Triple-Phase Watermarking for Reusable IP Core Protection during Architecture Synthesis

Anirban Sengupta, *Member IEEE,* Dipanjan Roy, *Student Member IEEE*, and Saraju P. Mohanty, *Senior Member IEEE*

***Abstract*— Reusable Intellectual property (IP) cores used in the consumer electronic devices, representing years of valuable investment, need protection against threats such as piracy and illegal claim of ownership. This paper introduces a novel 7-variable signature encoding driven triple-phase watermarking methodology during high level synthesis (HLS)/architectural synthesis for IP core protection of vendor rights. The proposed approach is extremely robust against external threats as it involves vendor signature comprising of 7-variable combination embedded through three independent phases of high level synthesis. This research is the first work in the HLS literature that presents a triple-phase watermarking process during HLS compared to single phase watermarking techniques so far. The proposed approach incurs zero delay overhead and minimal hardware overhead while embedding as well as yields average cost reductions of 7.38 % and 6.25 % compared to two similar approaches. Further, the proposed triple-phase watermark approach achieves a lower P_c value by ~$3.2*10^{27}$ times in magnitude compared to similar approaches. Additionally, the proposed approach is $3.4*10^{43}$ and $2.8*10^{19}$ times more tamper tolerant than similar approaches.**

***Index Terms*— IP core, Watermark, High level Synthesis, Hardware Protection, Triple phase**

I. Introduction

Consumer electronics (CE) along with information and communication technology make emerging smart cities a reality [1]. In such a CE based framework, security and protection of its' intellectual property (IP) cores are considered as major challenges. Thus, the use of secured IPs is of paramount importance. In the era of smart cities, swelling CE hardware design complexity is out-striding the designer productivity, ensuing into greater endeavours. Further, the current generation of CE design process is massively dependent on global IP supply chains. To maximize design productivity and minimize design time the use of IP cores, often delivered by a third party vendor, has become a de-facto customary in the industry. However, there are rising threats to security and surging piracy issues that threaten global supply chains as CE system-on-chip (SoC) design becomes increasingly commoditised [1]-[3]. Fig.1. shows a thematic representation of a secured IP core in a SoC used in a CE device. The theme denotes the need for protection of modern CE devices against piracy/ownership abuse.

With the surge in globalization hardware design and manufacturing process and rivalry between the IP vendors,

A. Sengupta (asengupta@iiti.ac.in) and D. Roy are with the Computer Science and Engineering at Indian Institute of Technology Indore, India. S. P. Mohanty (saraju.mohanty@unt.edu) is with the Computer Science and Engineering, University of North Texas, Denton, Texas, USA.

Fig. 1. IP Protection of Consumer Electronics Hardware.

threats such as IP piracy/counterfeiting, false claim of ownership are intensifying. As a consequence, the requirements for protection of IP-core designs and the know-how they epitomize has become of prominence to industry [1]-[5]. As a reusable IP core represents many man-years of design, research and verification testing, a key question is how to protect this investment. It is well acknowledged that the rights of the original IP owner can be abused both deliberately and inadvertently. For example, direct piracy where duplicitous means or reverse engineering may enable direct theft/copying of the IP for re-use without authorization. As a result of counterfeiting the IP, the adversary may even claim the IP to be their own. Thus process of nullifying false claims of IP ownership is obligatory. A typical vendor's signature based IP core protection mechanism for CE devices is shown in Fig.2. In this figure, to safeguard against potential threats (attackers), vendor signature is embedded in the IP core of CE hardware during this design process in the vendor house itself. These secured/protected CE hardware are used in consumer devices.

For a rightful owner of an IP core, it is both challenging and exclusive to prove that their IP is being used illegally in a

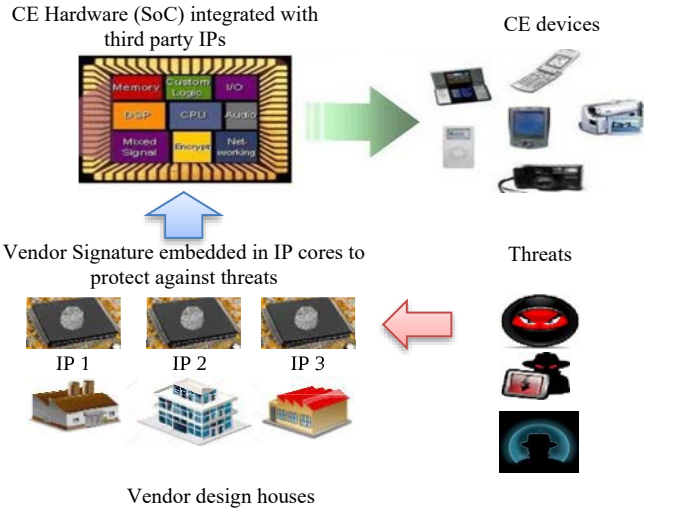

Fig. 2. IP Protection for Secured Hardware in Consumer Electronics in the Global Supply Chain Framework

product. Additionally, compromise of functionality and quality is strictly forbidden while deploying protective measures in an IP core [2, 3]. One of the effective possibilities may be embedding a vendor (third party IP provider) watermark to protect its IP core against false claim of ownership and IP infringements [6]-[14]. For example, in [6] IP protection is performed by locking and enabling each working IP remotely by the IP designers. Further, a reversible data hiding approach is proposed in [7]. Moreover, authors in [8] and [9] have proposed information hiding techniques through watermarking and steganography. Chaotic map based digital watermarking scheme is proposed in [10]. Authors in [11] propose a watermarking approach for multimedia applications by embedding additional data. The research in the last few years have been on IP protection to secure its value, but only few major security mechanisms have been proposed and employed so far during architecture synthesis. There exist other IP protection (active & passive) techniques with different objectives such as IP fingerprinting useful to protect the user rights [12], logic locking of IP to secure it via keys [13], controlling multiple IPs remotely by the IP owner [14], IP metering useful for detecting illegal copies [17], obfuscation [18] useful for preventing reverse engineering (thereby thwarting third party attacks in the supply chain). Obfuscation hides the structure/function of the design thus making reverse engineering process extremely difficult. However, obfuscation may not easily nullify ownership conflict between a genuine seller and a genuine buyer i.e. protects an IP core from a dishonest genuine buyer who claims false ownership. For example, when a genuine buyer purchases an IP from a genuine seller then the IP design cannot be delivered as functionally obfuscated. In such a scenario should a buyer make a false claim of ownership of the IP then it may not be protected. Neither structural nor functional obfuscation can nullify genuine buyer's false claim of ownership. In such a case, IP watermark is useful.

The rest of the paper is organized as follows: Section II summarizes the novel contributions. Section III discusses the proposed watermarking methodology. Section IV provides a motivational example. Possible threat scenarios are discussed in Section V. Experimental results are presented in Section VI, followed by conclusion in Section VII.

II. Contributions of This Paper to State of the Art

A. *Novel contributions of the paper*

The **novel contributions of this paper** in terms of improving the state-of-art are as follows:

- Proposes a novel triple-phase watermarking methodology to protect the reusable IP core during HLS.
- Proposes a novel highly robust 7-variable signature encoding scheme for embedding watermark during consecutive scheduling phase, hardware allocation phase and register allocation phase of HLS.
- Yields lower cost overhead in terms of hardware and latency compared to state of the art [4] [5].

B. *Motivation: Embedding a watermark at high level*

Embedding watermark in higher abstraction (architectural level) is more beneficial compared to lower design abstraction during design process, as watermark embedded at higher abstraction protects the design in subsequent lower levels (as watermark constraint propagates with synthesis) as well as incurs less design overhead and implementation complexity [20, 21, 22].

C. *Prior related approaches*

At lower abstraction level such as Gate level, watermarking has been proposed [15]-[16]. For instance in [15], the netlist and bit stream of an IP design were used to insert a vendor watermark. Moreover, in [16] in-synthesis IP watermarking scheme has been proposed. Additionally, at higher abstraction level such as architectural level, watermarking has also been embedded. For instance, [4] and [5] implant watermark in register allocation phase of architectural synthesis. More explicitly, authors in [4] employ binary encoding rules (combination of two variables) for signature encoding of watermark whereas authors in [5] present a more secured multi variable (combination of four variables) encoding scheme for IP watermarking. The proposed work provides greater protection than the aforesaid approaches in terms of lower probability of co-incidence (due to triple-phase watermark) and greater tamper resistance (due to 7 variable encoding) from an attacker's perspective. Additionally, the proposed approach provides protection at lower overhead than similar approaches.

This has been proved in details in experimental results later. However, a brief summary is provided below:

(a) It is well acknowledged that inserting watermark in a design may incur design overhead in terms of area and latency, however the proposed approach reduces the average cost overhead by ~7.38% and ~ 6.25 % compared to [4] and [5], respectively.

(b) The strength (robustness) of a watermark is indicated using probability of coincidence (P_c) metric. Lower the P_c

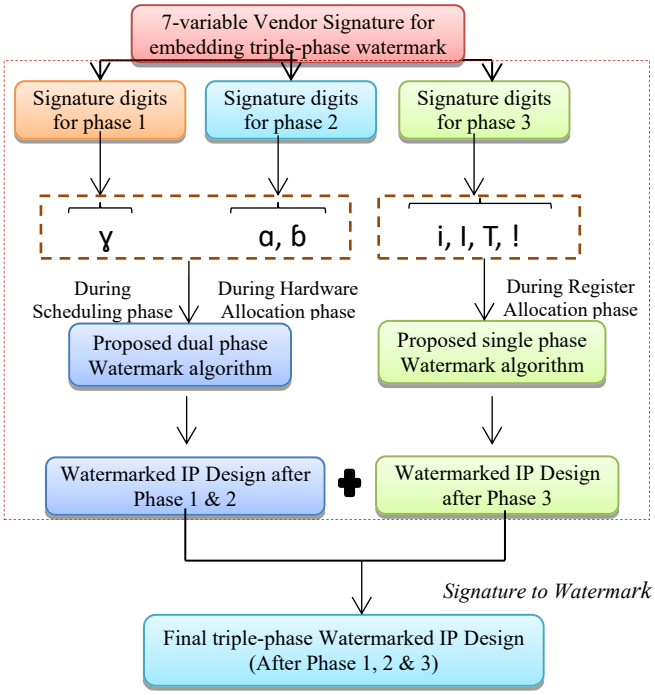

Fig.3. Proposed triple-phase watermark at architecture level

value higher the watermark strength. The proposed triple-phase watermark approach achieves a lower P_c value by ~$3.2*10^{27}$ times in magnitude compared to [4] and [5].

(c) Further, the signature of the proposed watermark approach is a combination of 7 encoding variables while a combination of 2 and 4 encoding variable for [4] and [5] respectively. Therefore the proposed approach is $3.4*10^{43}$ and $2.8*10^{19}$ times more tamper tolerant than [4] and [5] respectively (considering signature strength = 80 digits).

III. PROPOSED WATERMARKING METHODOLOGY

A. *Problem Formulation*

Input/Output: *Inputs*: (a) data flow graph (DFG) and (b) user specified hardware resources set (X) = N (R_1), N (R_2),…N (R_D), where N (R_D) is the number of hardware type R_D; *Output*: Watermarked IP core design.

Threat Model: This paper targets **vendor protection** of reusable IP core from **false claim of ownership.**

Target Platform: The watermarking approach proposed in this paper can be seamlessly adapted to any EDA tool. Hardware description language used for IP can easily be amalgamated with proposed approach in design tools.

B. *Proposed Watermark Encoding*

The diagrammatic depiction of the proposed approach is shown in Fig.3. The proposed method does not handle designs that use non-HDL IPs (such as standard blocks/symbols available in the in-built library of a logic CAD tool). The proposed methodology is applicable for ownership protection of a macro reusable IP core (HDL based). It is assumed that micro-IPs from two different third party vendors are available for implementation of a macro IP core through proposed approach.

The proposed triple-phase watermark is embedded in consecutive scheduling, hardware allocation and register allocation phases of high level synthesis. Besides, triple-phase embedding, the vendor signature is a 7-variable encoding that makes the watermark extremely robust with minimal chances of any malicious alteration. Triple phase and 7-variable encoding enhance protection strength and increase tamper-tolerance ability. Further, it is extremely difficult for an attacker to identify which HLS phases (and how watermark constraints) are embedded in the design.

Additionally, the 3rd phase watermark is independent of both 1st and 2nd phase watermarks. Similarly, 1st phase is independent of both 2nd and 3rd phase watermarks. These advantages make the proposed watermarking method extremely robust and tamper tolerant against threats related to false claim of ownership (details provided in Section V). However proposed watermark has a limitation: In proposed approach, 2nd phase watermark is dependent on 1st phase watermark, therefore tampering of 1st phase watermark may affect 2nd phase watermark constraints. Nevertheless, as discussed before it is considered extremely difficult in which step 1st phase watermark is inserted along with its encoding rule. Moreover, 3rd phase watermark being fully independent is capable to determine the real owner, despite possible tampering in 1st phase. Tampering in 3rd phase alone is also possible however, since 3rd phase is independent of 1st and 2nd phase watermark, therefore 1st and 2nd phase watermark also enables independent protection of original IP owner (as 1st and 2nd phase watermark contains constraints remain un-tampered in the design).

In the proposed approach, an IP design in terms of scheduling, allocation and register allocation phases are represented through the following proposed tables (a)

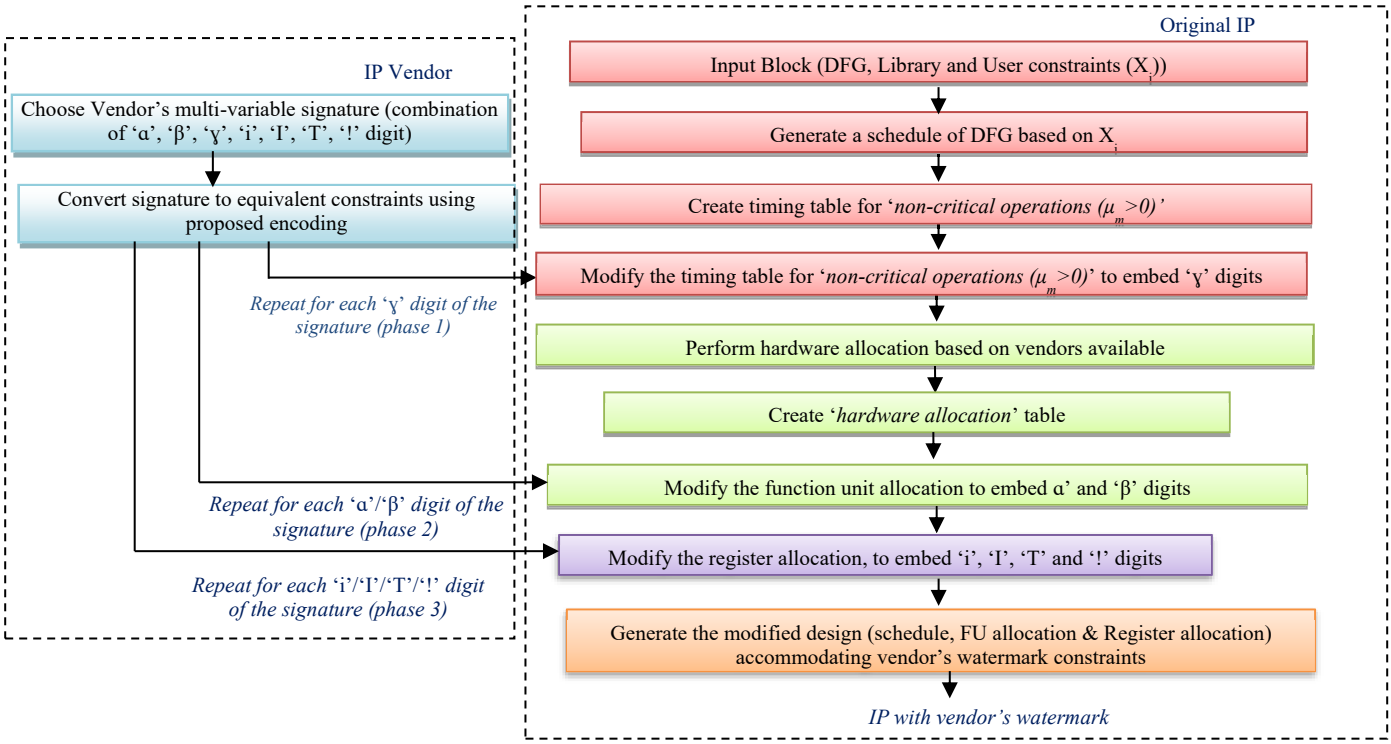

Fig.4. Proposed high level synthesis flow for reusable IP core protection using triple-phase watermark

"*Functional unit allocation*" table, (b) "*Non-critical operations (μ_m > 0)*" timing table, where μ_m denotes the mobility of the operation (c) "*Register allocation" table (inspired from [5])*. The proposed watermarking methodology consists of seven different variables viz. 'α', 'ϐ', 'ɣ', 'i', 'I', 'T', '!' where 'ɣ' digit embeds the vendor 1st phase watermark by modifying the "*non-critical operations (μ_m > 0)*" table, 'α' and 'ϐ' digits embeds the 2nd phase vendor watermark by modifying the functional unit allocation table and finally, 'i', 'I', 'T', '!' digits embeds the 3rd phase vendor by modifying the *"register allocation"* table respectively.

The encoding rules of all seven signature digits are defined as follows:

- 'α' = *For odd control step:* odd operation will be assigned to hardware of vendor type 1 (U1) and even operation will be assigned to hardware of vendor type 2 (U2).
- 'ϐ' = *For even control step:* odd operation is assigned to hardware of vendor type 2 (U2) and even operation is assigned to hardware of vendor type 1 (U1).
- 'ɣ' = Move an operation of non-critical path with highest mobility into immediate next control step (cs).
- 'i'= encoded value of edge with node pair as (prime, prime)
- 'I' = encoded value of edge with node pair as (even, even)
- 'T' = encoded value of edge with node pair as (odd, even)
- '!' =encoded value of edge with node pair as (0, any integer)

C. *Algorithm for Embedding Watermark during IP Design*

The diagrammatic depiction of the triple-phase watermark embedding algorithm is shown in Fig.4. The proposed approach accepts user specified hardware components (e.g. # adders, #multipliers etc.) as input. Thus the proposed method does not affect component allocation during watermark insertion.

To insert the proposed watermark, the following algorithm is followed:

1) **Pre-embedding steps (1 – 5):** Based on user provided hardware resources, schedule the DFG.
2) Perform functional unit allocation based on user provided hardware.
3) To represent an IP design before embedding watermark, generate a '*Functional unit (FU) allocation*" table for all operations and a "*non-critical operations (μ_m > 0)*" timing table.
4) Sort the operations based on their number in increasing order in each control step.
5) Select a 7- variable vendor signature in the form of any combination of 'α', 'ϐ', 'ɣ', 'i', 'I', 'T', '!' digits.
6) **Embedding 1st phase watermark (step 6)**: Move/shift an operation of non-critical path by scanning from control step 1 onward (without repeating) for each occurrence of 'ɣ' such that:
    a. It has no child operation in immediate next control step.
    b. Shifting does not violate the hardware constraints.
    c. It has the highest mobility (if conflict occurs between more than one operation)
7) **Embedding 2nd phase watermark (step 7):** Functional unit re-allocation is performed in the scheduling as per the encoding rules for each occurrence of 'α' and/or 'ϐ' (Note: *encoding rule is applied on sorted operations in step 4*).

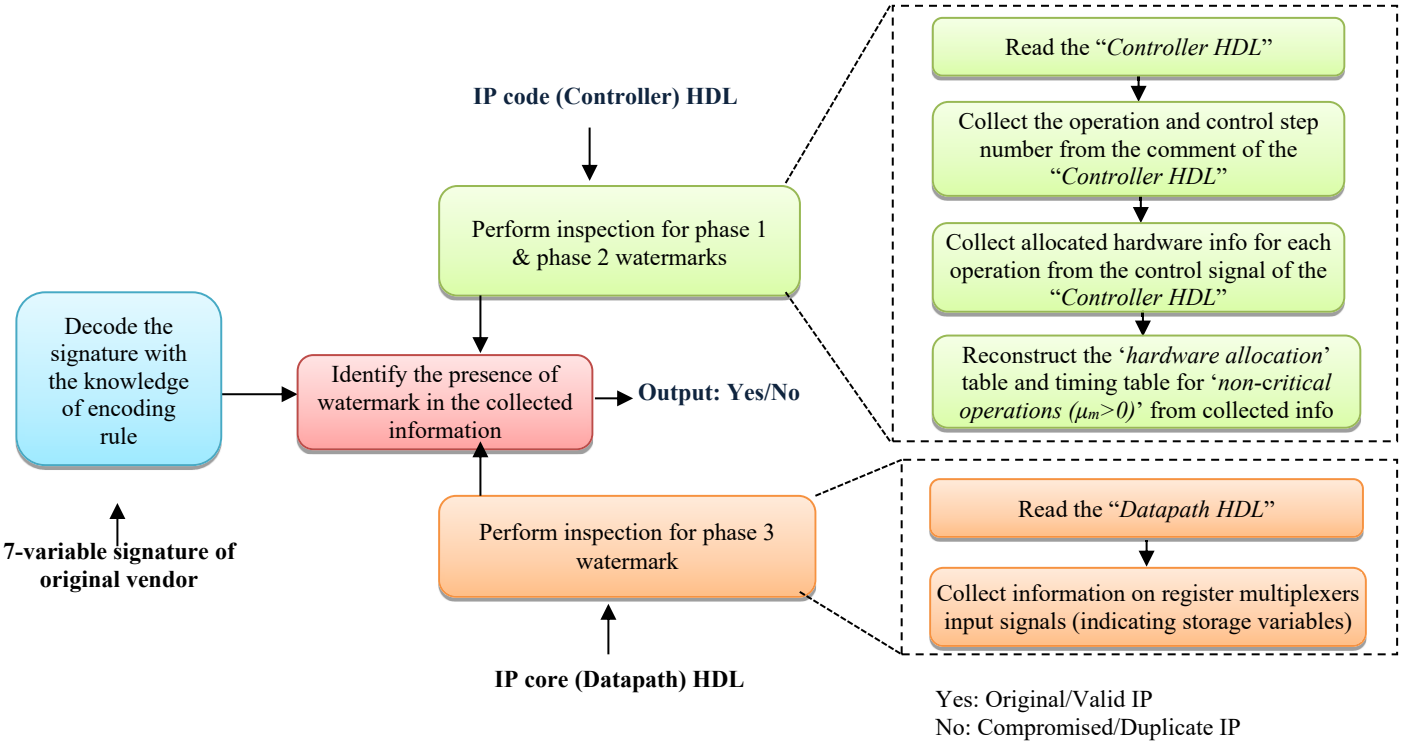

Fig.5. Signature detection process

8) Modify "*hardware allocation*" table and "*non-critical operations ($\mu_m > 0$)*" table for each encoded digit based on step 6 & 7 to represent a watermarked IP design.
9) **Embedding 3rd phase watermark (step 9 - 16):** Assign storage variables in the double phased watermarked schedule (obtained in step 7).
10) Create a coloured interval graph to find the minimum number of registers required for register allocation.
11) Create a *'register allocation'* table for the double phased watermarked scheduling obtained till step 7.
12) Sort storage variables as per their number in increasing order.
13) Feed the 3rd phase vendor signature in the form of i, I, T, !, in which the characters hold the encoded value of additional edges to be inserted.
14) Create a list of additional edge pairs corresponding to its encoded values by traversing the sorted nodes.
15) Insert the 3rd phase watermarking constraints in the coloured interval graph.
16) Modify the "*register allocation*" table representing IP design based on coloured interval graph in last step.

D. *Signature Detection*

Signature detection is a compulsory step when using watermark for resolving vendor ownerships conflicts. In our proposed approach, signature detection is a two-step process, where the first step comprises of two sub-steps:

1) *Inspection:* Proposed approach is performed in two sub-steps. The first sub-step performs inspection of the IP design hardware description language (HDL) files to assist in identification of 1st & 2nd phase watermarks, while the second sub-step performs inspection of the IP HDL file to assist in identification of 3rd phase watermark. For instance in our proposed approach, the first sub-step inspection is performed to collect information on "*functional unit allocation*" and "*non-critical operations ($\mu_m > 0$)*" timing. This is done by feeding the 'Controller (timing) HDL code file' of the IP core design. On the other hand, second sub-step inspection is performed such that the information on "*register allocation"* can be read. This is done by feeding the 'Datapath HDL code file' of the IP core design.
2) *Signature Verification:* The objective of this step is to verify the presence of vendor signature (watermark) in the collected information from two sub-steps earlier. In this step, vendor's watermark is decoded (converted to constraints) using the knowledge of 7-variable signature encoding rules. Finally, the presences of triple-phased watermark in the form of decoded constraints are verified in the collected information. The proposed signature detection process is shown in Fig 5.

IV. Motivational Example and Properties of Proposed Embedding Watermark

A. *Motivational Example: Proposed Watermarking Process*

Fig. 7 shows the scheduled DFG for DWT benchmark based on user provided hardware resources i.e. 3 adders and 3 multipliers. In the proposed approach the concept of two distinct IP vendors is used to attain added security in the encoded signature and possible overall minimization of design area/latency. For example, for each hardware type, two instances are obtained from vendor type 1 (V1) and one instance from vendor type 2 (V2). The IP design schedule prior embedding watermark contains random FU allocation. The respective operation numbers (1-17) appear in the left and the randomly allocated FU type appear in the right of each operation (Fig. 6). For example, the left most operation of the first control step is numbered as operation number (1). 'M2' indicates the multiplier obtained from vendor type 2 which is allocated for this operation. The operations are sorted based on their operation number and the corresponding allocated FU is shown in Table IV. The 1st and 3rd row in the table indicates the operation number of odd and even control step respectively while 2nd and 4th row denotes the corresponding allocated FU. The next step is generation of a table representing timing information of IP design, prior embedding watermark. *Note: timing info of operations in critical path is not shown as watermark is not implanted there*. The list of non-critical path's operation and their corresponding control steps are shown in Table II. The first row indicates the operation number and next row denotes the corresponding control step number. In table II, operations present in the same control step are shown sorted based on mobility. The next subsequent step is selection of a desired vendor signature

Table I: Vendor signature and its decoded meaning (watermark constraints)

Desired Signature	Corresponding operation to shift (Phase 1)	Allocate FU type (Phase 2)	Additional edges to insert between nodes in the colored interval graph (Phase 3)	Observations
ɣ	opn 2 from c.s. 1 to 2	-----	-----	c.s. shift to be done
ɣ	opn 9 from c.s. 3 to 4	-----	-----	c.s. shift to be done
ɑ	-----	opn 1 with vendor 1	-----	FU reallocation to be done
ɓ	-----	opn 2 with vendor 1	-----	No change occurred
ɑ	-----	opn 3 with vendor 1	-----	No change occurred
ɓ	-----	opn 4 with vendor 1	-----	FU reallocation to be done
ɓ	-----	opn 5 with vendor 2	-----	FU reallocation to be done
i	-----	-----	(v2, v3)	Exists by default
I	-----	-----	(v2, v4)	Exists by default
I	-----	-----	(v2, v6)	New edge to be added
T	-----	-----	(v1, v2)	Exists by default
!	-----	-----	(v0, v1)	Exists by default

TABLE II
TIMING TABLE FOR NON-CRITICAL OPERATIONS ($\mu_M > 0$)
SORTED IN INCREASING ORDER OF MOBILITY
(BEFORE EMBEDDING WATERMARK)

Operation No.	3	2	5	4	7	9	8
Control Step	1		2			3	

TABLE III
TIMING TABLE FOR NON-CRITICAL OPERATION ($\mu_M > 0$)
(AFTER EMBEDDING WATERMARK IN PHASE 1)

Operation	3	5	4	7	2	8	9
Control Step	1	2				3	4

TABLE IV
FU ALLOCATION TABLE (BEFORE EMBEDDING WATERMARK)

ODD C.S.	Operation	1	2	3	8	9	10	12	14	16
	Allocated FU	M2	M1	M1	A1	A2	A1	A1	A1	A2
EVEN C.S.	Operation	4	5	6	7	11	13	15	17	--
	Allocated FU	M2	M1	A2	A1	M2	M2	M1	A1	--

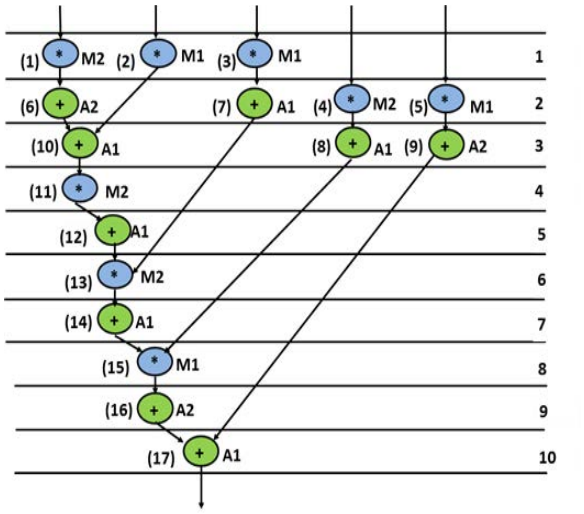

Fig.6. Scheduled DFG (using 3 adders and 3 multipliers) of DWT with random functional unit allocation before embedding watermark

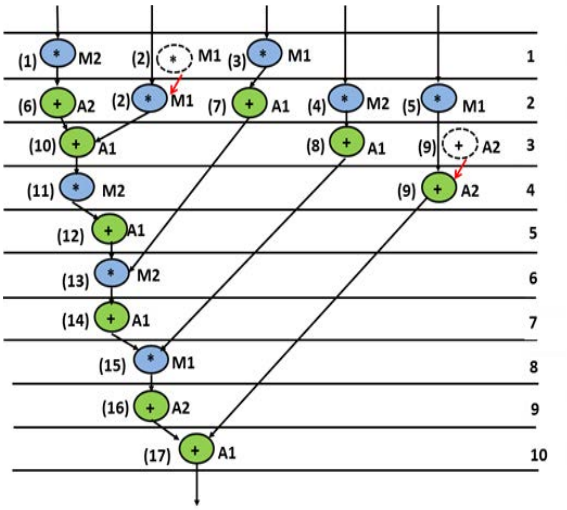

Fig.7. Modified scheduled DFG after embedding phase 1 watermark (‘ɣ’ digits)

provided as watermark. Let us assume a signature: “ɣ ɣ ɑ ɓ ɑ ɓ ɓ i I I T !”. The signature to watermark constraint conversion is shown in Table I. In proposed approach, this signature is inserted as watermark constraints during three consecutive

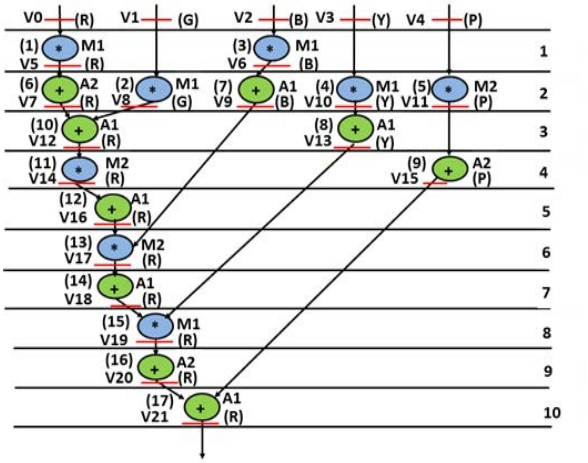

Fig.8. Modified scheduled DFG after embedding phase 1 & 2 watermarks (‘ɑ’, ‘ɓ’ and ‘ɣ’ digits)

phases: scheduling phase, FU allocation and register allocation of HLS respectively.

1st phase watermark embedding: Two consecutive ‘ɣ’ digits exist in the selected signature. According to the encoding rule # 6, a ‘ɣ’ digit moves an opn 2 from c.s. 1 (refer to Table II) to its immediate next control step (i.e. c.s 2). Further again as per the encoding rule, opn number 9 is moved from c.s 3 to 4. This is because other operations of non-critical path viz. 3, 4, 5, 7 and 8 do not satisfy the rule 6. After inserting two ‘ɣ’ digits (as mentioned in the sample signature) the modified scheduled DFG (with first phase watermark embedded) is shown in Fig 7. The modified “*non-critical operations ($\mu_m > 0$)*” table after embedding 1st phase watermark is shown in Table III.

2nd phase watermark embedding: The 1st phase watermarked schedule after embedding ‘ɣ’ digits is used as an input for 2nd phase watermark embedding. Table IV shows the FU allocation of each operation in 1st phase watermarked schedule (before embedding 2nd phase watermark). ‘ɑ’ and/or ‘ɓ’ digits are inserted in the FU allocation phase of HLS. Now as per the selected signature, the 3rd digit of the signature is ‘ɑ’. According to the encoding rule this ‘ɑ’ re-allocates hardware M1 to opn 1. Similarly, the 6th digit i.e. ‘ɓ’ re-allocates hardware M1 to opn 4. Similarly for other encoded digits. The modified “*FU allocation*” table after embedding 2nd phase watermark are shown in Table V. Further, the modified DFG with new allocation (after embedding 2nd phase watermark based on ‘ɑ’ and/or ‘ɓ’ digits) is shown in Fig 8. Additionally, initial random mapping of storage variables to registers (indicated with R, G, B, Y, P) is also denoted in Fig.8. This mapping is shown because the 2nd phase watermarked DFG will be fed as an input for 3rd phase watermarking (based on register re-allocation rule).

3rd phase watermark embedding: The 2nd phase watermarked schedule is used as an input for 3rd phase watermark for inserting ‘i, I, I, T, !’ signature digits into the design. As discussed in previous paragraph, the initial random register assignment for storage variables is shown in Table V.

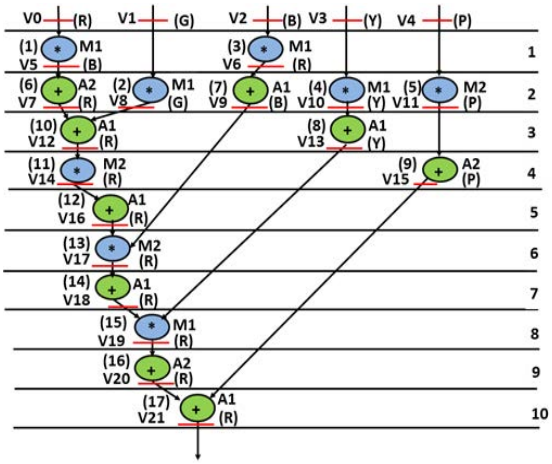

Fig.9. Final scheduled DFG after embedding phase 1, 2 & 3 watermarks (‘α’, ‘δ’, ‘ɣ’, 'i', 'I', 'T' and '!' digits)

The decoded watermark constraints in the form of additional edges to be inserted between nodes of coloured interval graph corresponding to signature digits ‘i, I, I, T, !’ have already been shown in Table I. The coloured interval graph is a graph where nodes represent the storage variables and edges indicate connectivity between nodes (storage variables) whose lifetime overlap (i.e. cannot be allocated to same register). Inserting the additional edges as watermarking constraints, indicate that the storage variables of a colored interval graph are forced to execute through distinct registers. As seen from Table I, an edge between storage variables (v2, v6) is to be added indicating v2 & v6 have to be allocated to distinct registers. Thus as shown in Table VI, v6 is re-allocated to ‘R’ register while v2 remains allocated to ‘B’ register. The DFG with re-allocated registers and its respective coloured interval graph with inserted edges as watermark constraints are shown in Figs. 9 and 10, respectively. The red-dotted line indicates the new edge inserted as watermark constraint corresponding to digit ‘I’. The other digits do not

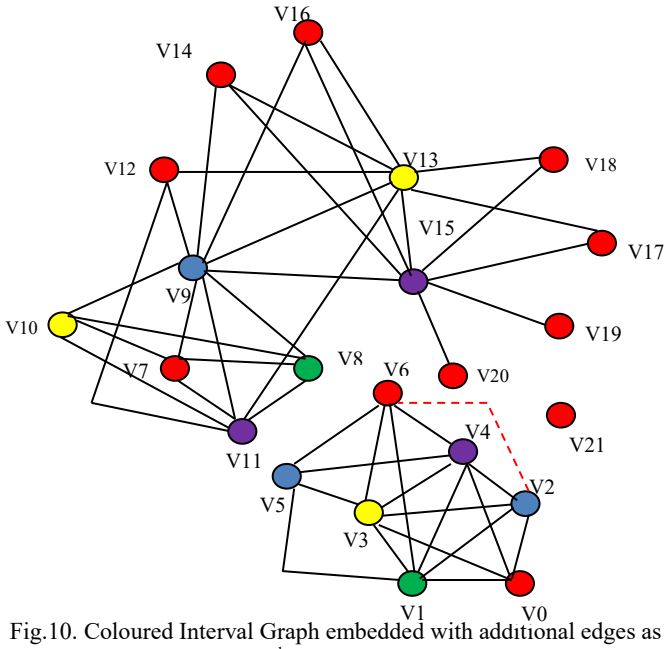

Fig.10. Coloured Interval Graph embedded with additional edges as per 3rd phase watermark

insert any new edge as they exist by default.

In this proposed approach the overhead in scheduling phase is zero. The overhead in allocation phase depends on the hardware from vendor V2. If the difference of hardware in terms of area and latency between V2 and V1 is negligible then the overhead of the watermarked IP design is less.

B. *Properties of Generated Watermark*

A selected properties of the generated watermark includes the following:

(a) *Embedding cost:* The proposed approach produces watermark that incurs low design overhead of area and latency. Further, register overhead is found to be minimal (refer to Table IX).

(b) *Robustness:* The proposed approach implants watermark in three different design phases of HLS. Thus the

TABLE V
HARDWARE ALLOCATION TABLE (AFTER EMBEDDING WATERMARK IN PHASE 2 DURING FUNCTIONAL UNIT ALLOCATION)

C.S. Count	Functional Unit Allocation						Register Allocation					
								R	G	B	Y	P
0	Operation	--	--	--	--	--	Storage Variable	V0	V1	V2	V3	V4
	Allocated Hardware	--	--	--	--	--						
1	Operation	1	3	--	--	--	Storage Variable	V5	V1	V6	V3	V4
	Allocated Hardware	M1	M1	--	--	--						
2	Operation	2	4	5	6	7	Storage Variable	V7	V8	V9	V10	V11
	Allocated Hardware	M1	M1	M2	A2	A1						
3	Operation	8	10	--	--	--	Storage Variable	V12	--	V9	V13	V11
	Allocated Hardware	A1	A1	--	--	--						
4	Operation	9	11	--	--	--	Storage Variable	V14	--	V9	V13	V15
	Allocated Hardware	A2	M2	--	--	--						
5	Operation	12	--	--	--	--	Storage Variable	V16	--	V9	V13	V15
	Allocated Hardware	A1	--	--	--	--						
6	Operation	13	--	--	--	--	Storage Variable	V17	--	--	V13	V15
	Allocated Hardware	M2	--	--	--	--						
7	Operation	14	--	--	--	--	Storage Variable	V18	--	--	V13	V15
	Allocated Hardware	A1	--	--	--	--						
8	Operation	15	--	--	--	--	Storage Variable	V19	--	--	--	V15
	Allocated Hardware	M1	--	--	--	--						
9	Operation	16	--	--	--	--	Storage Variable	V20	--	--	--	V15
	Allocated Hardware	A2	--	--	--	--						
10	Operation	17	--	--	--	--	Storage Variable	V21	--	--	--	--
	Allocated Hardware	A1	--	--	--	--						

TABLE VI
FINAL HARDWARE ALLOCATION TABLE (AFTER EMBEDDING WATERMARK IN PHASE 1, 2 & 3)

C.S. Count	Functional Unit Allocation						Register Allocation					
								R	G	B	Y	P
0	Operation	--	--	--	--	--	Storage Variable	V0	V1	V2	V3	V4
	Allocated Hardware	--	--	--	--	--						
1	Operation	1	3	--	--	--	Storage Variable	V6	V1	V5	V3	V4
	Allocated Hardware	M1	M1	--	--	--						
2	Operation	2	4	5	6	7	Storage Variable	V7	V8	V9	V10	V11
	Allocated Hardware	M1	M1	M2	A2	A1						
3	Operation	8	10	--	--	--	Storage Variable	V12	--	V9	V13	V11
	Allocated Hardware	A1	A1	--	--	--						
4	Operation	9	11	--	--	--	Storage Variable	V14	--	V9	V13	V15
	Allocated Hardware	A2	M2	--	--	--						
5	Operation	12	--	--	--	--	Storage Variable	V16	--	V9	V13	V15
	Allocated Hardware	A1	--	--	--	--						
6	Operation	13	--	--	--	--	Storage Variable	V17	--	--	V13	V15
	Allocated Hardware	M2	--	--	--	--						
7	Operation	14	--	--	--	--	Storage Variable	V18	--	--	V13	V15
	Allocated Hardware	A1	--	--	--	--						
8	Operation	15	--	--	--	--	Storage Variable	V19	--	--	--	V15
	Allocated Hardware	M1	--	--	--	--						
9	Operation	16	--	--	--	--	Storage Variable	V20	--	--	--	V15
	Allocated Hardware	A2	--	--	--	--						
10	Operation	17	--	--	--	--	Storage Variable	V21	--	--	--	--
	Allocated Hardware	A1	--	--	--	--						

generated watermark is extremely robust. The P_c value for proposed approach is $3.2*10^{27}$ times lower than [4] & [5] (refer to Table VII).

(b) *Tamper tolerance:* The proposed approach produces watermark that is tolerant to tampering as the watermark is inserted in three phases of HLS and dispersed throughout the design. The proposed approach is $3.4*10^{43}$ and $2.8*10^{19}$ times tamper tolerant than [4] and [5] respectively (refer to Table VIII).

(c) *Watermark creation and detection time:* The watermark generated through proposed approach is fast. Further, the detection process is straightforward for a genuine entity (who has complete knowledge of encoding rules) however extremely tough to penetrate for an adversary.

V. THREAT SCENARIOS OF FALSE CLAIM OF OWNERSHIP

Entity 'A' owns a watermarked design (D_w) which entity 'B' has purchased from 'A'. In such a case entity 'B' can create the following threats [4]:

(a) *Extracting unintended signature*: Entity 'B' may try to extract his signature through inverse watermark calculation in the original watermarked design 'D_w'. The design may contain attackers signature (besides A's signature) as he may randomly/arbitrarily claim any existing information of the design as his signature [4]. For example, an attacker may claim "*all operations of CS 1 should be allocated to Vendor 1*" (refer to Fig. 9) as his signature encoding rule, which may work for a single design, but will prove to be non-meaningful for other watermarked designs. Thus, this false claim is not strong for proving ownership compared to proposed watermark encoding. In such a conflict, the entity with a more meaningful and stronger watermark (such as proposed watermark) will be the real owner.

(b) *Inserting unauthorized signature*: Entity 'B' may insert his own signature into the original watermarked design of 'A' and claim ownership. Here entity 'B' applies the watermarking constraints corresponding to his own signature on the top of original watermarked design (containing 'A''s signature). In such a conflict the actual owner 'A' can prove his ownership as 'A''s design only contains his watermark (corresponding to his signature), however, 'B''s design contains watermark of both 'A' and 'B'.

(c) *Tampering original signature in the design:* Here 'B' may apply some alterations to the original watermarked design of 'A', trying to create his own unauthorized design. In such a conflict, as the proposed watermarking scheme distributes a strong signature throughout the design in three phases of pre-synthesis, thus complete tampering of all watermarking constraints (corresponding to the strong signature embedded) is extremely difficult. Further, tampering of original signature of the proposed approach may result in latency and hardware overhead as well as the need of performing all the pre-synthesis steps post-tampering.

VI. EXPERIMENTAL RESULTS

The original design before embedding any constraints is termed as baseline design. The baseline design, proposed approach, related works [4] and [5] all are implemented in java and run on AMD A8- 4500M APU with 4 GB DDR3 memory at 1.9GHz. 15nm technology scale based on NanGate is used to evaluate the area and latency of a watermark design [19]. The proposed approach is capable to handle any medium to large size application ranging from 40 components (e.g. EWF) to excess of 100 components (e.g. JPEG IDCT) in the register transfer level designs. Therefore, the proposed watermarking methodology is highly robust for complicated designs such as JPEG IDCT, MPEG MV etc.

A. Evaluation of Robustness of Proposed Watermark

Table VII reports the probability of coincidence (P_c) for proposed approach, [4] and [5]. It calculates the probability of

generating same solution before and after embedding watermark and indicates strength of watermark (lower the value of P_c better it is). The metric is derived from [4]:

$$P_c = \left(1 - \frac{1}{c \times \prod_{i=1}^{D} N(R_i)}\right)^w \quad (1)$$

In the above expression, 'w' is the number of digit used for watermark, $N(R_i)$ is number of hardware of type i, 'D' indicates type of hardware and 'c' is number of colour used in register allocation phase. Comparison of probability of coincidence (P_c) as shown in Table VII indicates that the proposed approach achieves much lower P_c values compared to [4] and [5] i.e. for the proposed approach the strength of watermark is much stronger than both [4] and [5]. The proposed approach have $3.2*10^{27}$ times lower P_c value in average compare to [4] and [5]. Further for the proposed approach, as the watermark constraints (strength) increases, the probability of coincidence decreases. Thus with increasing signature strength, the proof of ownership (watermark robustness) is stronger. For example, the probability of coincidence of MPEG benchmark is $7.7*10^{-18}$, $1.5*10^{-23}$ and $3.8*10^{-31}$ for watermark sizes 45, 60, 80 respectively. P_c is only used to measure robustness (strength of authorship). Decreasing P_c as much as possible is desirable, however not at the expense of too much design cost. Decreasing Pc as much as possible is simply possible by increasing the size of the watermark constraints (signature digits), but simply increasing watermarking constraints to a large extent may increase in register and hardware overhead, thereby increasing design cost. The range of desirable signature strengths is reported in Table VIII. Fig.11. shows the probability of coincidence decreases (strength of watermark increases) with the increment of number of watermark embedding phase. Further, fig.11 also indicates comparison of P_c for each watermarking phases (1st , 2nd and 3rd) of proposed approach with [4] and [5]. As evident, the proposed watermark robustness (through a combination of three watermark phases) is significant higher (at-least 6 $x10^{15}$ times) than [4] and [5].

In the proposed approach, the number of variables cannot be reduced from scheduling phase as it contains only a single variable (ɣ) for signature encoding. Further, two encoded variables of hardware allocation phase cannot be reduced as it employs multi-vendor concept during watermark embedding. However, the number of variables in the register allocation phase of watermarking is four, which may be reduced to obtain the same value of P_c overall occasionally but at the expense of reduced tamper-tolerance ability. This is because, tamper-tolerance ability is directly proportional to the number of variables used (refer eqn. 2).

TABLE VII
COMPARISON OF STRENGTH OF WATERMARK INDICATED THROUGH PROBABILITY OF COINCIDENCE (AS PROOF OF AUTHORSHIP) BETWEEN PROPOSED [4] AND [5] FOR SIGNATURE SIZE (80DIGITS)

Benchmarks [4,5]	# of register before watermark	P_c			# of times lower P_c of proposed approach compared to [4] & [5]
		Proposed	[4]	[5]	
ARF	8	$3.3x10^{-27}$	$2.2x10^{-5}$	$2.2x10^{-5}$	$6.9x10^{21}$
DCT	8	$3.7x10^{-21}$	$2.2x10^{-5}$	$2.2x10^{-5}$	$6.1x10^{15}$
DWT	5	$8.3x10^{-35}$	$1.7x10^{-8}$	$1.7x10^{-8}$	$2.1x10^{26}$
EWF	4	$6.8x10^{-39}$	$1.0x10^{-10}$	$1.0x10^{-10}$	$1.5x10^{28}$
IDCT	8	$3.3x10^{-27}$	$2.2x10^{-5}$	$2.2x10^{-5}$	$6.9x10^{21}$
MPEG MV	14	$3.8x10^{-31}$	$2.6x10^{-3}$	$2.6x10^{-3}$	$6.9x10^{27}$
JPEG IDCT	12	$1.9x10^{-23}$	$9.4x10^{-4}$	$9.4x10^{-4}$	$5.0x10^{19}$

B. Evaluation of Tamper-Tolerance of Proposed Watermark

Table VIII shows the maximum number of possible signatures that can be generated by the combinations of encoding variables (v) for proposed approach, [4] and [5] for different signature strength. A watermark is more tamper tolerant (T_t) if finding its equivalent signature is tougher through brute force analysis i.e. higher the number of combination of possible signature digits, higher is the time consumed and cost expended for an attacker to identify the exact match through brute-force search. The exhaustive possible signature combinations of a 'v' variable encoded signature of strength 'w' digits are given by the following expression:

$$T_t = v^w \quad (2)$$

For example, the maximum number of signature combinations for proposed approach (with respect to w = 80 digits and v = 7 variables) is $4.1*10^{67}$, which is $3.4*10^{43}$ and $2.8*10^{19}$ times higher in tamper tolerance capability (in terms of brute force search) than [4] and [5] respectively.

C. Evaluation of Design Cost of Proposed Watermark:

Evaluation is performed in terms of watermark design area, latency and cost respectively, where watermark design cost is evaluated based on the function adopted from [4], [5]:

$$C_f(X_i) = w_1 \frac{L_T}{L_{max}} + w_2 \frac{A_T}{A_{max}} \quad (3)$$

In the above expression, $C_f(X_i)$ is the watermark cost of the design solution for user provided hardware configuration X_i, L_T and A_T indicates design latency and design area of a watermarked design. The values of area and latency of a watermarked solution is performed based on the models proposed in [4], [5] where area/latency components include functional units, registers and multiplexers. A_{max} and L_{max} indicate maximum possible area and latency in the design space, w_1 and w_2 are user defined weights kept at 0.5 to provide equal priority.

The tradeoff of proposed approach (in terms of design overhead) with the baseline for area, latency and cost is reported in Table IX. Since the proposed watermarking approach (containing 7 variable signatures) may impose area overhead nominally compared to an un-protected design, thus the power overhead of the design may increase trivially.

TABLE VIII
COMPARISON OF TAMPER TOLERANCE BETWEEN PROPOSED, [4] AND [5] FOR DIFFERENT SIGNATURE STRENGTH

Signature Size (digits)	# of possible signature combination			# of times higher tamper-tolerance of proposed approach compared to [4] & [5]	
	Proposed	[4]	[5]	[4]	[5]
15	$4.8*10^{12}$	32768	$10.7*10^{8}$	$14.5*10^{7}$	4421
30	$2.3*10^{25}$	$1.1*10^{9}$	$1.2*10^{18}$	$2.1*10^{16}$	$19.5*10^{6}$
45	$1.1*10^{38}$	$3.5*10^{13}$	$1.2*10^{27}$	$3.0*10^{24}$	$8.6*10^{10}$
60	$5.1*10^{50}$	$1.2*10^{18}$	$1.3*10^{36}$	$4.4*10^{32}$	$3.8*10^{14}$
80	$4.1*10^{67}$	$1.2*10^{24}$	$1.5*10^{48}$	$3.4*10^{43}$	$2.8*10^{19}$

TABLE IX
COMPARISON OF PROPOSED APPROACH WITH BASELINE IN TERMS OF AREA, LATENCY, COST AND COST OVERHEAD %

Benchmarks [4,5]	Resource Configuration	Area (μm²)		Latency (ns)		Cost		Cost Overhead %
		Baseline	Proposed	Baseline	Proposed	Baseline	Proposed	Proposed approach with respect to baseline
ARF	5(+), 3(*)	191.1	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	5.00
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(*)	747.9	756.55	14.9	15.92	0.72	0.76	5.56

TABLE X
COMPARISON OF PROPOSED APPROACH WITH [4] AND [5] IN TERMS OF REDUCED WATERMARK DESIGN AREA, LATENCY AND COST FOR SIGNATURE STRENGTH: 80

Benchmarks [4,5]	Hardware configuration	Watermark Design Area (μm²)			Watermark Design Latency (ns)			Watermark Design Cost		
		Proposed	[4]	[5]	Proposed	[4]	[5]	Proposed	[4]	[5]
ARF	5(+), 3(*)	209.19	225.71	223.35	3.11	3.11	3.11	0.87	0.92	0.90
DCT	6(+), 3(*)	263.45	290.98	288.62	4.19	4.51	4.51	0.84	0.94	0.92
DWT	2(+), 4(*)	165.94	182.37	180.01	2.08	2.43	2.43	0.81	0.93	0.92
EWF	3(+), 2(*)	197.39	209.19	204.47	3.82	3.89	3.89	0.95	0.99	0.98
IDCT	5(+), 3(*)	253.23	280.96	278.4	4.16	4.34	4.34	0.83	0.91	0.89
MPEG	3(+), 8(*)	287.05	309.85	309.85	2.59	2.77	2.77	0.76	0.81	0.81
JPEG	5(+), 5(*)	756.55	783.29	783.29	15.92	16.52	16.52	0.76	0.79	0.79

Therefore system reliability as function of power is marginally affected sometimes; however the robustness of the system due to triple phase and 7 variables is guaranteed to increase manifold. Table X shows the comparative study between the proposed approach, [4] and [5] in terms of watermark design area, watermark design latency and watermark design cost. Table XI reports the reduction % of watermark design area, watermark design latency and watermark design cost for proposed compared to [5]. For example the watermark design area, latency and cost reduction obtained through proposed approach for JPEG benchmark is 3.41%, 3.63% and 3.80% respectively. For the tested benchmarks, the average reduction % of area, latency and cost obtained compared to [5] are 6.65%, 5.37% and 6.25% respectively. Table XI also reports the number of registers required for proposed approach and [5] for a watermarked design. Table XII the reduction % of watermark design area, watermark design latency and watermark design cost for proposed compared to [4]. For example the area, latency and cost reduction obtained through proposed approach for IDCT benchmark is 9.87%, 4.14% and 8.79% respectively. For the tested benchmarks, the average reduction % of area, latency and cost obtained compared to [4] are 7.44%, 5.37% and 7.38% respectively. Reductions through proposed approach have been obtained due to use of multi-IP vendor concept, unlike [4] and [5]. Table XII also reports the number of registers required for proposed approach and [4] for a watermarked design. Finally, Fig. 12 summarizes the reductions of watermark design cost obtained through proposed approach compared to [4] and [5].

Despite of embedding watermark in three different phases the proposed approach achieves significant reduction in area, latency and cost than [4], [5] due to the following reasons:

(a) For 80 digits signature reported for embedding watermark, the proposed approach uses register allocation based watermark (i, I, T, !) partially, while the remainder signature digits are embedded through hardware

TABLE XI
REDUCTION PERCENTAGE (%) OF PROPOSED APPROACH COMPARED TO [5] FOR WATERMARK DESIGN AREA, LATENCY & COST AND COMPARISON OF STORAGE HARDWARE WITH [5]

Benchmarks [4,5]	Area (redu. %)	Latency (redu. %)	Cost (redu. %)	# of storage hardware		
				Before watermark	Proposed (after watermark)	[5] (after watermark)
ARF	6.34	0	3.33	8	8	8
DCT	8.72	7.09	8.70	8	8	8
DWT	7.82	14.40	11.96	5	6	6
EWF	3.85	1.80	3.06	4	4	4
IDCT	9.04	4.14	6.74	8	9	9
MPEG	7.36	6.50	6.17	14	14	14
JPEG	3.41	3.63	3.80	12	12	12

TABLE XII
REDUCTION PERCENTAGE (%) OF PROPOSED APPROACH COMPARED TO [4] FOR WATERMARK DESIGN AREA, LATENCY & COST AND COMPARISON OF STORAGE HARDWARE WITH [4]

Benchmarks [4,5]	Area (redu. %)	Latency (redu. %)	Cost (redu. %)	# of storage hardware		
				Before watermark	Proposed (after watermark)	[4] (after watermark)
ARF	7.32	0	5.44	8	8	9
DCT	9.46	7.09	10.64	8	8	9
DWT	9.01	14.40	12.90	5	6	7
EWF	5.64	1.80	4.04	4	4	6
IDCT	9.87	4.14	8.79	8	9	10
MPEG	7.36	6.50	6.17	14	14	14
JPEG	3.41	3.63	3.80	12	12	12

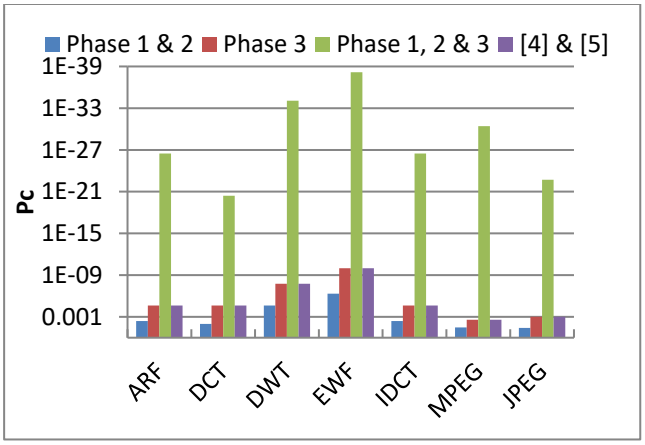

Fig.11. Increment of watermark strength with increase in watermark embedding phases in terms of P_c of proposed approach compared to [4], [5].

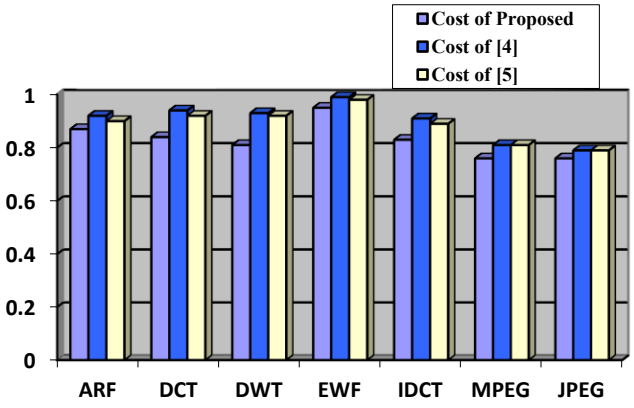

Fig.12. Reduction of cost for proposed approach compared to [4], [5].

allocation and scheduling (other two phases). On the contrary, [4] [5] employs register allocation based watermark for the entire 80 digit signature for watermark embedding. Since, register allocation based watermark incurs register overhead in most cases, thus [4], [5] consumes more area always than proposed approach

(b) The proposed approach uses multi-vendor concept in hardware allocation phase (signature digits: α, β) of watermark compared to single vendor hardware allocation watermark in [4], [5], thus proposed approach is more likely to have optimized (minimized) area and delay than [4] [5].

(c) During scheduling phase, the proposed approach embeds signature digits (ɣ) in the non-critical path of the design which may result into occasional or zero latency overhead. Thus, the latency overhead is mostly minimal because the watermarked signature digits are not embedded in the critical path operations. This contributes to lower design cost in proposed approach.

In Table XIII, the details of component allocation (hardware assignment) to operations for proposed approach, [4] and [5] are shown. As evident from Table XIII the proposed approach optimizes the component allocation through multi-vendor concept (through signature digits: α, β during hardware allocation phase of watermark embedding), where, delay of multiplier and adder from vendor U2 < delay of multiplier and adder from vendor U1. On the contrary, for [4], [5] component allocation to all operations is entirely done through single vendor U1. As shown in Table XIII, for DWT benchmark all operations (9 additions, 8 multiplications) are allocated to components of vendor U1 in case of [4], [5]. However, for proposed approach, 7 additions & 5 multiplications are allocated to vendor U1 and 2 additions & 3 multiplications are allocated to vendor U2 based on α, β digits of watermark signature. As mentioned earlier, since delay of vendor U2 < delay of vendor U1, thus proposed approach achieves lower delay than [4], [5] due to hybrid allocations of multiple vendors (through watermark insertion). Similar trend is observed for other applications. The above explanation can be summarized through the following latency (delay) models:

$$L_T^{[4],[5]} = \sum_{n=1}^{N} Max\{(T^{A1}), (T^{M1})\} \quad (4)$$

$$L_T^{proposed} = \sum_{n=1}^{N} Max\{(T^{A1}), (T^{M1}), (T^{A2}), (T^{M2})\} \quad (5)$$

Where, $L_T^{[4],[5]}$ and $L_T^{proposed}$ are the latencies of [4], [5] and proposed approach respectively; T^{A1}, T^{M1} = delay of adder and multiplier of vendor U1, T^{A2}, T^{M2} = delay of adder and multiplier of vendor U2; N = total number of control step in a watermarked schedule. In the context of above equations, $L_T^{proposed} < L_T^{[4],[5]}$ due to hybrid component allocation using multi-vendor concept in proposed approach.

Further Table XIII shows the length of the critical path (in control step) for proposed approach, [4] and [5]. While insertion of 'ɣ' digits of proposed watermarking, the operations of non-critical path may are shifted to lower control steps. Since the critical path remains untouched thus the final latency remains same as the critical path length. This is evident from Table XIII which also shows the length of the non-critical path after 'ɣ' insertion. Since the length of the non-critical path after 'ɣ' insertion < the length of the critical path, thus there is no latency overhead due to 'ɣ' digit insertions i.e. latency remains same as [4], [5]. For example in case of DWT the critical path length is 10cs (equivalent to 2.08ns as shown in Table VII), while the non-critical path length after 'ɣ' insertions finishes at 9th control step. This indicates no latency overhead is incurred. However, the latency reduction compared to [4], [5] is effectively obtained in the proposed approach through insertion of 'α', '6' digits of watermark using multi-vendor component allocation concept as explained in earlier paragraph. Additionally Table XIV indicates the complexity of proposed approach, [4], [5] in terms of implementation run-time (complexity). As evident from the Table XIV proposed approach incurs slightly more implementation complexity than [4], [5] due to triple phase watermark insertion. However, at the cost of nominal increase in run-time, proposed approach offers stronger robustness (authorship proof) and greater tamper tolerance than [4], [5].

VII. Conclusion and Future Work

A novel highly robust 7-variable signature encoding based triple-phase watermarking for IP core protection of CE hardware during architectural synthesis has been proposed. The presented watermarking achieved a several magnitude higher robustness (evident through strength of watermark) as well as average reduction of cost by 7.38% and 6.25 %

TABLE XIII
COMPARISON OF PROPOSED APPROACH WITH [4] AND [5] IN TERMS OF VENDOR ALLOCATION

Benchmarks	Total DFG operations	Component allocation to opns from multi-vendor (Un) due to 'α' and 'ϐ' insertion in watermarked design (for proposed approach)		Component allocation to opns from single vendor in watermarked design (for [4], [5])	Proposed approach (Impact on latency)	
		Vendor U1	Vendor U2	Vendor U1	Length of critical path (in cs)	Length of non-critical path after 'ɣ' insertion (in cs)
ARF	12(+), 16(*)	8(+), 10(*)	4(+), 6(*)	12(+), 16(*)	8	7
DCT	29(+), 13(*)	18(+), 8(*)	11(+), 5(*)	29(+), 13(*)	8	8
DWT	9(+), 8(*)	7(+), 5(*)	2(+), 3(*)	9(+), 8(*)	10	9
EWF	26(+), 8(*)	14(+), 4(*)	12(+), 4(*)	26(+), 34(*)	14	14
IDCT	29(+), 13(*)	17(+), 7(*)	12(+), 6(*)	29(+), 13(*)	6	5
MPEG	14(+), 14(*)	9(+), 7(*)	5(+), 7(*)	14(+), 14(*)	4	4
JPEG	75(+), 37(*)	44(+), 20(*)	31(+), 17(*)	75(+), 37(*)	8	5

TABLE XIV
COMPARISON OF PROPOSED APPROACH WITH [4] AND [5] IN TERMS OF WATERMARK EMBEDDING COMPLEXITY

Benchmarks	Watermark Embedding Time (ms)		
	Proposed	[4]	[5]
ARF	76	24	46
DCT	88	29	69
DWT	26	15	17
EWF	64	28	51
IDCT	87	31	62
MPEG	80	34	65
JPEG	138	68	109

compared to [4] and [5] respectively. Further reductions of area, latency compared to [4] and [5] were also obtained. The proposed approach demonstrates strong tamper tolerance ability besides robust proof of authorship. Other desirable properties of watermark including low embedding cost, low watermark creation time, adaptability to modern CAD/EDA tools, are exhibited by the proposed approach.

Our future work is geared towards protecting CE hardware from other forms of threats such as illegal licensing, sub-licensing etc. Further we intend to analyse the effect of applying stronger signature encoding scheme on the watermark design overhead and tamper tolerance ability. Another dimension of our future research intends to integrate robust user fingerprint with strong vendor watermark for symmetrical IP core protection during architectural synthesis. The overall design overhead incurred due to both forms of secret mark will also be analysed in the future.

REFERENCES

[1] S. P. Mohanty, U. Choppali, and E. Kougianos, "Everything You wanted to Know about Smart Cities", *IEEE Consumer Electronics Magazine*, Volume 6, Issue 3, 2016, pp. 60—70.

[2] R. Maes, D. Schellekens and I. Verbauwhede, "A Pay-per-Use Licensing Scheme for Hardware IP Cores in Recent SRAM-Based FPGAs," in *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 1,. 2012, pp. 98-108.

[3] A. Cui, G. Qu and Y. Zhang, "Ultra-Low Overhead Dynamic Watermarking on Scan Design for Hard IP Protection," in *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 11, 2015, pp. 2298-2313.

[4] F. Koushanfar, I. Hong, and M. Potkonjak, "Behavioral synthesis techniques for intellectual property protection", *ACM Trans. Des. Autom. Electron. Syst*., Vol. 10, No. 3, 2005, pp. 523-545.

[5] A. Sengupta, S. Bhadauria, and S. P. Mohanty, "Embedding Low Cost Optimal Watermark During High Level Synthesis for Reusable IP Core Protection, in *Proc. of 48th IEEE Intl Symposium on Circuits & Systems (ISCAS)*, 2016, pp. 974–977.

[6] Y. Alkabani, F. Koushanfar, and M. Potkonjak, ``Remote activation of ICs for piracy prevention and digital right management," in *Proc. IEEE/ACM International. Conf. Computer-Aided Design*, 2007, pp. 674-677.

[7] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Trans. Circuits System Video Technologies*, vol. 16, no. 3, 2006, pp. 354-362.

[8] L. M. Marvel, "Information hiding: Steganography and watermarking," in *Optical and Digital Techniques for Information Security (Advanced Sciences and Technologies for Security Applications),* vol. 1. B. Javidi, Ed. New York, NY, USA: Springer, 2005, pp. 113-133.

[9] I. J. Cox, M. L. Miller, J. A. Bloom, J. Fridrich, and T. Kalker, *Digital Watermarking and Steganography*. San Mateo, CA, USA: Morgan Kaufmann, 2007.

[10] Y.-T. Wu and F. Y. Shih, "Digital watermarking based on chaotic map and reference register," *Pattern Recognit*., vol. 40, no. 12, pp. 3753_3763, 2007. [Online].

[11] E. Kougianos, S. P. Mohanty, and R. N. Mahapatra, ``Hardware assisted watermarking for multimedia," *Comput. Elect. Eng*., vol. 35, no. 2, pp. 339-358, 2009.

[12] D. Maltoni, D. Maio, A. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*. New York, NY, USA: Springer, 2009.

[13] J. A. Roy, F. Koushanfar, and I. L. Markov, ``EPIC: Ending piracy of integrated circuits," in Proc. Design, Autom. Test Europe (DATE), 2008, pp. 1069-1074.

[14] Y. Alkabani and F. Koushanfar, ``Active control and digital rights management of integrated circuit IP cores," in *Proc. Int. Conf. Compil., Archit. Synth. Embedded Syst. (CASES)*, 2008, pp. 227-234.

[15] T. Nie, L. Zhou, Y. Li, "Hierarchical watermarking method for FPGA IP protection*," IETE Tech. Rev*., vol. 30, no. 5, 2013, pp. 367–374.

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

[17] Y. M. Alkabani and F. Koushanfar, "Active hardware metering for intellectual property protection and security," in *Proc. 16th USENIX Security Symposium*, 2007, Art. no. 20.

[18] R. S. Chakraborty and S. Bhunia, "HARPOON: An obfuscation based SoC design methodology for hardware protection," *IEEE Trans. CAD* vol. 28, no. 10, 2009, pp. 1493–1502.

[19] NanGate 15 nm library, http://www.nangate.com/?page id=2328, 2016.

[20] Anirban Sengupta "Protection of IP-Core Designs for CE Products", *IEEE Consumer Electronics Magazine*, Vol 5, 2015, pp. 83- 89.

[21] Anirban Sengupta "Hardware Security of CE Devices: Threat Models and Defence against IP Trojans and IP Piracy", *IEEE Consumer Electronics Magazine*, Volume: 6, Issue: 1, 2017, pp. 130 – 133.

[22] Anirban Sengupta, Dipanjan Roy "Anti-Piracy aware IP Chipset Design for CE Devices: Robust Watermarking Approach", *IEEE Consumer Electronics Magazine*, Volume: 6, Issue: 2, 2017, pp. 118 – 124.

Anirban Sengupta is working as Assistant Professor (Associate Professor appointment approved) in the Discipline of Computer Science and Engineering at Indian Institute of Technology (I.I.T) Indore, where he directs the research lab on 'Secured and Reliable IP core design'. He holds a Ph.D. & M.A.Sc. in Electrical & Computer Engineering from Ryerson University, Toronto (Canada) and is a registered Professional Engineer of Ontario (P.Eng.). In the past, he was also affiliated with Indian Institute of Science (IISc) Bangalore as a visiting research scholar.He holds an external affiliation as 'Honorary Chief Scientist' at VividSparks IT Solutions Pvt Ltd, besides his regular affiliation at IIT-I.

His research/sponsored projects are funded by Department of Science & Technology (Science & Engineering Research Board), Ministry of Electronics & IT (MeitY) as well as supported by Intel Corporation and VividSparks IT Solutions Pvt Ltd. He has 110 Publications & Patents which include Journals, Patents and Invited Book Chapters from IEEE, IET, Elsevier, Springer and USPTO/CIPO/IPO. He is owner 11 Patents (granted/published/pending). In the past, his Patents generated funding from Ontario Center of Excellence (OCE), Canada. He has been 'Awarded highest rating "Excellent" by expert committee of Department of Science & Technology (DST) based on the performance (output) in externally funded project in 2017. He currently serves in Editorial positions of 10 IEEE Transactions/Journals, Elsevier, & IET Journals including IEEE Transactions on Aerospace and Electronic Systems (TAES), IEEE Transactions on VLSI Systems, IEEE Aceess Journal, IET Journal on Computer & Digital Techniques, Elsevier Microelectronics Journal, IEEE Consumer Electronics Magazine, IEEE VLSI Circuits & Systems Letter. He further serves as Guest Editor of IEEE Transactions on Consumer Electronics, IEEE Transactions on VLSI Systems and IEEE Access Journals. He serves as Program Chair of 36th IEEE International Conference on Consumer Electronics (ICCE) 2018, Las Vegas, 3rd IEEE International Symposium on Nanoelectronic and Information Systems (iNIS) 2017 and 15th International Conference on Information Technology (ICIT) 2016. Further he member of the Technical Program Committee of IEEE ICCE, IEEE-CS ISVLSI, ACM GLVLSI, IEEE CCECE and ICIT. He has supervised 4 Ph.D. candidates (2 completed and 2 pursuing) and 10 B.Eng candidates.

He had performed industry interactive research extensively with Calypto, Bluespec, BEECube, Huawei Canada during development of his Ryerson Design Space Exploration Tool arising from his Patent. For his excellence in doctoral research, he has been awarded/nominated by Ministry of Training, Colleges and Universities, Ontario for multiple years through OGS as well as by Ryerson University through GREA, RGA and NSERC ICA for many years. More about him can be found at: www.iiti.ac.in/~asengupt.

Dipanjan Roy is a research scholar in Discipline of Computer Science and Engineering at Indian Institute of Technology (I.I.T) Indore. He is currently pursuing his Ph.D. in CSE. He received his M.Tech. degree in Banking Technology & Information Security from University of Hyderabad, India. He worked as a software development engineer in "Amazon Development Center", Bangalore before joining IIT Indore.

Saraju P. Mohanty (SM'08) is a Professor at the Department of Computer Science and Engineering (CSE), University of North Texas (UNT), where he directs the NanoSystem Design Laboratory (NSDL). He obtained a Ph.D. in Computer Engineering from the University of South Florida (USF) in 2003, a Master's degree in Systems Science and Automation (SSA) from the Indian Institute of Science (IISc), Bangalore, India in 1999, and a Bachelor's degree (Honors) in Electrical Engineering from Orissa University of Agriculture and Technology (OUAT), Bhubaneswar, India in 1995. Prof. Mohanty's research is in "Energy-Efficient High-Performance Secure Electronic Systems". Prof. Mohanty's research has been funded by National Science Foundation (NSF), Semiconductor Research Corporation (SRC), and Air Force. Dr. Mohanty is an inventor of 4 US patents. Prof. Mohanty is an author of 220 peer-reviewed journal and conference articles, and 3 books. His Google Scholar h-index is 27 and i10-index is 82. Prof. Mohanty was conferred the Glorious India Award in 2017 for his exemplary contributions to the discipline. He received Society for Technical Communication (STC) 2017 Award of Merit for his outstanding contributions to IEEE Consumer Electronics Magazine. He was the recipient of 2016 PROSE Award for best Textbook in Physical Sciences & Mathematics from the Association of American Publishers for his book titled "Nanoelectronic Mixed-Signal System Design" published by McGraw-Hill in 2015. He was conferred 2016-17 UNT Toulouse Scholars Award for sustained excellent scholarship and teaching achievements. Prof. Mohanty has been serving on the editorial board of several peer-reviewed international journals or transactions. He currently serves on the editorial board of 6 peer-reviewed international journals, including IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD) and ACM Journal on Emerging Technologies in Computing Systems (JETC). He is currently the Editor-in-Chief (EiC) of the IEEE Consumer Electronics Magazine. Prof. Mohanty currently serves as the Chair of Technical Committee on Very Large Scale Integration (TCVLSI), IEEE Computer Society (IEEE-CS) to oversee a dozen of IEEE conferences. He serves on the steering, organizing, and program committees of several international conferences. Prof. Mohanty has supervised 8 Ph.D. dissertations and 26 M.S. theses; eight of these advisees have received outstanding student awards at UNT. He has received Honors Day recognition as an inspirational faculty at the UNT for multiple years. He has also received UNT Provost's Thank a Teacher recognition for multiple years. More about his biography, research, education, and outreach activities can be obtained from his website: http://www.smohanty.org.