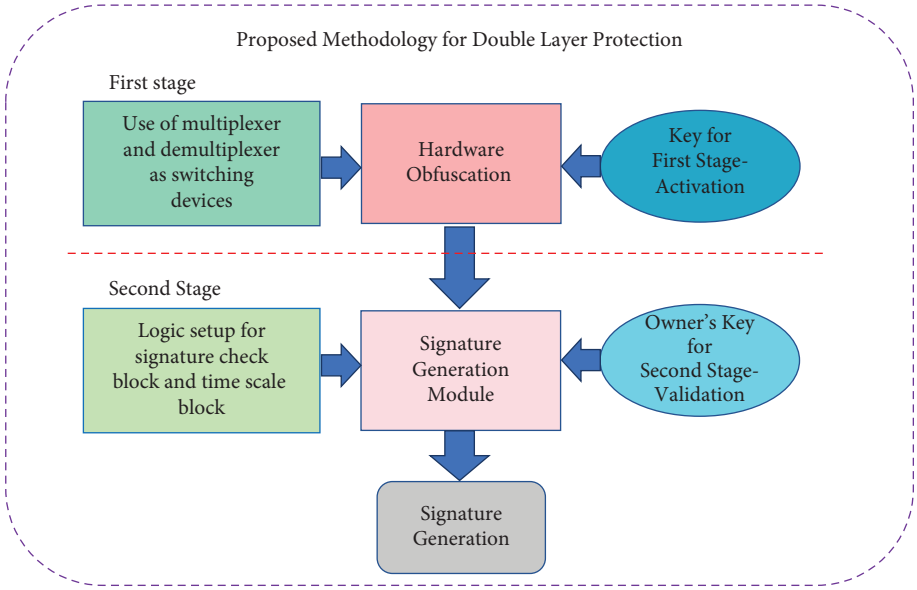

Figure 1: Overview of the proposed methodology.

verification block, and signature generation block, and the logic is set to identify correct owner's key and generate authenticate owner's signature through this stage. An important part of this proposed methodology is design of the key and owner's key, one of which is a combination of data streams and clock tick delays. After successful completion of two stages namely activation and validation, the owner's signature is generated which upholds ownership claim. Details of the proposed methodology are as follows.

3.1. System Architecture. In general, IP cores are connected to basic inputs and outputs (I/O's) such as input signal data bus, output signal data bus, and system clock of SOC/integrated circuits. The proposed watermark utilizes these basic connected I/Os of IP core. Through the logic encryption-based hardware obfuscation method, the watermark is featured as if it encapsulates the IP Core. Therefore, it does not interfere with IP Core's inner circuits. Figure 2 reveals the logic encryption-based hardware obfuscation method, wherein the watermarked IP core is comprised of SGM enable logic, MUX and DEMUX, and SGM, along with the two input buses and one output bus. In order to obfuscate the circuit, the SGM (watermark) is integrated in such a way that it encapsulated the IP core, for which the original IP core input and output bus is altered and regulated via select line of DEMUX and MUX, respectively. As can be seen in Figure 2, the watermarked IP core shows two inputs, one of which accepts key for enable logic and other accepts the input to be given selectively either to IP core or to SGM. For logic encryption, an enable logic block is integrated with select line of DEMUX and MUX to manipulate the input and output of the watermarked IP core. This allows selective communication of input either to IP core as system input signal or to SGM as owner's key via DEMUX and also output signal to be selectively chosen from IP Core or SGM via MUX. Thus, the access of input to IP

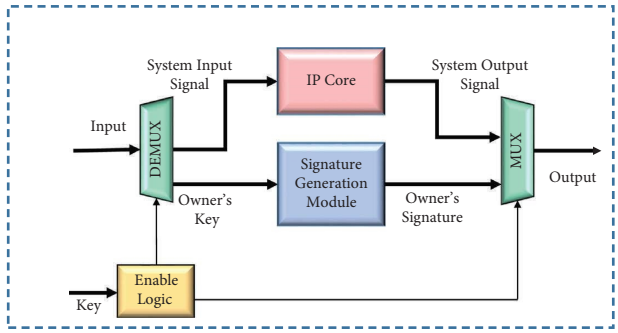

Figure 2: Block diagram of proposed mechanism.

core or SGM and final output is controlled by using logic encryption-based hardware obfuscation technique. Thus, the owner's identification signature generation is controlled by using logic encryption-based hardware obfuscation technique. Therefore, through switching mechanism, the embedded watermark can be made functional and accessible to legitimate end users off chip via test pin.

3.2. Mechanism of Hardware Obfuscation. In case of IP core without watermark, it is connected to system input and output signal. However, as described earlier, after the watermark integration, original IP core input and output data flow is controlled as depicted in Figure 2. When this watermarked IP core is integrated with the SoC, in the default state, enable logic select line is set to communicate input signal to IP core via DEMUX, and the IP core output is communicated as system output via MUX whereas, to access SGM, a separate test pin is introduced as a part of watermarked IP core, and correct key is needed to be fed at the enable logic, only after which enable logic select line is set to communicate system input signal to SGM via DEMUX, and the SGM output is communicated via MUX. Thus, the IP core or SGM can be accessed now selectively only through

select line of the enable logic of the watermark. In other words, when the input signal enters the watermarked IP core, it is to be accessed by IP core or SGM is decided by select line of DEMUX which is controlled by enable logic.

3.3. Workflow of Owner's Signature Generation Process. Herein, the two-stage mechanism for generation of the owner's signature using proposed method is described in detail.

3.3.1. First Stage: Activation. The Enable logic is set as an activation stage of the proposed SGM. In order to execute first stage of activation, the owner must feed the correct "Key" as an input to enable logic (refer Figure 3). The Key is an input data sequence that is designed, optimized, and stored as 8-bit sequence in Enable logic block. After receiving Key signal, the "Enable Logic" compares this input sequence with predefined Key stored in memory. If the input Key matches with the stored sequence, the select line switches from the default condition, due to which the next incoming input on *Input* bus will no longer be given to IP core, but it will be accessed by SGM. This switching of *Input* bus to SGM is referred to as the activation stage because it activates the SGM module and makes it ready to receive input and generate owner's signature.

3.3.2. Second Stage: Validation. After activation of the SGM, it is now ready to accept input from the user. The input from the user is validated by SGM after which owner's signature is generated. To generate this signature at output, a predefined input pattern signal needs to be fed to SGM. This input pattern signal "Owner's Key" is uniquely featured as a combination of alternate data streams and clock delay sequences, and the details of it are discussed later. Similarly, the SGM is designed to validate this Owner's Key. Insights of SGM are revealed in Figure 3. As can be seen in the figure, SGM comprises of signature check block and time scale verification block, and further, the output from both of these sections is AND together, and its output is fed to signature generation block. When the input pattern, which is an alternate data stream and clock delay sequences, enters in the SGM, data stream is validated through a signature check block, and time scale verification block validates clock delay sequences. It is necessary that both these blocks validate the complete input signal correctly, and therefore, outputs from both the blocks are AND together, and its output is fed to the signature generation block. Furthermore, the signature generation block generates the owner's signature.

3.3.3. Description of Key. The key signal is a predefined 8-bit binary signal, which is stored in the memory of Enable logic block. This key signal acts as a gateway for SGM. In order to activate SGM, this Key signal is to be fed to Enable logic. While integrating the watermark with IP core, the owner can design the specific Owner's Key signal and store it in the memory of the watermark, which is utilized later for verification.

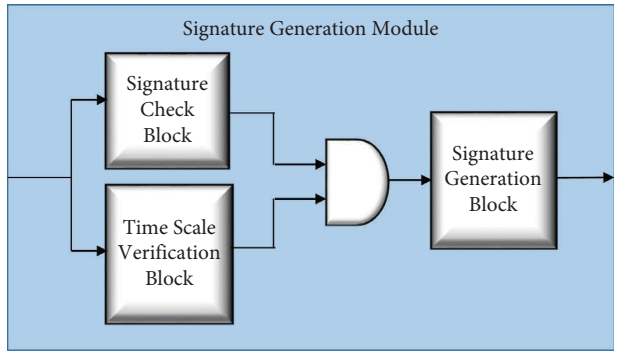

Figure 3: Insights of signature generation module.

3.3.4. Description of Owner's Key. After the activation stage is cleared, the Owner's Key input pattern signal is to be fed to SGM for validation. After validation of the input pattern signal, a unique signature is generated from SGM. As can be seen in Figure 4, the input pattern signal is the unique pattern made up of data streams and clock delays. In Figure 4, data streams are represented as colored blocks, and the clock delays are represented as white colored blocks, and they are arranged alternately.

In a similar way, the signature pattern of the owner is also designed as a unique pattern made up of data streams and clock delays. Both owner's key and owner's signature pattern are owner configurable at the design stage of watermark insertion, and the length of data streams and clock delays is also user/owner configurable. The data streams can be configured in multiple bits, and clock delay can be configured in multiple system clock ticks.

3.3.5. Flowchart. In Figure 5, the mechanism of the Owner's Signature generation is explained using a flowchart. As discussed earlier, the proposed mechanism consists of two stages, activation stage and validation stage, which are highlighted in green and blue colors, respectively. The step-by-step description of the flowchart is summarized in the form of the algorithm as follows.

3.3.6. Algorithm

(1) Input Key signal is accepted

(2) Activation stage sets/enables the entire SGM via select lines of DEMUX and MUX upon successful verification of "Key" input signal

(3) For validation stage, SGM accepts input pattern signal "Owner's Key" (alternate data streams and clock delay sequences) for verification, and this performs comparison in synchronization with system clock

(4) The data streams from input pattern signal are validated by signature check block

(5) Simultaneously, the time-scale verification block validates the clock delays between two data streams of input pattern signal in synchronization with system clock

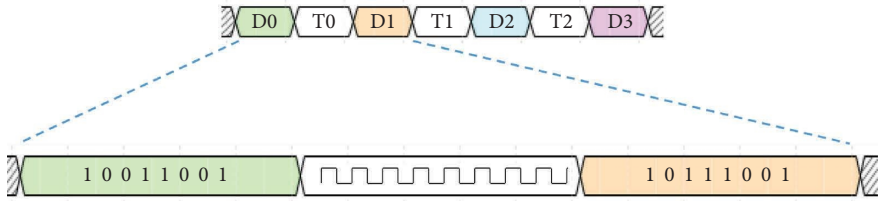

Figure 4: System input signal pattern.

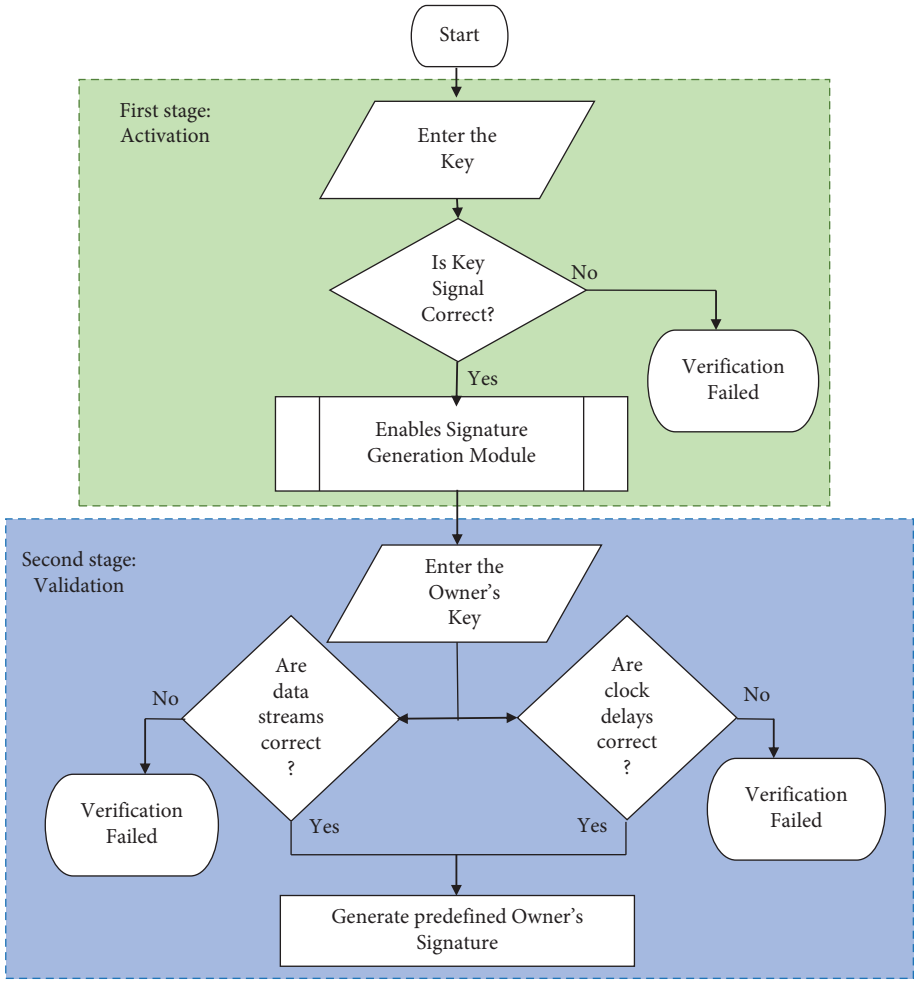

Figure 5: Flowchart of the proposed mechanism.

(6) Upon simultaneous and successful validation of input pattern, by signature check block and time-scale verification block, both blocks will generate "1"

(7) These two outputs are AND together which triggers the signature generation block to generate output

(8) Signature generation block will generate a predefined owner's signature which is stored in the memory as a signature of owner and formulated as clock ticks delay in between output signal

The algorithm explains step-by-step concise execution of whole SGM. The major secrecy is provided by the input pattern signal which is a combination of data streams and clock delays. To enhance the security, it is important to design Key pattern and input signal pattern trickily.

4. Result and Discussion

4.1. Tool Flow. The open-source tools are used to implement the present watermark technique. The benchmark circuits referred to for watermarking are in high-level synthesis (HLS) format. To convert HLS into Verilog file format, Bambu Panda tool is used. The proposed watermark is implemented in Verilog. These watermark-embedded

circuits are then converted to RTL format using RTL YOSYS tool. Furthermore, to convert RTL into GDSII design, OPENROAD tool with FreePDK45 (45 nm library) is used.

4.2. Demonstration of Owner's Signature Generation

4.2.1. Case Study. After implementing the proposed watermark generation method, various case studies are presented to describe how the first layer of protection and the second layer of protection work. Along with this, the case is described wherein the successful generation of owner's signature is explained. In all the cases, it is presumed that the watermark is embedded with IP core, and three possible cases are discussed.

4.2.2. Case 1. In the present case, let's consider that a random Key is fed to watermarked circuit. As shown in Figure 6(a), if the Key signal fed to Enable logic does not match with the Key signal stored in the memory of the watermark, then Enable logic will not enable or activate SGM. In this case, the next input pattern signal (owner's key) will not be fed to SGM for validation; hence, the obfuscated hardware will not generate the owner's signature. Moreover, the circuitry-producing owner's signature will remain inaccessible to the adversary. As mentioned earlier, in this situation, the obfuscated circuit will keep the system in the default state; i.e., next, input will be fed to IP Core. Thus, the purpose of the first layer of protection will be served.

4.2.3. Case 2. In this case, let's assume that the correct Key is fed, and after successful verification of the Key signal, Enable Logic enables/activates SGM as a consequence of which the select line of DEMUX is set to feed the system input to SGM and MUX is set to accept the output from SGM. Herein, the second stage of validation is started. After enabling the SGM, in order to generate owner's signature from SGM, it must be fed with a predetermined input signal which is a combination of alternate data stream and clock delay sequence (details are described in Case 3).

As shown in Figure 6(b), now let's assume that the owner's key is not correct. Since the output of both the blocks viz. signature check block and time scale verification block from SGM are AND together, it is necessary that both these blocks validate the input signal. Unsuccessful validation from either of these blocks will not generate the owner's signature, and the ownership of the IP will not be proven. Thus, the purpose of a second layer of protection will be served. In essence, two-stage verification is necessary for the generation of the owner's signature from the SGM.

4.2.4. Case 3. This case is presented to demonstrate the generation of the owner's signature using the proposed methodology (Figure 6(c)). Let's consider that the following authentic Key is fed to Enable Logic.

Key signal: 8 bit-1000 0000b.

Since the Key is authentic, Enable Logic after successful verification of the Key signal enables/activates SGM. Now, in the present case, suppose the authentic owner's key is designed using seven blocks wherein, D0, D1, D2, D3 are data stream blocks, and each block consists of 8 bits. Similarly, T0, T1, T2 are clock delay blocks, and each block consists of 8 ticks of clock delays. As shown below, all the data stream blocks and clock delay blocks are arranged in an alternate fashion, starting with data stream as a first block. This uniquely formulated input pattern signal is stored in the memory of watermark at the time of its integration with IP core.

Block 1 D0 =>1001 1001-8 bits
Block 2 T0 =>8 clock ticks delay
Block 3 D1 =>1011 1001-8 bits
Block 4 T1 =>8 clock ticks delay
Block 5 D2 =>1111 1001-8 bits
Block 6 T2 =>8 clock ticks delay
Block 7 D3 =>1111 1011-8 bits

When this correct input pattern signal is being fed to SGM, the data stream is validated through a signature check block, and the time scale block validates clock delay sequences. It is necessary that both these blocks validate the complete input signal correctly and therefore outputs from both the blocks are AND together, and its output is fed to the signature generation block wherein the signature generation block generates the valid owner's signature.

In the present case, the owner's signature is predesigned as follows. It is featured as a combination of seven blocks wherein D0, D1, D2, D3 are data stream blocks, and each block consists of 8 bits. Similarly, T0, T1, T2 are clock delay blocks, and each block consists of 10 ticks of clock delays. As shown as follows, all the data stream blocks and clock delays blocks are arranged in an alternate fashion, starting with data stream as a first block. This uniquely featured owner's signature is stored in the memory of watermark at the time of its integration with IP core.

Block 1 D0 =>1000 0001-8 bits
Block 2 T0 =>10 clock ticks delay
Block 3 D1 =>1000 0011-8 bits
Block 4 T1 =>10 clock ticks delay
Block 5 D2 =>1000 0111-8 bits
Block 6 T2 =>10 clock ticks delay
Block 7 D3 =>1000 1111-8 bits

When the correct input pattern is being fed to SGM, it generates this predefined owner's signature as the output of the SGM. The correctly generated signature at the output of SGM confirms the identification of the IP owner.

4.2.5. Variations. In the proposed method, Key for Enable Logic, Owner's Key, and Owner's Signature pattern are configurable at the design stage of watermark insertion. The data streams can be configured in multiple bits, and clock delay can be configured in multiple system clock

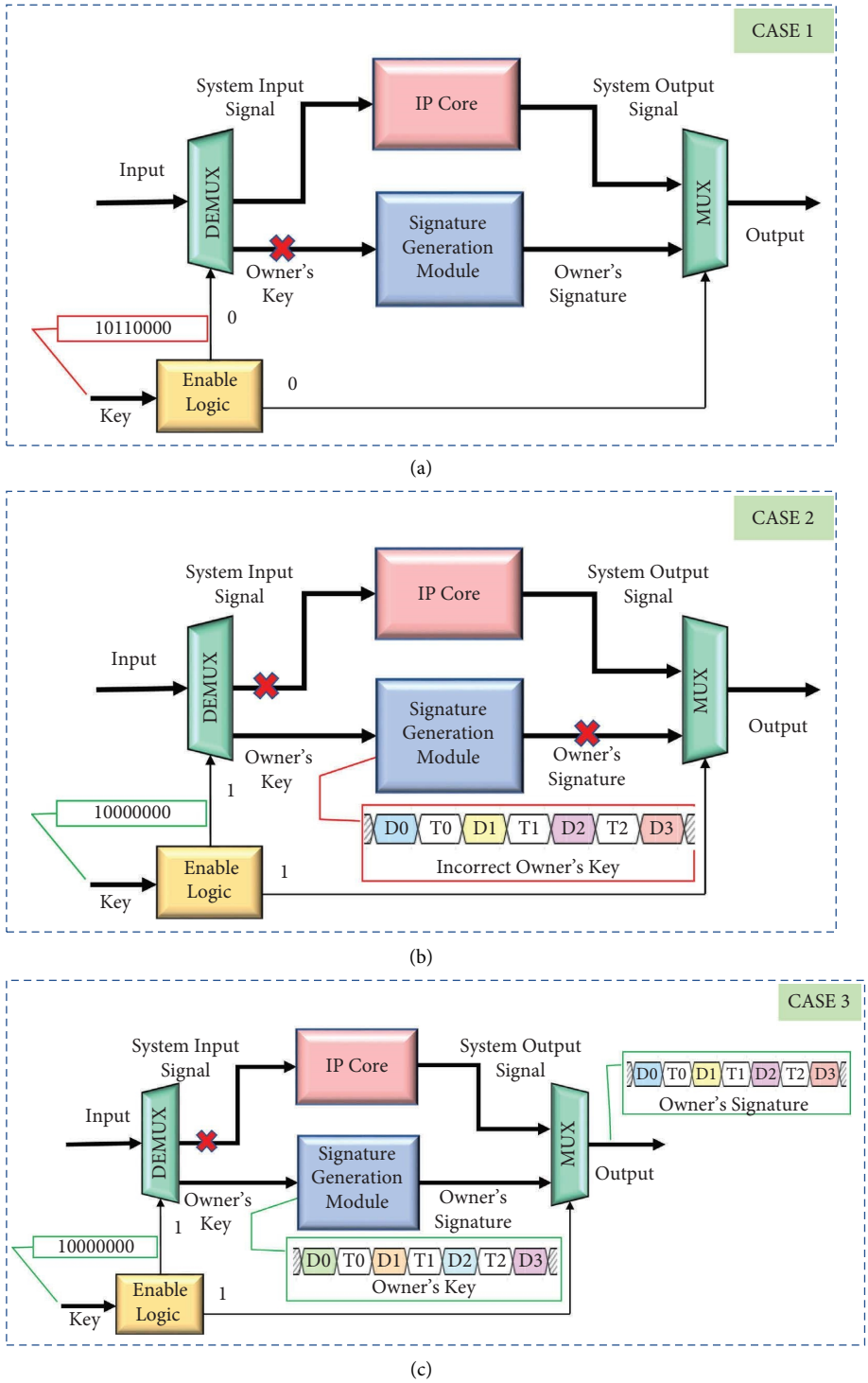

Figure 6: Pictorial representation of the cases: (a) case 1 with incorrect key signal to Enable logic, (b) case 2 with correct key signal to Enable logic but incorrect owner's key, and (c) case 3 with both key signal and owner's key are correct.

ticks. Therefore, to analyze the viability of these changes and its effect on area overhead and probability of coincidence, the study is extended for a few more combinational changes of data stream lengths and presented in the table format. Detail discussion of all these aspects is as follows.

4.3. Evaluation on Benchmarks. It is important to evaluate the proposed watermark technique on various benchmark circuits. Therefore, the evaluation of the proposed watermark techniques is carried out on EPFL [55] and IWLS [56] benchmark suites, from which the parameters such as the probability of coincidence (Pc) [57], area overhead, and

latency are estimated by designing for data and delay mentioned in Case 3. These parameters are analysed and summarized in the table format for each circuit before and after the watermark is being integrated with it.

4.3.1. Overhead and Comparison. Initially, the area for the circuits from EPFL and IWLS benchmark suites is calculated without integrating the watermark and summarized in the baseline column of Table 1. In these tables, IWLS benchmarks are differentiated from the EPFL benchmarks by highlighting them in grey colour. To estimate the area, the RTL file of each circuit is converted to GDSII design using OPENROAD tool with Free PDK45 (45 nm library), wherein the total area of each benchmark circuit is available.

To evaluate the proposed watermark method, different combinations of Key and Owner's Key are designed. These combinations are integrated with each benchmark circuit, and area overhead is calculated using the same tool and summarized. We compare our watermarking technique with few other techniques which are proposed as constraint-based IP watermarking, at different abstraction levels. We compared with polymorphic gates based watermarking method described in [16], the behavioural synthesis watermarking as proposed in [57], and the high-level synthesis based watermarking schemes described in [58, 59]. Those resulting values for [16, 59] are gathered from the study represented in [59]. In order to match with the configuration, some of the results are interpolated or extrapolated and summarized in Table 1. In order to evaluate the proposed watermarking system, various combinations of Key (bits length variation) and owner's key (data stream lengths and number of clock ticks variations in each block) are considered for watermark. Corresponding area overhead and probability of coincidence are calculated.

A nomenclature method is used to denote combinations, such as if Watermark 1 is denoted as K8:4D8:3C8, then this suggests that in a Watermark 1, K8 gives a description of Key and implies that Key is of 8 bits. Similarly, 4D8:3C8 gives a description of owner's key and implies that in an owner's key, 4D8 means 4 data stream blocks are used, of which each data stream block consists of 8 bits and 3C8 means 3 clock delays blocks are used, of which each clock delays block consists of 8 clock ticks delays. Area overhead and probability of coincidence are evaluated for all sixteen combinations and analysed.

In general, the components use to formulate watermark such as register, hardware functional units, and switching components contributes to the area overhead. In the present case, the use of basic components such as DEMUX and MUX exhibited lower overhead compared to other switching and compositional components used in other methods. For the method in the present study, sixteen different watermark combinations are proposed, and the respective area overhead due to all these combinations is calculated for each benchmark circuit. Table 1 shows the comparative analysis of Watermark 1 and Watermark 16 with the previous works [58, 59]. Area overhead and probability of coincidence for the rest of all 14 Watermarks are summarised in the table and given as supplementary information. It can be observed from Table 1 that for most of the benchmark circuits, the method proposed in the present study exhibits less area overhead compared to previous works [58, 59]. Watermark 1 is K8:4D8:3C8, which is the smallest proposed watermark in terms of the number of bits in the Key, and similarly, data string length per block and clock tick delays per block of owner's key exhibit area overhead below 0.6% for all benchmark circuit except fft8 (since baseline area for fft8 is very small). However, Watermark 16, which is the largest considered watermark in terms of the number of bits in the Key, and similarly, data string length per block and clock tick delays per block of owner's key, exhibits area overhead below 1.4% for all benchmark circuits (except fft8). This marginal increase in area overhead suggests that the proposed method allows a lot of variations within the watermark combination realm with no significant impact on area overhead whereas Watermark methods proposed in references [58, 59] exhibited an area overhead maximum up to 9.5% and 4.5%, respectively, which is significantly larger compared to the area overhead proposed in the present study.

Similarly, Table 2 summarises the latency exhibited by the proposed method and its comparison with other methods. Since the watermark in the present method is embedded outside the IP core and it does not utilize any of the resources from IP core, it does not cause any latency in the IP core functionality. On the other hand, as can be seen from the table, other methods wherein watermarking is designed using resources of the IP components exhibit latency to the circuit.

4.3.2. Probability of Coincidence. It is the probability of guessing the correct input signal pattern to generate owner's signature. In the case of the watermark, its protection strength can be indicated by this probability of coincidence. It is worth to mention here that the word "probability" does not signify its exact mathematical meaning strictly; rather, the term indicates here the approximation of the actual probability [16]. Therefore, in the present case, design robustness in terms of guessing the correct key signal for Enable logic can be calculated as $2^8 = 256$ different possible combinations which implies that guessing the correct key for Enable logic will be time-consuming. However, only enabling the SGM module will not produce the owner's signature and will not be helpful in presenting the false claim. Moreover, for two-stage verification and generation of owner's signature, the correct input owner's key pattern is needed to feed SGM. Therefore, design robustness in terms of guessing the correct input pattern of key and owner's key can be calculated as permutation for all possible values. This can be expressed in terms of probability of coincidence (P_C) and calculated as $P_C = 1/(\text{permutations of key bits } x \text{ permutations of bits in each data stream block} + \text{clock tick delays})$. This reflects the possibility of guessing the authentic Key and owner's key. In accordance with this, P_C is calculated for all sixteen studied watermark combinations and summarized in Table 3.

Table 1: Comparative analysis of area overhead and probability of coincidence (P_C).

Benchmark circuit	Baseline	[58]		[59]		Watermark 1 K8: 4D8:3C8		Watermark 16 K64:4D64:3C64	
	Area (μm^2)	Area (μm^2)	% overhead	Area (μm^2)	% overhead	Area (μm^2)	% overhead	Area (μm^2)	% overhead
Ethernet	87.462	95.741	9.466	88.511	1.2	87.951	0.56	88.675	1.39
dft	229.643	245.258	6.8	233.983	1.89	230.134	0.214	230.856	0.53
Md5Core	210.335	220.881	5.014	213.237	1.38	210.827	0.234	211.548	0.58
fft8	10.78	11.107	2.87	11.388	5.47	11.229	4	12.011	11.23
Eliptic wave filter	222.756	237.903	6.8	225.985	1.45	225.248	0.221	223.969	0.54
JPEG: inverse discrete cosine transform	189.869	192.052	1.15	191.482	0.85	190.423	0.292	191.082	0.64
		$P_c = 9.56e-18$		$P_c = 3.72e-53$		$P_c = 9.09e-13$		$P_c = 4.68e-97$	

Table 2: Latency due to watermark.

Benchmark circuit	Latency (ns)			
	Baseline	[58]	[59]	Watermark 1
Ethernet	5.13	5.47	5.57	5.13
dft	18.23	21.32	19.23	18.23
Md5Core	20.3	23.54	24.8	20.3
fft8	4.901	5.2	5.6	4.901
Elliptic wave filter	4.21	4.63	4.46	4.21
JPEG: inverse discrete cosine transform	19.37	22.46	20.05	19.37

Table 3: Probability of coincidence (P_C) for watermarks.

Watermark		Pc
Watermark 1	K8:4D8:3C8	$9.09E-13$
Watermark 2	K8:4D16:3C16	$2.12E-22$
Watermark 3	K8:4D32:3C32	$1.15E-41$
Watermark 4	K8:4D64:3C64	$3.37E-80$
Watermark 5	K16:4D8:3C8	$3.55E-15$
Watermark 6	K16:4D16:3C16	$8.27E-25$
Watermark 7	K16:4D32:3C32	$4.48E-44$
Watermark 8	K16:4D64:3C64	$1.32E-82$
Watermark 9	K32:4D8:3C8	$5.42E-20$
Watermark 10	K32:4D16:3C16	$1.26E-29$
Watermark 11	K32:4D32:3C32	$6.84E-49$
Watermark 12	K32:4D64:3C64	$2.01E-87$
Watermark 13	K64:4D8:3C8	$1.26E-29$
Watermark 14	K64:4D16:3C16	$2.94E-39$
Watermark 15	K64:4D32:3C32	$1.59E-58$
Watermark 16	K64:4D64:3C64	$4.68E-97$

However, to present a consolidated view on analysis and comparison with the previous works, the probability of coincidence for Watermark 1, 16, and previous works [58, 59] is listed in Table 1 at the bottom. Watermark 1 is the smallest watermark (in terms of the number of bits in the Key) and, similarly, data string length per block and clock tick delays per block of owner's key, i.e., K8:4D8:3C8 and exhibits area overhead below 0.6% for all benchmark circuit. For Watermark 1, P_C is 9.09 $e-13$, which implies that almost infinite combinations are to be explored by attackers to guess the correct input pattern to generate owner's signature. Similarly, Watermark 16 is the largest considered watermark (K64:4D64:3C64), exhibits area overhead below 1.4% for all benchmark circuits, and shows P_C as 4.68 $e-97$ whereas, for the methods proposed in the previous works, [58, 59] shows Pc as 9.56 $e-18$ and 3.72 $e-53$ (value considered for 64-bit Watermark for comparison), respectively. In the present study, the analysis revealed that for the proposed method, P_C can drop down drastically if the watermark combination contains a large number of bits in the Key, and similarly, data string length per block and clock tick delays per block of owner's key are increased. Moreover, the marginal increase in area overhead due to various watermark combinations demonstrates the robustness of the proposed method.

4.4. Attack Analysis. In the case of watermark methods, mainly, three types of attack analysis are to be taken into consideration, viz. removal, masking, and forging.

4.4.1. Strength of Hardware Obfuscation. Hardware obfuscation offers a preventive measure by obscuring the system design and architecture. The hardware obfuscation resulted from the deliberate changes or restructuring in the interconnectivity of the various functional units. This method is used for obscuring the data path architecture and controlling logic without affecting the functionality of the IP core. Due to this, understanding the logic architecture becomes unobvious for an adversary, which makes reverse engineering or identification of the correct functional flow of the design harder. Also, since the architecture of the design remains concealed from the adversary, the possibility of malicious logic insertion is reduced. In addition to this, a Key based Enable Logic circuitry enhances the strength of hardware obfuscation. It is demonstrated that the second stage of validation can be accessed through this obfuscated circuit only when the correct Key is fed to Enable Logic. This significantly enhances the strength of the first layer of protection.

4.4.2. Removal and Masking of Watermarks. The watermark presented in this study comprises MUX and DEMUX, which are also one of the basic components used to design IP cores, because of which identifying these components as a part of watermark is very difficult. Furthermore, while integrating the watermark with IP core, the input and output of the IP core are restricted to be accessed through DEMUX and MUX only. Additionally, enable logic select line is set to give system input to IP core via DEMUX, and the system output is passed via MUX. Thus, this watermarked IP core can be accessed now only through the select line of the Enable logic of the watermark. In other words, when the system input signal enters the watermarked IP core, it is to be accessed by IP core or SGM is decided by the select line of DEMUX which is controlled by enable logic. Certainly, in case of removal of the watermark, the IP core will not receive the system input, and this will leave the IP core redundant, making it no longer useable. A similar type of IP core malfunctioning will occur in case of masking of watermark since in this case also input will no longer be accessible to IP core.

4.4.3. Forging of Watermarks. The possibility of forging attacks can be considered, wherein the adversary may add his own watermark to the original IP. In this case, the owner can rightfully prove the IP ownership by presenting an IP with his own watermark demonstration while the adversary can be easily figured out due to the presence of two watermarks.

Moreover, as a future scope of the proposed method, a few modifications can be considered. In the design of SGM (consider Figure 2), additional circuitry can be added next to AND gate, which will accept the input from the AND gate. Herein, based on the data streams inserted as a part of owner's key, encryption-based submodule can be designed for owner's signature generation. However, this will cause the area overhead to the circuitry. This also highlights the ever-existing challenge of achieving a better trade-off between the security offered by the watermark and area overhead, latency it causes. At this point, the choice of a cost-effective solution is a true challenge.

5. Conclusions

This paper presented a new robust dynamic watermarking method. Instead of using any resources of IP core, it is isolated to keep IP core functionality unaltered, and outside resources are utilized for watermarking. A novel way of embedding the authorship information using logic encryption-based hardware obfuscation method is employed. This method is configured to allow input-based selective running of either IP core or SGM (watermark). Furthermore, the two-stage mechanism is set to generate owner's signature. As a first stage, the proposed method obscures the data path architecture and controls logic without affecting the functionality of the IP core. This offers counter-measures against reverse engineering or makes identification of the correct functional flow of the design harder. Additionally, the first stage acts as a pathway for second stage (access to SGM). Therefore, second stage of validation can be accessed through this obfuscated circuit only when the correct Key is fed to Enable Logic. Second stage is SGM which is a validation circuit designed to identify authentic owner's key and generate owner's signature. Thereafter, data streams and clock tick delays are used in combination to formulate an owner's key pattern and an owner's signature. This innovative design of keys makes it harder to guess the correct pattern and the probability of coincidence drops down dramatically (P_C as $4.68\ e-97$). Besides this, since the watermark in the proposed method is embedded outside the IP core, it does not cause any latency to the IP core functionality. Moreover, the basic components used to formulate the watermark architecture, and the embedding method exhibits lower area overhead (up to 1.4%), yet the proposed method is capable of providing the competitive way of IP core watermarking.

Data Availability

The data used to support the findings of this study are available in the manuscript.

Disclosure

This research is part of PhD. program enrolled by the primary author at the College of Engineering, Pune.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] M. Schmid, D. Ziener, and J. Teich, "Netlist-level IP protection by watermarking for LUT-based FPGAs," in *Proceedings of the 2008 International Conference on Field-Programmable Technology*, pp. 209–216, IEEE, Taipei, Taiwan, December 2008.

[2] M. M. Khan and S. Tragoudas, "Rewiring for watermarking digital circuit netlists," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 7, pp. 1132–1137, 2005.

[3] P. Bricaud, *Reuse Methodology Manual: For System-On-A-Chip Designs*, Springer Science & Business Media, Berlin, Germany, 2012.

[4] H. Chang, L. Cooke, M. Hunt, G. Martin, A. McNelly, and L. Todd, *Surviving the SOC Revolution: A Guide to Platform-Based Design*, Kluwer Academic publisher, Alphen aan den Rijn, Netherlands, 1999.

[5] G. Martin and H. Chang, *Winning the SoC Revolution: Experiences in Real Design*, Springer US, New York, NY, USA, 2011.

[6] B. Le Gal and L. Bossuet, "Automatic low-cost IP watermarking technique based on output mark insertions," *Design Automation for Embedded Systems*, vol. 16, no. 2, pp. 71–92, 2012.

[7] M. G. Pecht and S. Tiku, "Bogus: electronic manufacturing and consumers confront a rising tide of counterfeit electronics," *IEEE Spectrum*, vol. 43, no. 5, pp. 37–46, 2006.

[8] D. S. Fernandez, "Intellectual property protection in the EDA industry," in *Proceedings of the 31st annual Design Automation Conference (DAC '94)*, pp. 161–163, Association for Computing Machinery, San Diego, CA, USA, June 1994.

[9] J. A. Nestor, "Work in progress-a new course on Intellectual Property, innovation, and ethics," in *Proceedings of the 2009 39th IEEE Frontiers in Education Conference*, pp. 1-2, San Antonio, TX, USA, October 2009.

[10] J. Lach, H. William, Mangione-Smith, and M. Potkonjak, "Signature hiding techniques for FPGA intellectual property protection," in *Proceedings of the 1998 IEEE/ACM international conference on Computer-aided design (ICCAD '98)*, pp. 186–189, Association for Computing Machinery, San Jose, CA, USA, November 1998.

[11] M. Barni, F. Bartolini, I. J. Cox, J. Hernandez, and F. Perez-Gonzalez, "Digital watermarking for copyright protection: a. communications perspective," *IEEE Communications Magazine*, vol. 39, no. 8, pp. 90-91, 2001.

[12] A. Sengupta and S. Bhadauria, "Exploring low cost optimal watermark for reusable IP cores during high level synthesis," *IEEE Access*, vol. 4, pp. 2198–2215, 2016.

[13] A. Sengupta and M. Rathor, "Hardware (IP) watermarking during behavioral synthesis," *Behavioral Synthesis for Hardware Security*, pp. 119–145, Springer, New York, NY, USA, 2022.

[14] R. Karmakar and S. Chattopadhyay, "Hardware IP protection using logic encryption and watermarking," in *Proceedings of the 2020 IEEE International Test Conference (ITC)*, Washington, DC, USA, November 2020.

[15] C. Pilato, S. Garg, K. Wu, R. Karri, and F. Regazzoni, "Securing hardware accelerators: a new challenge for high-level synthesis," *IEEE Embedded Systems Letters*, vol. 10, no. 3, pp. 77–80, 2018.

[16] A. Sengupta, D. Roy, and S. P. Mohanty, "Triple-phase watermarking for reusable IP core protection during architecture synthesis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 4, pp. 742–755, 2018.

[17] J. G. de Lamadrid and S. Choi, "Hardware watermarking for finite state machines, with symmetric circuit encryption," 2022, https://arxiv.org/abs/2203.12097.

[18] P. Santikellur, R. S. Chakraborty, and S. Bhunia, "Hardware IP protection using register transfer level locking and obfuscation of control and data flow," *Behavioral Synthesis for Hardware Security*, pp. 57–69, Springer, New York, NY, USA, 2022.

[19] A. Sengupta, E. R. Kumar, and N. P. Chandra, "Embedding digital signature using encrypted-hashing for protection of DSP cores in CE," *IEEE Transactions on Consumer Electronics*, vol. 65, no. 3, pp. 398–407, 2019.

[20] A. Anshul and A. Sengupta, "IP core protection of image processing filters with multi-level encryption and covert steganographic security constraints," in *Proceedings of the 2022 IEEE International Symposium on Smart Electronic Systems (iSES) 2022*, pp. 83–88, Warangal, India, December 2022.

[21] A. Sengupta and M. Rathor, "Enhanced security of DSP circuits using multi-key based structural obfuscation and physical-level watermarking for consumer electronics systems," *IEEE Transactions on Consumer Electronics*, vol. 66, no. 2, pp. 163–172, May 2020.

[22] R. Naveenkumar, N. M. Sivamangai, A. Napolean, and G. A. Nissi, "Hardware obfuscation for IP protection of DSP applications," *Journal of Electronic Testing*, vol. 38, no. 1, pp. 9–20, Feb. 2022.

[23] M. Rathor, A. Anshul, K. Bharath, R. Chaurasia, and A. Sengupta, "Quadruple phase watermarking during high level synthesis for securing reusable hardware intellectual property cores," *Computers & Electrical Engineering*, vol. 105, Article ID 108476, 2023.

[24] A. Cui, C.-H. Chang, and L. Zhang, "A hybrid watermarking scheme for sequential functions," in *Proceedings of the 2011 IEEE International Symposium of Circuits and Systems (ISCAS)*, pp. 2333–2336, IEEE, Rio de Janeiro, Brazil, May 2011.

[25] A. T. Abdel-Hamid, S. Tahar, and E. M. Aboulhamid, "A survey on IP watermarking techniques," *Design Automation for Embedded Systems*, vol. 9, no. 3, pp. 211–227, 2005.

[26] A. L. Oliveira, "Robust techniques for watermarking sequential circuit designs," in *Proceedings of the 36th annual ACM/IEEE Design Automation Conference*, pp. 837–842, New Orleans, LA, USA, June 1999.

[27] A. L. Oliveira, "Techniques for the creation of digital watermarks in sequential circuit designs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 20, no. 9, pp. 1101–1117, 2001.

[28] I. Torunoglu and E. Charbon, "Watermarking-based copyright protection of sequential functions," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 3, pp. 434–440, 2000.

[29] Abdel-Hamid, T. Amr, S. Tahar, and E. M. Aboulhamid, "A public-key watermarking technique for IP designs," in *Proceedings of the Design, Automation and Test in Europe*, pp. 330–335, IEEE, Munich, Germany, March 2005.

[30] H. J. Kim, W. H. Mangione-Smith, and M. Potkonjak, "Protecting ownership rights of a lossless image coder through hierarchical watermarking," in *Proceedings of the 1998 IEEE Workshop on Signal Processing Systems. SIPS 98. Design and Implementation (Cat. No. 98TH8374)*, pp. 73–82, IEEE, Cambridge, MA, USA, October 1998.

[31] A. Rashid, J. Asher, H. William, Mangione-Smith, and M. Potkonjak, "Hierarchical watermarking for protection of DSP filter cores," in *Proceedings of the IEEE 1999 Custom Integrated Circuits Conference (Cat. No. 99CH36327)*, pp. 39–42, IEEE, San Diego, CA, USA, May 1999.

[32] A. Cui and C.-H. Chang, "Intellectual property authentication by watermarking scan chain in design-for-testability flow," in *Proceedings of the 2008 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 2645–2648, IEEE, Seattle, WA, USA, May 2008.

[33] A. Cui and C.-H. Chang, "An improved publicly detectable watermarking scheme based on scan chain ordering," in *Proceedings of the 2009 IEEE International Symposium on Circuits and Systems*, pp. 29–32, IEEE, Taipei, Taiwan, May 2009.

[34] C.-H. Chang and A. Cui, "Synthesis-for-testability watermarking for field authentication of VLSI intellectual property," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 7, pp. 1618–1630, 2010.

[35] G. Qu and L. Yuan, "Secure hardware IPs by digital watermark," *Introduction to Hardware Security and Trust*, pp. 123–141, Springer, New York, NY, USA, 2012.

[36] G. Qu and M. Potkonjak, "Analysis of Watermarking Techniques for Graph Coloring Problem," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers*, pp. 190–193, San Jose, CA, USA, November 1998.

[37] T.-B. Huynh, T. H. Trong, and B. Trong-Tu, "A constraint-based watermarking technique using Schmitt Trigger insertion at logic synthesis level," in *Proceedings of the 2013 International Conference on Advanced Technologies for Communications (ATC 2013)*, pp. 115–120, IEEE, Ho Chi Minh City, Vietnam, October 2013.

[38] D. Kirovski, Y.-Y. Hwang, M. Potkonjak, and J. Cong, "Protecting combinational logic synthesis solutions," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 12, pp. 2687–2696, 2006.

[39] S. Meguerdichian and M. Potkonjak, "Watermarking while preserving the critical path," in *Proceedings of the 37th Annual Design Automation Conference*, pp. 108–111, Los Angeles, CA, USA, June 2000.

[40] M. Ni and Z. Gao, "Constraint-based watermarking technique for hard IP core protection in physical layout design level," in *Proceedings of the 7th International Conference on Solid-State and Integrated Circuits Technology*, pp. 1360–1363, IEEE, Beijing, China, October 2004.

[41] R. Chapman and T. S. Durrani, "IP protection of DSP algorithms for system on chip implementation," *IEEE Transactions on Signal Processing*, vol. 48, no. 3, pp. 854–861, 2000.

[42] M. Lewandowski, R. Meana, M. Morrison, and K. Srinivas, "A novel method for watermarking sequential circuits," in *Proceedings of the 2012 IEEE International Symposium on Hardware-Oriented Security and Trust*, pp. 21–24, IEEE, San Francisco, CA, USA, June 2012.

[43] D. Kirovski and M. Potkonjak, "Intellectual property protection using watermarking partial scan chains for sequential logic test generation," in *Proceedings of the IEEE High Level Design, Verification, and Test Conference*, San Francisco, CA, USA, June 1998.

[44] J. Echavarria, A. Morales-Reyes, R. Cumplido, and M. A. Salido, "FSM merging and reduction for IP cores watermarking using genetic algorithms," in *Proceedings of the 2014 International Conference on ReConFigurable Computing and FPGAs (ReConFig14)*, pp. 1–7, IEEE, Cancun, Mexico, December 2014.

[45] V. V. Sergeichik, A. A. Ivaniuk, and C.-H. Chang, "Obfuscation and watermarking of FPGA designs based on constant value generators," in *Proceedings of the 2014 International Symposium on Integrated Circuits (ISIC)*, pp. 608–611, IEEE, Singapore, December 2014.

[46] F. Leitao, "Intellectual property (IP) protection using Watermarking and Fingerprinting techniques," in *Proceedings of the 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, pp. 433–438, IEEE, Bangalore, India, July 2016.

[47] X. Huang, A. Cui, and C.-H. Chang, "A new watermarking scheme on scan chain ordering for hard IP protection," in *Proceedings of the 2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–4, IEEE, Baltimore, MD, USA, May 2017.

[48] R. Karmakar, S. J. Suman, and S. Chattopadhyay, "A cellular automata guided finite-state-machine watermarking strategy for IP protection of sequential circuits," *IEEE Transactions on Emerging Topics in Computing*, vol. 10, no. 2, pp. 806–823, 2020.

[49] R. S. Chakraborty and S. Bhunia, "HARPOON: an obfuscation-based SoC design methodology for hardware protection," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 10, pp. 1493–1502, 2009.

[50] C. Collberg, C. Thomborson, and D. Low, "Manufacturing cheap, resilient, and stealthy opaque constructs," in *Proceedings of the Conference Record of the Annual ACM Symposium on Principles of Programming Languages*, pp. 184–196, San Diego, CA, USA, January 1998.

[51] S. Katkoori and A. I. Sheikh, *Behavioral Synthesis for Hardware Security*, Springer, New York, NY, USA, 2022.

[52] A. R. Desai, M. S. Hsiao, C. Wang, L. Nazhandali, and S. Hall, "Interlocking obfuscation for anti-tamper hardware," in *Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop*, Oak Ridge, TN, USA, January 2013.

[53] X. Zhuang, T. Zhang, H. H. S. Lee, and S. Pande, "Hardware assisted control flow obfuscation for embedded processors," in *Proceedings of the CASES 2004: International Conference on Compilers, Architecture, and Synthesis for Embedded Systems*, pp. 292–302, Washington, DC, USA, September 2004.

[54] D. Forte, B. Swarup, and M. T. Mark, *Hardware protection through Obfuscation*, Springer International Publishing, Heidelberg, Germany, 2017.

[55] L. Amarú, P.-E. Gaillardon, and G. de Micheli, "The EPFL combinational benchmark suite," 2015, https://github.com/lsils/benchmarks.

[56] C. Albrecht, "IWLS 2005 benchmarks," 2005, https://iwls.org/iwls2005/benchmarks.html.

[57] F. Koushanfar, I. Hong, and M. Potkonjak, "Behavioral synthesis techniques for intellectual property protection," *ACM Transactions on Design Automation of Electronic Systems*, vol. 10, no. 3, pp. 523–545, 2005.

[58] S. Rai, A. Rupani, P. Nath, and A. Kumar, "Hardware watermarking using polymorphic inverter designs based on reconfigurable nanotechnologies," in *Proceedings of the 2019 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pp. 663–669, IEEE, Miami, FL, USA, July 2019.

[59] A. Sengupta, S. Bhadauria, and S. P. Mohanty, "Embedding low cost optimal watermark during high level synthesis for reusable IP core protection," in *Proceedings of the 2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 974–977, Montreal, QC, Canada, July 2016.