Low Cost Dual-Phase Watermark for Protecting CE Devices in IoT Framework

Anirban Sengupta and Dipanjan Roy

Abstract Intellectual property (IP) core providers are increasingly aware of the need to protect their investment from either counterfeit/forgery or illegal ownership. This chapter presents a novel low cost dual phase watermarking methodology during high level synthesis (HLS) for IP core protection of vendor. Robust vendor signature is embedded in two subsequent phases of high level synthesis to form an integrated watermark. We present a dual-phase watermarking methodology that embeds a multi-variable double phase watermarking during high level synthesis for application specific IPs (application specific integrated circuits) that incurs zero delay and register overhead as well as minimal hardware overhead. The dual-phase watermarking approach yields average reduction of embedding cost of 6% (which includes average area reduction of 7% and average latency reduction of 4%) when compared to two recent HLS based watermarking approaches for application specific IPs. Additionally, the approach also achieves stronger proof of authorship compared to two recent HLS based watermarking approaches.

1 Introduction

Internet-of-Things (IoT) represents interconnection (communication system) of smart devices, sensors, computing devices etc (including consumer electronics) through modern network technologies. IoT is playing an essential role in home electronics, from entertainment to smart home control. In the domain of IoT, driven by consumer electronics (CE) hardware, the importance of Electronics Design Automation (EDA) is pivotal. In the modern era of EDA for CE device, surging complexity of design is out-pacing the design productivity. The significance of reusable Intellectual Property (IP) [1–3] core for CE hardware is to cope up with these complex design

A. Sengupta (✉) · D. Roy
Discipline of Computer Science and Engineering, Indian Institute of Technology Indore, Indore, India
e-mail: asengupt@iiti.ac.in

D. Roy
e-mail: phd1501201007@iiti.ac.in

R. S. Chakraborty et al. (eds.), *Security and Fault Tolerance in Internet of Things*, Internet of Things, https://doi.org/10.1007/978-3-030-02807-7_2

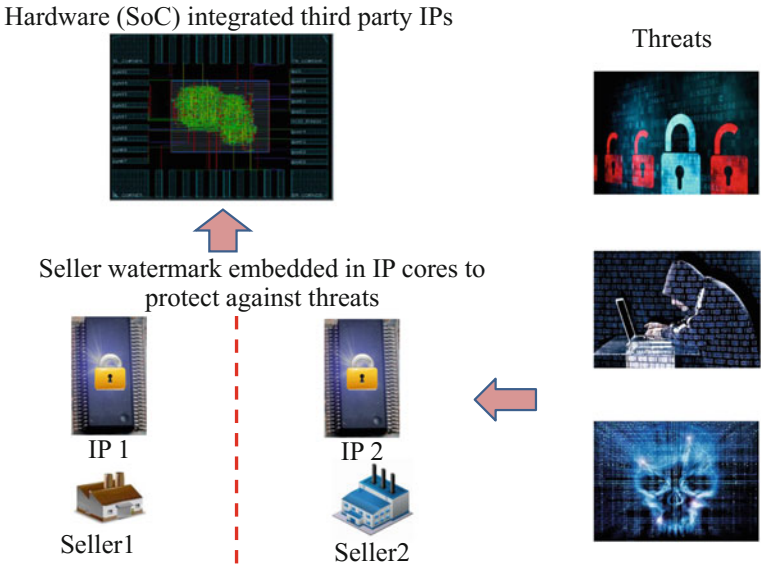

Fig. 1 Overview of IP core threats and its protection mechanism

requirements by reducing design time and enhancing design productivity. This makes IP core a promising and popular solution in the industry. However, with the increase in globalization and competition between the IP sellers/vendors, typical attacks such as IP counterfeiting/cloning, ownership abuse are escalating. Protection of IP core is essential to advance the benefits of reusable IP core [2–12]. However, while protecting an IP core correct functionality and accuracy must be retained. Ownership of IP ownership may be achieved by activating and deactivating each IP core by the system designer [13], through a reversible data hiding approach [14] or by information hiding techniques through steganography [15, 16]. Implanting a invisible owner's signature (known as watermark) is also one of the effective and promising option to protect a reusable IP core against ownership abuse and IP infringements (refer Fig. 1).

2 Overview of IP Core Protection Through Watermark

In this sub-section, we discuss some of the well-known powerful watermarking methodologies along with their differences with the presented dual-phase watermarking approach. Several watermarking techniques as discussed in [1, 11, 17–19] are embedded at lower design abstraction level. For example side channel based watermarking technique is discussed in [17], where, the vendor watermark is inserted into the netlist and bitstream of an IP design. An in-synthesis IP core watermarking

process is proposed in [11]. Though these approaches are useful however they do not protect designs at higher abstraction levels and incur more implementation complexity. Further [1, 18] is only applicable for Field-programmable Gate Arrays (FPGAs), not for Application Specific Integrated Circuits (ASICs). Additionally, both [18, 19] do not embed multi-phase watermarking during High-level synthesis (HLS) of IP design. Watermarking has been applied at higher design abstraction level also, for example, authors in [2, 3] insert watermarking in register assignment phase of HLS. Specifically, in [2] a dual variable encoding scheme (sequence of '0' and '1') is proposed for watermarking whereas in [3] a multi variable encoding scheme (sequence of 'i', 'I', 'T' and '!') is proposed for watermarking. Both of these approaches add additional edges in colored interval graph to achieve single-phase watermark (i.e. during register assignment). Thus both, robustness and tamper resistance of [1–3, 11, 18, 19], are significantly less than a multi-phase HLS based watermark [20].

Several multi-level hierarchical watermarking techniques are also discussed in [21–23]. These techniques embed watermark in multiple design abstraction levels which offer strong proof of ownership and high degree of tamper resistance, but does not explicitly implant dual-phase watermarking constraints at architectural level (during high level synthesis steps). Further, aforesaid approaches may also incur higher implementation complexity and design overhead as the watermarks are embedded in multiple independent levels.

2.1 Motivation of Embedding Dual Phase Watermark for IP Protection at Behavioral Level

This discussion in this chapter is motivated by the fact that embedding watermark at higher design abstraction (e.g. behavioral level) may be more advantageous than embedding watermark at lower design abstraction (e.g. gate level or layout level). This is because embedding watermark at higher design abstraction enables design protection in subsequent lower levels (as watermark constraints embedded at higher level propagate with design synthesis). Moreover, the embedding process incurs lesser implementation complexity. In addition to that, embedding watermark in dual phase within same abstraction level not only ensures more robustness, lower overhead and higher tamper resistance watermark but also increases difficulty of reverse engineering, compared to only single phase watermark or multiple design level based watermark designs [20].

The main focus of this chapter is to present a multi-variable dual phase (register assignment and scheduling) watermarking methodology embedded at architectural level for protecting against abuse of IP vendor's ownership. The additional design constraints due to watermark is embedded at architectural level in two different phases of high level synthesis i.e. functional unit assignment and scheduling. It must be noted that the presented dual-phase watermark embedding process embeds the watermark in same abstraction level but in multiple phases, however without incur-

ring much overhead and complexity [20]. Moreover, it is capable to achieve strong proof of ownership and high degree stronger robustness as it distributes watermarking constraints across two independent design phases compared to single phase. Though it is a well known fact that insertion of watermark may result in design overhead with respect to hardware area, execution delay, power etc., nevertheless, the design overhead obtained by other HLS-based watermark [2, 3] is larger than the presented dual-phase watermark due to encoding rules devised. Experimental results validate the reduction in overhead and increase in robustness compared to [2, 3], as highlighted in [20].

The remaining parts of this chapter is structured as: Sect. 2 explains the related work. Section 3 illustrate the dual-phase watermarking methodology while Sect. 4 demonstrate a motivational example. Section 5 reports the results and analysis, and conclusion in Sect. 6.

3 Dual-Phase Watermarking Methodology

3.1 Problem Formulation

From a given application in the form of data flow graph (DFG) and user specified hardware configuration $(X_i) = N(R_1), N(R_2), \ldots, N(R_D)$, design a watermarked IP core solution where $N(R_D)$ is the number of hardware of type R_D.

Threat Model: The methodology presented in this chapter protects a reusable IP core from following threats: ownership abuse and IP counterfeit/forgery. Thus, possible attacks of a watermarked IP are: illegal claim of an IP and partial/complete removal of an IP watermark.

Target Platform/Technology: The dual-phase watermarking technique is easily adaptable to any modern EDA tool. Any hardware description language (HDL) based EDA tool can be merged with the aforesaid approach.

3.2 Dual-Phase Watermark Encoding

This sub-section discusses the presented dual-phase watermark encoding process. Figure 2 presents a overview of the dual-phase watermarking approach. In the approach we assume that the IPs are imported from two different third party IP vendors because multi-vendor based IPs during system on chip (SoC) integration is a common practice in the industry. In several published literatures [24–26], the concept of two IP vendors for SoC design have been used. The dual-phase watermark is embedded during scheduling and hardware assignment phases of HLS [20]. The IP design in terms of assignment and scheduling phases are depicted through two different tables (a) "*non-critical operations* ($\mu_m > 0$)" table and (b) "*hardware*

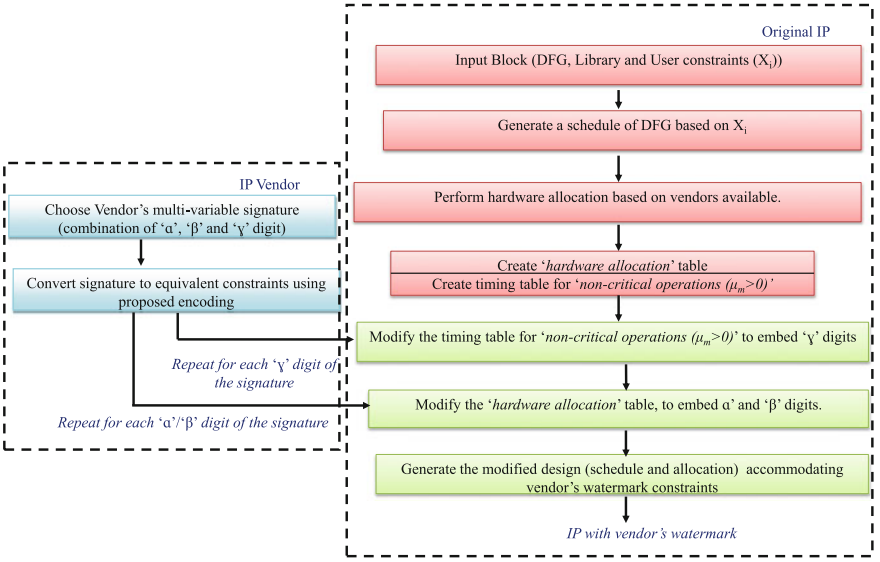

Fig. 2 IP core protection at behavioral level using dual-phase watermark [20]

assignment" timing table, where μ_m indicates the mobility of an operation. This dual-phase watermarking approach is comprises of three different digits viz. 'α', 'β', 'γ' where each digit indicates a specific encoding. A vendor's signature as a combination of 'α', 'β' and 'γ' is converted into its respective watermarking constraints based on the encoding rule of each digit, which is then subsequently embedded as additional constraints in the design. For example, 'α' and 'β' digit will embed a watermark by modifying the *hardware assignment* table and 'γ' will embed the vendor watermark by changing the "*non-critical operations* ($\mu_m > 0$)" table. The detailed encoding rule of each digit is defined as follows [20]:

- 'α' = Even operation and odd operation of odd control step will be assigned to hardware of vendor 2 (V2) and vendor 1 (V1) respectively.
- 'β' = Even operation and odd operation of even control step will be assigned to hardware of vendor 1 (V1) and vendor 2 (V2) respectively for.
- 'γ' = Shift an operation into its immediate next control step (cs) if the operation is in non-critical path with highest mobility.

In the dual-phase watermark encoding rule, no portion of the existing IP design requires to be hidden. Instead, the existing scheduling and hardware assignment of the IP design is locally altered to accommodate the signature encoded digits in the form of watermark. More specifically, in the scheduling phase, the operations of the non-critical path are locally shifted based on the encoding rule of γ as explained later in Tables 2 and 4. Next, in the hardware assignment phase the existing vendor assignment (allocation) is modified based on the encoding rule of α and β. As explained later Tables 1 and 3, due to signature digits of α and β, the existing hard-

Table 1 Hardware assignment table (before implanting watermark)

ODD C.S.	Operation no.	1	2	7	8	9	12	13
	Assigned hardware	M2	M1	A1	M1	M2	M2	A2
EVEN C.S.	Operation no.	3	4	5	6	10	11	14
	Assigned hardware	M1	M2	A1	A2	M2	A2	A1

Table 2 Timing table for non-critical operations (before implanting watermark)

Operation no.	2	3	4	6	8	7	10	13
Control step no.	1	2			3		4	5

Table 3 Modified hardware assignment table (after implanting watermark)

ODD C.S.	Operation no.	1	2	7	8	9	12	13
	Allocated hardware	M1	M2	A1	M1	M2	M2	A2
EVEN C.S.	Operation no.	3	4	5	6	10	11	14
	Allocated hardware	M2	M1	A1	A2	M2	A2	A1

Table 4 Modified timing table for non-critical operations (after implanting watermark)

Operation no.	2	3	4	6	7	10	13	8
Control step no.	1	2			3	4	5	

ware assignment before watermark (opn 1 to M2 and opn 2 to M1) is modified to a different hardware assignment (opn 1 to M1 and opn 2 to M2).

In the approach, watermark length can be controlled by varying the vendor signature strength i.e. the number of digits in the watermark embedded in an IP design depends on the size of the signature provided by the vendor. Moreover, the size of the design is also important to ensure all the watermarking constraints corresponding to the chosen signature are embedded. For example, a small size design can not accommodate a large size signature (watermark length). However, a large size design can accommodate various signature strengths. Thus to ensure that successive synthesis do not result in same watermark, signature strength should be accordingly chosen, e.g. for a small size design a very large size signature is undesirable.

The above encoding process is very useful as it enables to covertly insert watermark signature into an IP core during its design process [20]. Since the utility of multi-vendor hardware assignment model during IP core design is obvious, hence it provides a vehicle for the dual-phase watermarking approach to hiddenly embed the watermarking constraints into it. The watermarking constraints based on encoding

thus can be conveniently implemented in a hidden form. Thus an attacker (without the knowledge of encoding rules) will not have any hint where/how the watermark is embedded. The dual-phase watermarking approach does not aim to obscure an IP design to provide security, it only embeds vendor's signature secretly into the design (without disturbing IP functionality and obfuscating any IP design information) to protect the owner, in case of ownership conflict. Thus even if an attacker knows the complete design, its embedded watermark remains invisible. This is because our watermark embedded does not add any extra design component, logic or feature, but only performs local alteration of existing scheduling and hardware assignment, such that it does not hamper IP functionality or appearance. Further, since we employ local hardware re-assignment by exploiting multi-vendor assignment concept and local movement of non-critical path operation with mobility, to embed watermark, thus the dual-phase watermarking approach incurs low design overhead with respect to execution delay and design area. Additionally, the watermark generated through multi-variable encoding satisfies desirable properties such as minimal embedding cost, fault tolerance (as constraints are exclusive and distributed in nature), resiliency against threats (as specified in threat model) and low creation and detection time.

3.3 Process for Embedding Dual-Phase Watermark in IP Design

The following steps are used to embed dual-phase watermark [20]:

1. Schedule the DFG based on **list scheduling** algorithm and user provided hardware configuration.
2. Perform hardware assignment on the schedule DFG.
3. Create the "*hardware assignment*" table for all operation and timing table of "*non-critical operations* ($\mu_m > 0$)" to represent the IP design before embedding watermark.
4. Based on the operation number in each cs. sort the operations in increasing order.
5. Take the signature of the vendor as a combination of 'α', 'β', 'γ' digit's only.
6. For each occurrence of 'γ' digit shift/move an operation of non-critical path by scanning from cs 1 onward (without repeating) such that:
    a. In the immediate next cs. the operation should not has a child.
    b. Shifting/moving of the operation should not violate the user provided resource configuration constraints.
    c. The operation has the maximum mobility value will get priority to resolve conflict (if conflict arises between two or more operations).
7. For each occurrence of 'α' and/or 'β' hardware reassignment is performed in sorted order as per the encoding rules.
8. To represent a dual-phase watermarked IP core design, modify the "*hardware assignment*" table and "*non-critical operations* ($\mu_m > 0$)" table generated in step 3 based on steps 6 and 7.

In the dual-phase watermarking approach we have used classical resource constraint list scheduling algorithm published in traditional HLS papers such as [27–30] where priority function (e.g. mobility based, number of successor operations based) is used to resolve resource contention or conflict among operations. Similar techniques have been used in GAUT tool [31] where authors have used a modified list scheduling using bit width in addition to mobility. Further, contemporary algorithms as soon as possible (ASAP) scheduling has also been used in LegUp tool [32]. Thus the dual-phase watermarking approach uses the realistic list scheduling algorithm adopted from classical sources. Moreover, in watermark embedding process (in [20]), the 1st three steps are pre-watermark stage of the approach, where first step uses list scheduling to generate a schedule (refer to Sect. 5.1 that shows a motivational example of a valid schedule generated using list algorithm). Subsequently step 4 onward watermark embedding process begins where operations are sorted according to their sequence numbers (i.e. name ordering). This sorting is necessary to accommodate the watermark constraints by local alteration of scheduling (only non critical path) and hardware assignment. Therefore, operations are sorted only for inserting signature, not for generating a regular valid scheduling. For the sake of brevity, all the features of authors HLS tools have not been included while demonstrating dual-phase watermarking. However, our HLS methodology/tools (published in [26, 33–36]) combines many features (used in other HLS tools) such as data pipelining, loop pipelining, initiation interval, loop unrolling, loop folding, tree height transformation, logic transformation, redundant operation elimination, loop invariant code motion etc. Thus for preserving succinctness, the dual-phase watermarking method has not been demonstrated using above features.

3.4 Signature Detection

The signature detection for original owner of dual-phase watermarking approach can be achieved in two steps:

1. *Inspection*: The objective of inspection is to collect the relevant information from an IP design (hardware description language (HDL) files) such that the presence of watermark can be identified by only a knowledgeable user (of encoding rules). For instance in the dual-phase watermarking approach inspection can be performed to collect the information of "*non-critical operations* ($\mu_m > 0$)" timing and "*hardware assignment*" details from the source code of hardware description language.
2. *Verification*: The verification of signature is required to validate the presence of vendor signature in the reverse engineered IP design. In order to perform this, vendor's signature needs to be decoded (converted to constraints) using signature encoding rules (mentioned in Sect. 3.2). Finally, the presence of decoded constraints are verified in the reverse engineered design. The design flow of devised signature detection process is presented in Fig. 3.

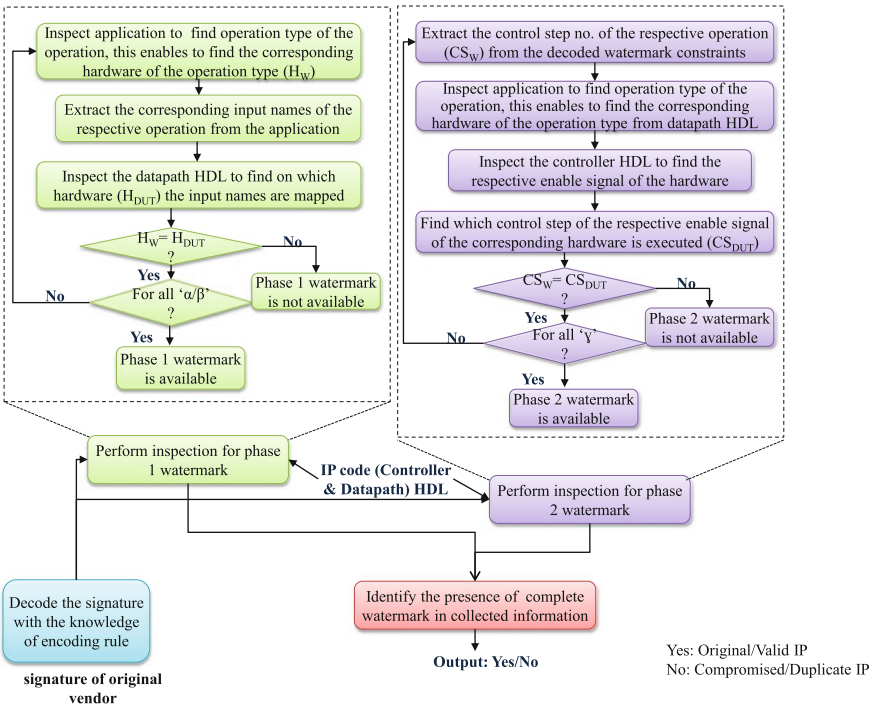

Fig. 3 Signature detection process of dual-phase watermarking approach

4 Motivational Example for Dual-Phase Watermarking Approach

4.1 Motivational Example for Dual-Phase Watermarking Scheme

Figure 4 shows a list scheduled DFG of MESA benchmark based on 2 adders and 2 multipliers which is provided as user input. In control step 1 the number of ready to schedule multiplication operations are 4 however, the number of multipliers available are 2 (according to user resource constraints). In the dual-phase watermarking approach this type of resource conflict is resolved using mobility based list scheduling. As operation 1 and operation 2 has lesser mobility than operation 3 and operation 4 they are scheduled in control step 1. Similarly, for other resource conflicts the operation having lesser mobility gets higher priority to schedule in a control step. Two instances are obtained for each hardware type, by importing one from vendor 1 (V1) other from vendor 2 (V2). The initial (before embedding watermark) IP schedule performs random hardware assignment. The corresponding operation numbers (1–14) of each functional unit appears in the left and the assigned hardware type of

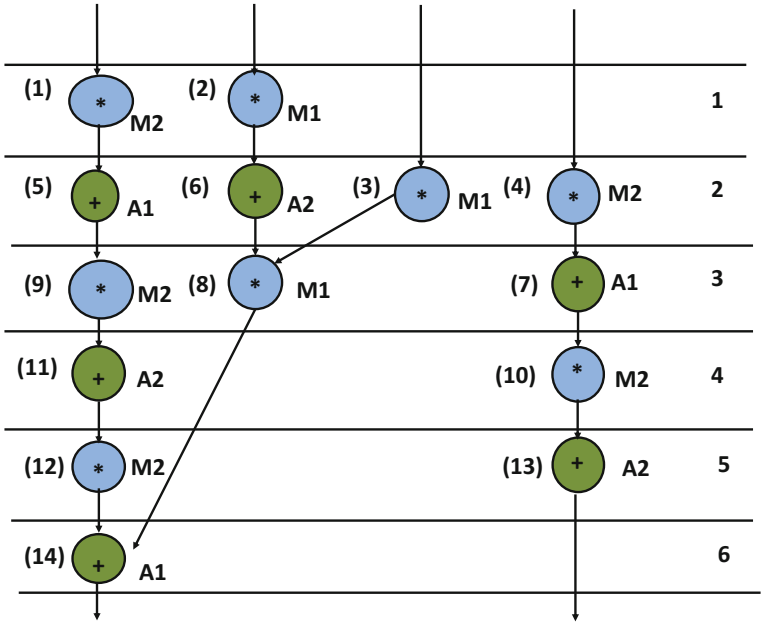

Fig. 4 Scheduled DFG of MESA using 2 adders and 2 multipliers with random hardware assignment before implanting any constraints

each functional unit appears in the right (Fig. 4). For instance, the first operation of cs 1 is marked as operation number (1). 'M2' indicates the multiplier assigned to this operation is imported from vendor type 2. The operations and its corresponding assigned hardware is presented in Table 1. The first and third row of the table represent the operation # of odd cs and even cs respectively and the second and fourth row shows the corresponding assigned hardware. Next step is to create a table containing details of timing information for IP design (before embedding watermark). *Note: We do not include timing info of critical path operations as watermark is not inserted there*. The table consists of list of all non-critical path operations and their respective cs number. The first and second row of the table indicate the operation number and the corresponding cs number respectively. In the table, based on mobility, operations of same cs are reported in sorted order. Next step is selecting a unique vendor signature provided as watermark. Assuming: "$\gamma\gamma\alpha\beta\beta\alpha\alpha$".

The dual-phase watermarking methodology inserts watermarking constraints during scheduling and hardware assignment phases of HLS respectively. According to the encoding rule, a single encoded digit 'γ' shall move an operation of non-critical path with higher mobility (refer Fig. 4) to its immediate next cs. Now as per the rule 6 in Sect. 3.3 operation number 8 is the first eligible operation to move in cs 4 and then subsequently to cs 5 due to occurrence of two consecutive 'γ' digits. The reason is that, other operations of non critical path viz. 2, 3, 4, 6, 7 and 10 does not satisfy the rule 6, while operation number 13 is not eligible because lack of

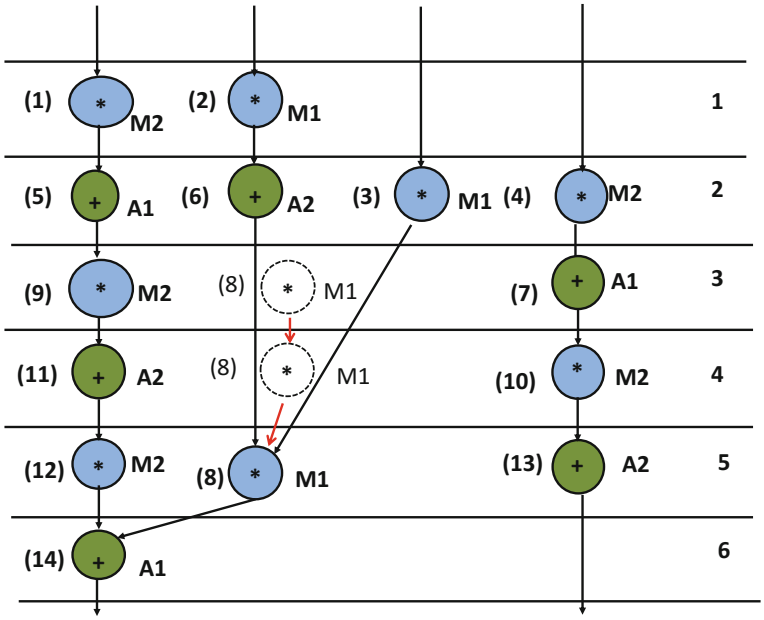

Fig. 5 Scheduled DFG after implanting first-phase of watermark ('γ' digits)

another 'γ' digit in signature chosen. After inserting two 'γ' digits (according to the sample vendor signature) the modified scheduled DFG with first phase watermarking constraints embedded is shown in Fig. 5. The aforesaid scheduled DFG after embedding 'γ' is further used to embeds watermark in the hardware assignment phase to insert 'α' and/or 'β'. Now according to the chosen signature, the 3rd digit of the watermark is α. Therefore, due to this α re-assignment of hardware M1 to operation 1 is performed. Similarly, the 4th digit i.e. β e-assignment of hardware M2 to operation 3 is performed. This process continues for other signature digits. The modified "*hardware assignment*" table and "*non-critical operations* ($\mu_m > 0$)" table after implanting watermark are present in Table 3 and Table 4 respectively. The final dual-phase watermark implanted schedule is presented in Fig. 6. The overhead due to scheduling phase is nil, while hardware assignment phase in minimum. Further, register overhead is nil.

4.2 Properties of Generated Watermark

A watermark embedded in an IP core design must comprise of several desirable properties. The watermarking methodology discussed in this chapter is capable to generate those desirable properties. We discuss these properties achieved below:

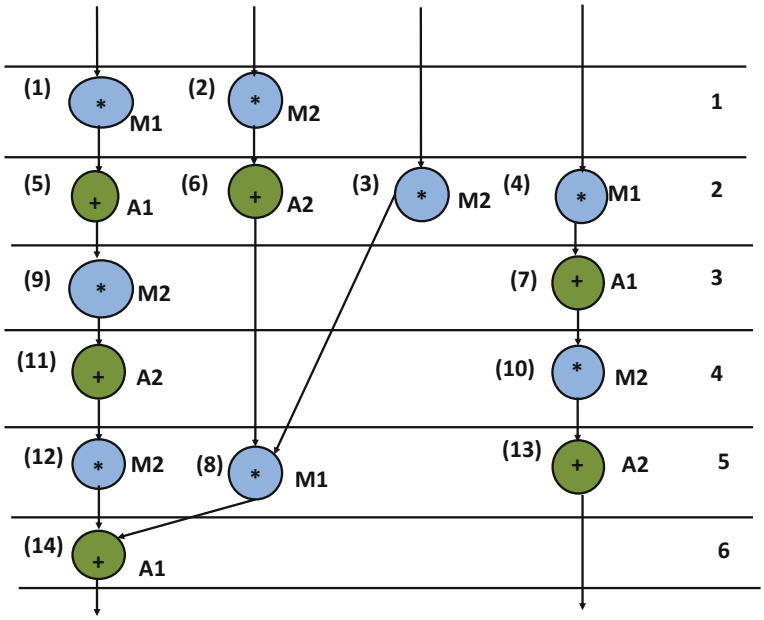

Fig. 6 Scheduled DFG after embedding dual-phase watermark ('α','β','γ' digits)

1. *Low embedding cost*: The watermark generated through dual-phase watermarking scheme incurs low area and latency overhead. Additionally zero register overhead is imposed.
2. *Resiliency against threats*: The watermark is embedded in two different phases using multi-variable encoding scheme. This makes the watermark strong and robust against typical threats such as false claim of ownership, IP counterfeit/forgery and watermark tampering.
3. *Tamper Tolerance*: As the watermark is embedded in two phases and distributed over the complete design, the ownership remains intact even after any tampering or partial removal of watermark (i.e. if the watermarking constraints of phase 1 are removed by an attacker, the ownership can be still proven by the watermarking constraints of phase 2 or vice versa). Additionally, partial removal could also occur if some watermarking constraints of a specific phase is only removed. In such a scenario, the dual-phase watermarking approach is also capable to detect watermark as the remaining watermark constraints are still distributed in the design.
4. *Watermark creation and detection time*: The dual-phase watermarking is fast such that the creation time is very less i.e. in the order of few milli-seconds. Besides, it is simple for a genuine entity (with complete knowledge of encoding rules) to detect the watermark while tough for an outsider.

5 Results and Analysis

In subsequent discussion of this chapter, we term a pre-watermark design (before embedding any watermark constraints) as baseline design. The dual-phase watermarking solution, related approaches [2, 3] and baseline design, all have been implemented in Java and run on AMD processor. Subsequent subsections present the comparative results with [2, 3]. Following are the major reasons for choosing [2, 3] for comparison:

1. Both [2, 3] have integrated watermark in one of the three major phases (i.e. scheduling, hardware and register assignment, binding) of HLS prior to datapath and control unit generation. However, [11] has not integrated watermark in one of these phases, but has embedded in IP control unit. Since the dual-phase watermarking approach has also embeds watermark in two of these major phases of HLS [20] it is more closer to [2, 3] from comparison perspective.
2. Additionally, [2, 3] targets HLS for application specification integration circuits (ASICs), however, [11] targets HLS for field programmable gate array (FPGA). Since the approach also targets HLS for ASICs, it includes similar platform as [2, 3], compared to [11].
3. Finally, the approach in [11] suffers from a limitation that lack of sufficient "temporally free" output slots result into erroneous watermarking process. Whereas, [2, 3] do not suffer from any such type of limitation like the dual-phase watermarking approach.

A 15 nm technology based on NanGate library [37] is used to calculate the design area and execution latency of each design. HLS benchmarks are adopted from [38, 39] has been used in the dual-phase watermarking approach in this chapter.

5.1 Typical Attack Scenarios

In this section of the chapter, we discuss various type of attack scenarios. Assuming, entity 'X' is the owner of a watermarked design (D_w) which entity 'Y' has brought from 'X'. In such a case the following threats may arises [2, 20]:

5.1.1 Unintended Signature Extraction

Entity 'Y' may try to identify an existing information in the watermarked design (D_w) through inspection or reverse engineering and claims as his watermarking constraint. Subsequently, 'Y' can then demand his ownership claiming design contains his signature. In such a case when the design contains both 'X's signature (original) and 'Y's signature (unauthorized) then, the entity having stronger and more meaningful watermark will be considered as the real owner.

5.1.2 Unauthorized Signature Insertion

Entity 'Y' may try to insert his own signature into D_w and claim for ownership. In that scenario, the newly generated watermarked design will contain 'Y's signature on the top of the 'X's signature. In such contradiction, 'X' can establish his authorship easily as 'X's design only contains his signature, whereas, 'Y's design contains signatures of both 'X' and 'Y'.

5.1.3 Tampering Original Signature

Entity 'Y' may try to remove the watermark (partially or fully) from D_w by performing some alterations to create an unauthorized design. However, as dual-phase watermarking methodology distributes additional constraints throughout the complete design in two different phases of HLS, therefore it is extremely difficult to tamper all the watermarking constraints in the design.

5.2 Strength of Protection and Tamper Tolerance Ability

Table 5 presents the probability of coincidence (P_c) of the dual-phase watermarking technique [2, 3]. Probability of coincidence [20] is defined as the possibility of generating the same hardware assignment and scheduling as the one obtained after embedding watermark. The metric for the approach is defined below:

$$P_c = \left(\frac{1}{\prod_{i=1}^{D} N(R_i)}\right)^{w} * \left(\frac{1}{\prod_{j=1}^{G} Mob(O_j)}\right) \tag{1}$$

where w is the total count of α/β digit available in the signature, $N(R_i)$ is number of hardwares of hardware type R_i and D is total types of hardwares. $Mob(O_j)$ indicates the mobility of the non-critical operation j (O_j) and G is the total count of γ digits available in the signature. Probability of coincidence (P_c) for related work [2, 3] is based on information provided in their paper. Table 5 shows the dual-phase watermarking approach have stronger credibility of authorship as lower P_c value is achieved compared to [2, 3]. Further,higher the signature size, stronger the proof of ownership (as P_c decreases).

Table 6 indicates the resilience of the dual-phase watermarking approach, [2, 3] against tampering i.e. tamper tolerance ability. More the watermarking constraints embedded throughout the design, more the difficulty for an attacker to tamper and remove it completely. In [2, 3], watermarking constraints are embedded as extra edges in the colored interval graph. However, in many cases all the watermarking constraints corresponding to the signature strength does not get added as extra edges due to presence of default edges in the design. In other words, number of actual (effective) watermarking constraints added to the design is much lesser than signature strength

Table 5 Comparison of (P_c) as proof of ownership for dual-phase watermarking approach [2, 3] for signature strength: 80 digits

Benchmarks	Probability of coincidence (P_c)		
	Dual-phase	[2]	[3]
2D-ARF	$5.8 * 10^{-76}$	$5.7 * 10^{-73}$	$5.7 * 10^{-73}$
DCT (8-tap)	$1.4 * 10^{-79}$	$5.7 * 10^{-73}$	$5.7 * 10^{-73}$
2D-DWT	$8.9 * 10^{-60}$	$1.2 * 10^{-56}$	$1.2 * 10^{-56}$
EWF	$3.2 * 10^{-60}$	$6.8 * 10^{-49}$	$6.8 * 10^{-49}$
IDCT (8-tap)	$3.3 * 10^{-76}$	$5.7 * 10^{-73}$	$5.7 * 10^{-73}$
MPEG-2 motion vector	$6.2 * 10^{-95}$	$2.0 * 10^{-92}$	$2.0 * 10^{-92}$
1D-JPEG-IDCT (8-tap)	$2.9 * 10^{-89}$	$4.6 * 10^{-87}$	$4.6 * 10^{-87}$
MESA feedback points	$7.3 * 10^{-96}$	$4.6 * 10^{-80}$	$4.6 * 10^{-80}$
MESA interpolate aux	$2.1 * 10^{-116}$	$1.3 * 10^{-101}$	$1.3 * 10^{-101}$
MESA matrix multiplication (4×4)	$1.2 * 10^{-83}$	$1.5 * 10^{-83}$	$1.5 * 10^{-83}$

Table 6 Comparison of tamper tolerance as proof of resilience for dual-phase watermarking approach [2, 3] for signature strength: 80 digits

Benchmark	Actually embedded watermarking constraints			% of actually embedded watermarking constraints		
	Dual-phase	[2]	[3]	Dual-phase	[2]	[3]
2D-ARF	56	40	48	70	50	60
DCT (8-tap)	80	56	64	100	70	80
2D-DWT	51	41	48	64	51	60
EWF	48	27	29	60	34	36
IDCT (8-tap)	80	54	64	100	68	80
MPEG-2 motion vector	80	56	56	100	70	70
1D-JPEG-IDCT (8-tap)	80	55	58	100	69	73
MESA feedback points	80	56	58	100	70	73
MESA interpolate aux	80	50	57	100	63	71
MESA matrix multiplication (4×4)	80	62	66	100	78	83

applied. Thus an attacker has to put much less effort in identifying the signature. On the contrary, for dual-phase watermarking approach, watermarking constraints are embedded in scheduling and assignment phases of the design as local alterations. This technique reduces chances of default constraints being present in the design, thus enabling more watermarking constraints to be effectively added to the design. Thus an attacker has to put much higher effort in identifying the signature by reverse engineering. Therefore, the tamper tolerance ability of the dual-phase watermarking approach is higher than [2, 3].

Table 7 Comparison of dual-phase watermarking approach with baseline with respect to area, latency, embedding cost and cost overhead percentage (%)

Benchmarks	Resource configuration	Area (μm^2)		Latency (ns)		Cost		Cost overhead %
		Baseline	Dual-phase	Baseline	Dual-phase	Baseline	Dual-phase	Dual-phase
ARF	5(+), 3(*)	191.10	209.19	2.67	3.11	0.77	0.87	12.98
DCT	6(+), 3(*)	250.87	263.45	3.95	4.19	0.80	0.84	5
DWT	2(+), 4(*)	162.79	165.94	1.98	2.08	0.78	0.81	3.85
EWF	3(+), 2(*)	184.81	197.39	3.24	3.82	0.85	0.95	11.76
IDCT	5(+), 3(*)	246.15	253.23	3.77	4.16	0.78	0.83	6.41
MPEG	3(+), 8(*)	280.76	287.05	2.44	2.59	0.73	0.76	4.11
JPEG	5(+), 5(*)	747.90	756.55	14.90	15.92	0.72	0.76	5.56
MESA-FP	4(+), 7(*)	370.41	380.63	4.88	4.94	0.71	0.74	4.23
MESA-IA	8(+), 8(*)	644.87	667.68	9.24	9.62	0.65	0.68	4.62
MESA-MM	4(+), 4(*)	526.12	534.77	9.52	9.96	0.71	0.73	2.82

5.3 Embedding Cost Comparison and Design Overhead Analysis

This sub-section discusses the embedding cost evaluation and design overhead analysis. Embedding cost is calculated in terms of area and latency of a watermarked design. Design overhead analysis is performed in terms of design area, design latency and embedding cost.The embedding cost is evaluated based on the following function [3]:

$$C_f(X_i) = w_1 L_T / L_{max} + w_2 A_T / A_{max} \tag{2}$$

where, $C_f(X_i)$ is the embedding cost of the solution for hardware configuration X_i, L_T and A_T indicates total execution latency and total hardware area of the watermarked design. L_{max} and A_{max} indicates maximum execution latency and hardware area, w_1 and w_2 are user specified weight factor,both of them is set as 0.5 to provide equal weightage.

The comparison of dual-phase watermarking approach with the baseline design in terms of area, latency and embedding cost is reported in Table 7. Further embedding cost overhead percentage (%) of the dual-phase watermarking approach compared to baseline design is reflected in the last column of the aforesaid table. As evident from aforesaid table the dual-phase watermarking approach incurs minimal overhead compare to a baseline design (no watermark).

Table 8 reports the comparative analysis between [2, 3] and the dual-phase watermarking approach in terms of design latency, design area and embedding cost of a watermarked design. For most of the benchmarks reductions in design parameters such as area, latency and embedding cost is achieved with respect to [2, 3]. Table 8 also reports the reduction % achieved by the dual-phase watermarking

Table 8 Comparison of dual-phase watermarking approach with [2, 3] with respect to (w.r.t) area, latency, embedding cost and reduction percentage (%)

Benchmarks	Area (μm^2)			Area reduction (%)		Latency (ns)			Latency reduction (%)		Cost			Cost reduction (%)	
	[2]	[3]	Dual-phase	W.r.t [2]	W.r.t. [3]	[2]	[3]	Dual-phase	W.r.t. [2]	W.r.t. [3]	[2]	[3]	Dual-phase	W.r.t. [2]	W.r.t. [3]
ARF	225.71	223.35	209.19	7.32	6.34	3.11	3.11	3.11	0	0	0.92	0.90	0.87	5.44	3.33
DCT	290.98	288.62	263.45	9.46	8.72	4.51	4.51	4.19	7.09	7.09	0.94	0.92	0.84	10.64	8.70
DWT	182.37	180.01	165.94	9.01	7.82	2.43	2.43	2.08	14.40	14.40	0.93	0.92	0.81	12.90	11.96
EWF	209.19	204.47	197.39	5.64	3.85	3.89	3.89	3.82	1.80	1.80	0.99	0.98	0.95	4.04	3.06
IDCT	280.96	278.40	253.23	9.87	9.04	4.34	4.34	4.16	4.14	4.14	0.91	0.89	0.83	8.79	6.74
MPEG	309.85	309.85	287.05	7.36	7.36	2.77	2.77	2.59	6.50	6.50	0.81	0.81	0.76	6.17	6.17
JPEG	783.29	783.29	756.55	3.41	3.41	16.52	16.52	15.92	3.63	3.63	0.79	0.79	0.76	3.80	3.80
MESA-FP	403.44	403.44	380.63	5.65	5.65	4.95	4.95	4.94	0.20	0.20	0.77	0.77	0.74	3.90	3.90
MESA-IA	701.50	701.50	667.68	4.82	4.82	9.74	9.74	9.62	1.23	1.23	0.70	0.70	0.68	2.86	2.86
MESA-MM	554.44	554.44	534.77	3.55	3.55	10.13	10.13	9.96	1.68	1.68	0.75	0.75	0.73	2.67	2.67

Table 9 Comparison of storage hardware between baseline, dual-phase watermarking approach [2, 3] (for 80 digits watermark size)

Benchmarks	Resource configuration	# of storage hardware			
		Baseline	Dual-phase	[2]	[3]
ARF	5(+), 3(*)	8	8	9	8
DCT	6(+), 3(*)	8	8	9	8
DWT	2(+), 4(*)	5	5	7	6
EWF	3(+), 2(*)	4	4	6	5
IDCT	5(+), 3(*)	8	8	10	9
MPEG	3(+), 8(*)	14	14	14	14
JPEG	5(+), 5(*)	12	12	12	12
MESA-FP	4(+), 7(*)	21	21	21	21
MESA-IA	8(+), 8(*)	48	48	48	48
MESA-MM	4(+), 4(*)	24	24	24	24

approach compared to [2, 3] with respect to latency, area and implementation cost. Firstly, reductions compared to [2, 3] have been achieved because both approaches embed the complete signature sequence during register assignment phase of HLS (resulting into higher chances of register overhead incurring more area/latency). Secondly, both aforesaid approaches do not leverage upon the promising concept of multi-vendor model (with different area/delay values) for hardware assignment. Using a single vendor for hardware assignment throughout the design may result in higher design area/delay.

Table 9 reports the number of register required for baseline, dual-phase watermarking approach [2, 3]. It is observed that register overhead of the dual-phase watermarking approach is **zero** for all the tested benchmarks compared to [2, 3], thus demonstrating that the dual-phase watermarking approach requires lower registers than [2, 3].

6 Conclusion

A dual-phase (embedded in same design abstraction level) watermarking methodology during high level synthesis has been presented in this chapter. The aforesaid approach is based on a new embedding scheme (that implants watermark during scheduling and assignment without change in functionality) and signature encoding mechanism. The dual-phase watermarking approach does not add/reduce any new design component while inserting watermark. The dual-phase watermarking methodology achieved an average reduction of design area, design latency and embedding cost of 7% (min = 3.41% and max = 9.87%), 4% (min = 0% and max = 14.40%) and

6% (min = 2.67% and max = 12.90%) respectively compared to two similar [2, 3] HLS based watermarking approaches. Further, stronger proof of authorship is also achieved compared to [2, 3]. In addition, the dual-phase watermarking approach is more robust, scalable, tamper tolerant and easily adaptable to any modern CAD tool.

One of the important aspects of future research could be geared towards development of stronger encoding mechanisms that would encompass more variables to increase tamper tolerance capability. Moreover, development of multi-phase watermark could be an important problem to investigate for the future that could increase the robustness of author credibility further. Additionally, another aspect of improvement could be refining the signature strength such that an optimal balance between strong tamper tolerance and low embedding time could be achieved. Finally, impact of watermark creation time on different type of applications such loop based, non-loop based and nested loop based data flow graphs could be analyzed in future research.

Acknowledgements Under grant no. 22/730/17/EMR-II CSIR has supported this work financially. The authors would like to thank CSIR for the same.

References

1. Castillo, E., Meyer-Baese, U., Garca, A., et. al.: IPP@HDL: efficient intellectual property protection scheme for IP cores. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **15**(5), 578–591 (2007)
2. Koushanfar, F., Hong, I., Potkonjak, M.: Behavioral synthesis techniques for intellectual property protection. ACM Trans. Des. Autom. Electron. Syst. **10**(3), 523–545 (2005)
3. Sengupta, A., Bhadauria, S.: Exploring low cost optimal watermark for reusable IP cores during high level synthesis. IEEE Access J. **4**(99), 2198–2215 (2016)
4. Fernandez, M., Soriano, M., Cotrina, J.: Tracing illegal redistribution using errors-and-erasures and side information decoding algorithms. IET Inf. Secur. **1**(2), 83–90 (2007)
5. Yuan, L., Qu, G., Ghouti, L., et. al.: VLSI design IP protection: solutions, new challenges, and opportunities. In: Proceedings of the 1st NASA/ESA Conference on Adaptive Hardware and System (AHS), pp. 469–476 (2006)
6. Abdel-Hamid, A.T., Tahar, S., Aboulhamid, E.M.: A public-key watermarking technique for IP designs. In: Proceedings of the Conference on Design, Automation and Test in Europe, vol. 1, pp. 330–335 (2005)
7. Roy, J.A., Koushanfar, F., Markov, I.L.: EPIC: ending piracy of integrated circuits. In: Proceedings of the Design, Automation and Test in Europe (DATE), pp. 1069–1074 (2008)
8. Yu, T., Zhu, Y.: A new watermarking method for soft IP protection. In: Proceedings of the International Conference on Consumer Electronics, Communications and Networks (CECNet), pp. 3839–3842 (2011)
9. Nie, T., Zhou, L., Li, Y.: Hierarchical watermarking method for FPGA IP protection. IETE Tech. Rev. **30**(5), 367–374 (2013)
10. Ziener, D., Teich, J.: Power signature watermarking of IP cores for FPGAs. J. Signal Process. Syst. **51**(1), 123–136 (2008)
11. Le Gal, B., Bossuet, L.: Automatic low-cost IP watermarking technique based on output mark insertions. Des. Autom. Embed. Syst. **16**(2), 71–92 (2012)
12. Wu, Y.-T., Shih, F.Y.: Digital watermarking based on chaotic map and reference register. Pattern Recognit. **40**(12), 3753–3763 (2007)

13. Alkabani, Y., Koushanfar, F., Potkonjak, M.: Remote activation of ICs for piracy prevention and digital right management. In: Proceedings of the IEEE/ACM International Conference on Computer-Aided Design, pp. 674–677 (2007)
14. Ni, Z., Shi, Y.-Q., Ansari, N., Su, W.: Reversible data hiding. IEEE Trans. Circuits Syst. Video Technol. **16**(3), 354–362 (2006)
15. Marvel, L.M.: Information hiding: steganography and watermarking. In: Javidi, B. (ed.) Optical and digital techniques for information security. Advanced sciences and technologies for security applications, vol. 1, pp. 113–133. Springer, New York, NY, USA (2005)
16. Cox, I.J., Miller, M.L., Bloom, J.A., Fridrich, J., Kalker, T.: Digital Watermarking and Steganography. Morgan Kaufmann, San Mateo, CA, USA (2007)
17. Kufel, J., Wilson, P.R., Hill, S., et. al.: Sequence-aware watermark design for soft IP embedded processors. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **24**(1), 276–289 (2016)
18. Jain, A.K., Yuan, L., Qu, G., et. al.: Zero overhead watermarking technique for FPGA designs. In: Proceedings of the 13th ACM Great Lakes Symposium on VLSI, pp. 147–152 (2003)
19. Cui, A., Qu, G., Zhang, Y.: Ultra-low overhead dynamic watermarking on scan design for hard IP protection. IEEE Trans. Inf. Forensics Secur. **10**(11), 2298–2313 (2015)
20. Sengupta, A., Roy, D., Mohanty, S.P.: Triple-phase watermarking for reusable IP core protection during architecture synthesis. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. pp. 1–1 (2017). https://doi.org/10.1109/TCAD.2017.2729341
21. Rashid, A., Asher, J., Mangione-Smith, W. H., et. al.: Hierarchical watermarking for protection of DSP filter cores. In: Proceedings of the IEEE 1999 Custom Integrated Circuits Conference (Cat. No. 99CH36327), San Diego, CA, pp. 39–42 (1999)
22. Charbon, E.: Hierarchical watermarking in IC design. In: Proceedings of the IEEE 1998 Custom Integrated Circuits Conference (Cat. No. 98CH36143), Santa Clara, CA, pp. 295–298 (1998)
23. Cui, A., Chang, C. H., Zhang, L.: A hybrid watermarking scheme for sequential functions. In: IEEE International Symposium of Circuits and Systems (ISCAS), Rio de Janeiro, pp. 2333–2336 (2011)
24. Rajendran, J., Zhang, H., Sinanoglu, O., et. al.: High-level synthesis for security and trust. In: Proceedings of the IEEE 19th International On-Line Testing Symposium (IOLTS), pp. 232–233 (2013)
25. Karri, R., Rajendran, J., Rosenfeld, K., et. al.: Trustworthy hardware: identifying and classifying hardware trojans. Computer **43**(10), 39–46 (2010)
26. Sengupta, A., Bhadauria, S., Mohanty, S.P.: TL-HLS: methodology for low cost hardware trojan security aware scheduling with optimal loop unrolling factor during high level synthesis. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **36**(4), 655–668 (2017)
27. Thomas, D.E., Lahnese, E.D., Walker, R.A., et. al.: Algorithmic and Register-Transfer Level Synthesis: The System Architects? Workbench. Kluwer Academic Publisher (1990)
28. Heijligers, M.J.M., Cluitmans, L.J.M., Jess, J.A.G.: High-level synthesis scheduling and allocation using genetic algorithms. In: Proceedings of the Asia and South Pacific Design Automation Conference (ASP-DAC '95). ACM, New York, NY, USA (1995)
29. Zoltan, B.: Scheduling algorithms for high-level synthesis. ACAM Sci. J. **5**(1–2), 48–57 (1996)
30. Hwang, C.T., Lee, J.H., Hsu, Y.C.: A formal approach to the scheduling problem in high level synthesis. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **10**(42), 464–475 (1991)
31. Coussy, P., Chavet, C., Bomel, P., et. al.: 'GAUT: a high-level synthesis tool for DSP applications' in High-Level Synthesis: From Algorithm to Digital Circuit. Springer, Germany, Heidelberg (2008)
32. Canis, A., Choi, J., Aldham, M., et. al.: LegUp: an open-source high-level synthesis tool for FPGA-based processor/accelerator systems. ACM Trans. Embed. Comput. Syst. **13**(2), 1–27 (2013)
33. Sengupta, A.: Exploration of kc-cycle transient fault secured datapath and loop unrolling factor for control data flow graphs during high level synthesis. IET Electron. Lett. **51**(7), 562–564 (2015)
34. Sengupta, A., Roy, D.: Protecting an intellectual property core during architectural synthesis using high-level transformation based obfuscation. IET Electron. Lett. (2017). https://doi.org/10.1049/el.2017.1329

35. Mishra, V.K., Sengupta, A.: Swarm inspired exploration of architecture and unrolling factors for nested loop based application in architectural synthesis. IET Electron. Lett. **51**(2), 157–159 (2015)
36. Sengupta, A., Mohanty, S.P.: High-level synthesis of digital circuits in the nanoscale, mobile electronics era. In: IET Book: Nano-CMOS and Post-CMOS Electronics: Circuits and Design, pp: 219–261 (2016). e-ISBN: 9781785610004
37. NanGate 15 nm open library (2016). http://www.nangate.com/?page_id=2328
38. Express Benchmarks (2016). http://www.ece.ucsb.edu/EXPRESS/benchmark/
39. Mohanty, S.P., et. al.: Low-Power High-Level Synthesis for Nanoscale CMOS Circuits. Springer Science+Business Media, LLC (2008). https://doi.org/10.1007/978-0-387-76474-0