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

A. Sengupta, D. Roy and S. P. Mohanty, "Triple-Phase Watermarking for Reusable IP Core Protection During Architecture Synthesis," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 37, no. 4, pp. 742-755, April 2018, doi: 10.1109/TCAD.2017.2729341.

Introduction

- With the surge in globalization hardware design and manufacturing process and rivalry between the IP vendors, threats such as IP piracy and false claim of IP ownership are intensifying [23]-[27].

- Therefore, the requirements for protection of IP-core designs are paramount importance.

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

- The *novel contributions* of this paper are as follows.
  - Proposed a novel triple-phase watermarking methodology to protect the reusable IP core during HLS.
  - Proposed 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.

Proposed triple-phase watermark at architecture level

- The diagrammatic depiction of the proposed approach is shown in Fig. 1.
- 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.
- Further, it is extremely difficult for an attacker to identify which HLS phases (and how watermark constraints) are embedded in the design.
- 1st phase is independent of both 2nd and 3rd phase watermarks.
- 2nd phase watermark is dependent on 1st phase watermark, therefore, tampering of 1st phase watermark may affect 2nd phase watermark constraints.
- Since 3rd phase is independent of 1st and 2nd phase watermark, therefore, 1st and 2nd phase water mark also enables independent protection of original IP owner.

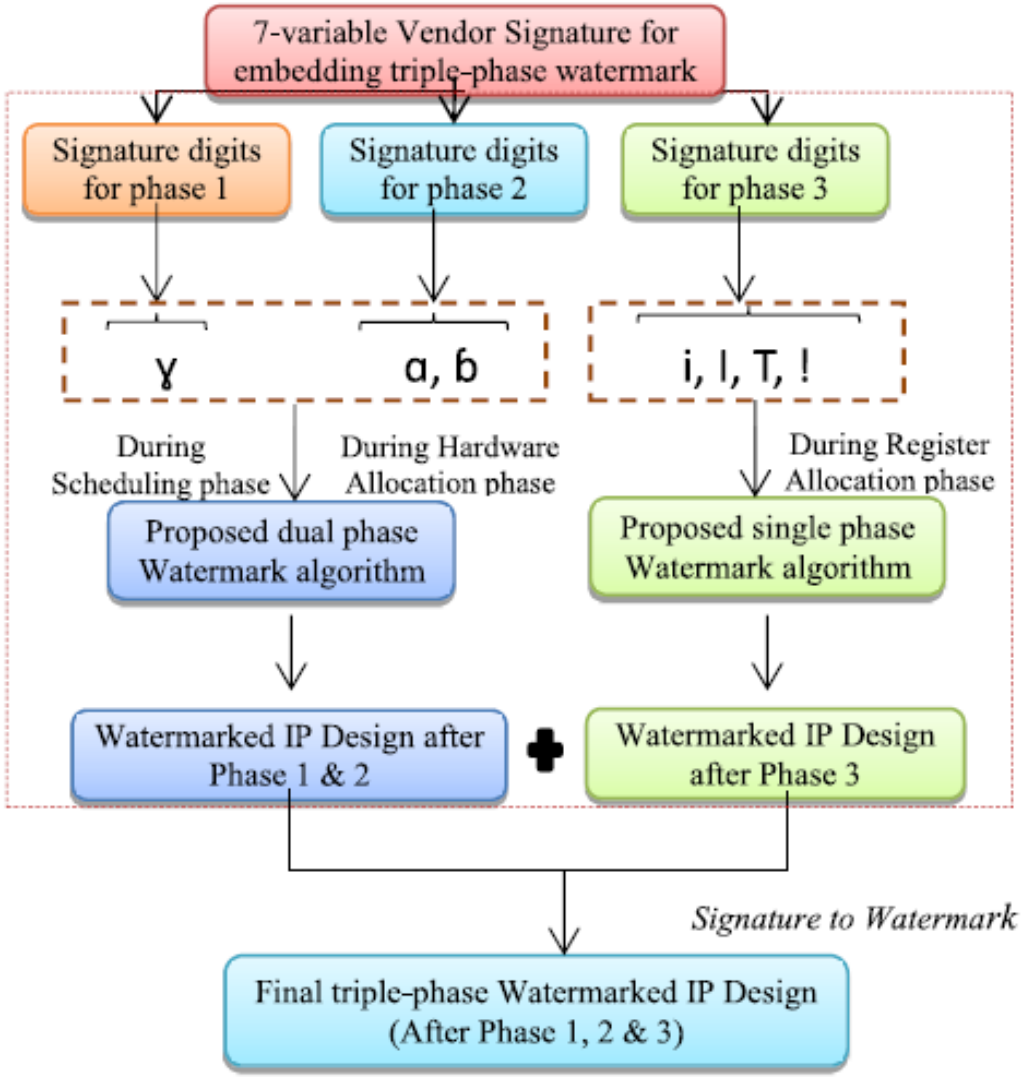

Fig. 1: Proposed triple-phase watermark at architecture level.

Proposed watermarking methodology

- The IP vendor selected seven different signature digits and their corresponding encoding rules are as follows:

  (i) **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),

  (ii) **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),

  (iii) **γ** = Move an operation of noncritical path with highest mobility into immediate next control step (cs),

  (iv) **i** = Embed an artificial edge between <prime, prime> node pairs (storage variables) in colored interval graph (CIG) of DSP application,

  (v) **I** = Embed an artificial edge between <even, even> node pairs (storage variables) in CIG of DSP application,

  (vi) **T** = Embed an artificial edge between <odd, even> node pairs (storage variables) in CIG of DSP application, and

  (vii) **!** = Embed an artificial edge between <0, any integer> node pairs (storage variables) in CIG of DSP application.

- *Representation with tables*:
  - Scheduling phase -> "noncritical operations ($\mu_m > 0$)" timing table, where μ_m denotes the mobility of the operation,
  - Hardware allocation phase -> "functional unit (FU) allocation" table, and
  - Register allocation phase -> "register allocation" table.

Proposed watermarking methodology (Contd.)

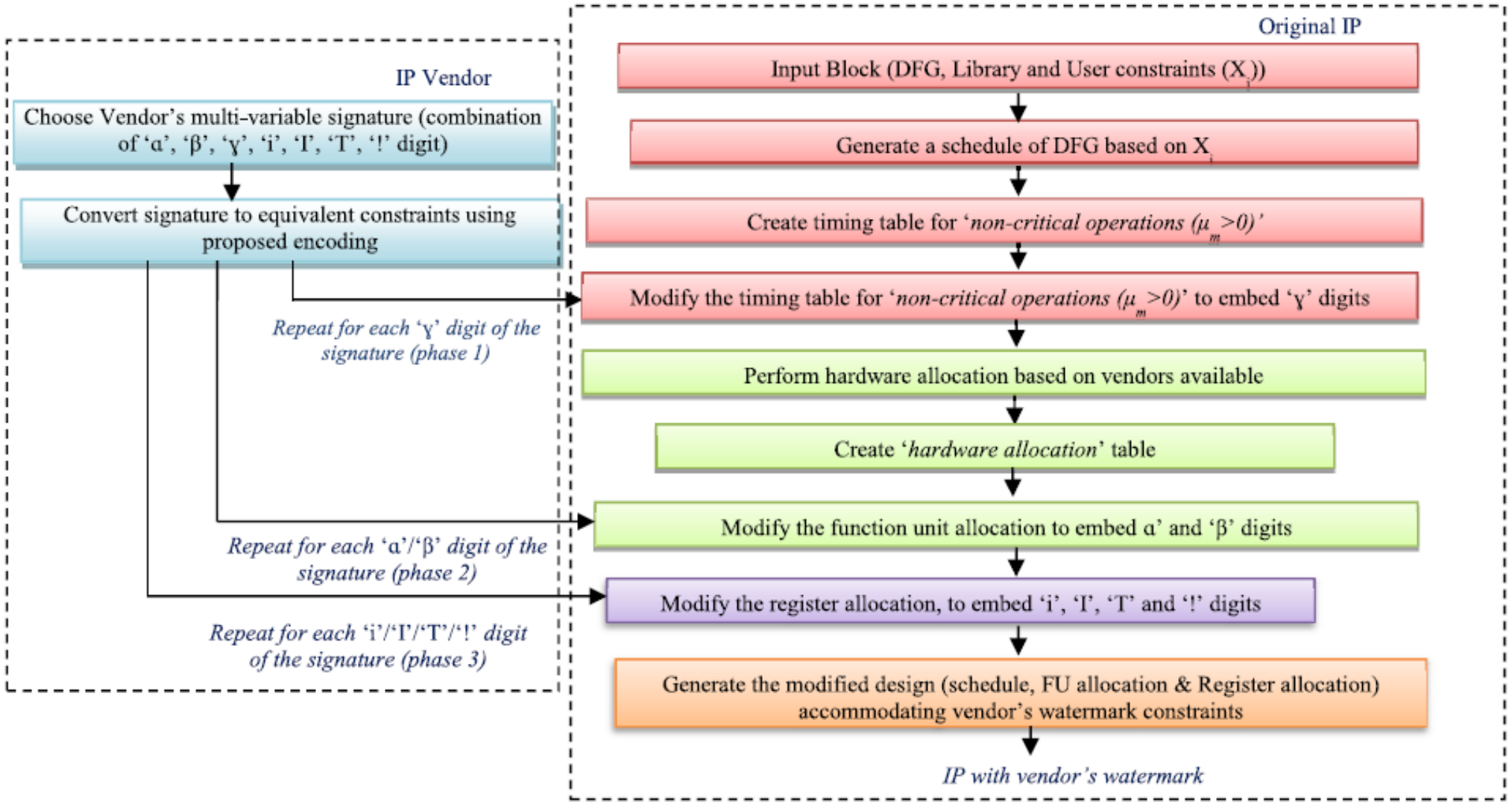

Fig. 2: Proposed HLS flow for reusable IP core protection using triple-phase watermark.

Proposed watermarking methodology (Contd.)

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

- Pre-Embedding Steps (1-5):
  - Based on user provided hardware resources, schedule the DFG.
  - Perform FU allocation based on user provided hardware.
  - To represent an IP design before embedding watermark, generate a FU allocation table for all operations and a noncritical operations ($\mu_m > 0$) timing table.
  - Sort the operations based on their number in increasing order in each control step.
  - Select a 7-variable vendor signature in the form of any combination of α, β, γ , i, I, T, and ! digits.
- *Embedding 1st phase watermark* (Step 6):
  - Move/shift an operation of noncritical path by scanning from control step 1 onward (without repeating) for each occurrence of γ 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).
- *Embedding 2nd Phase Watermark* (Step 7):
  - FU reallocation 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.)

Proposed watermarking methodology (Contd.)

- Modify "hardware allocation" table and noncritical operations ($\mu_m > 0$) table for each encoded digit based on steps 6 and 7 to represent a watermarked IP design (Step 8).
- *Embedding 3rd Phase Watermark* (Steps 9-16):
  - Assign storage variables in the double phased watermarked schedule (obtained in step 7).
  - Create a colored interval graph to find the minimum number of registers required for register allocation.
  - Create a register allocation table for the double phased watermarked scheduling obtained till step 7.
  - Sort storage variables as per their number in increasing order.
  - Feed the 3rd phase vendor signature in the form of i, I, T, and !, in which the characters hold the encoded value of additional edges to be inserted.
  - Create a list of additional edge pairs corresponding to its encoded values by traversing the sorted nodes.
  - Insert the 3rd phase watermarking constraints in the colored interval graph.
  - Modify the register allocation table representing IP design based on colored interval graph in last step.

Proposed watermarking methodology (Contd.)

As a summary:

- Here, in the ***first phase of watermarking***, non-critical operations (starting from CS 1) are moved to the immediate next CS for each occurrence of **'γ'** (shifting must not violate the data dependency and hardware constraints), and a modified timing table for non-critical operations is generated. A hardware allocation table is generated corresponding to different used functional units (hardware).

- Further, in the ***second phase of watermarking***, FUs are re-allocated according to the IP vendor selected encoding rules **'α'** and **'β'**, and a modified hardware allocation table is generated. After this, allocation of storage variables in the SDFG (double phased watermarked) is performed, and a CIG is created to find the minimum number of required registers for storage variables. Next, a register allocation table (RAT) is created from SDFG (assigned with storage variables).

- Then, in the ***third phase of watermarking,*** the additional artificial edges (security constraints) are determined based on the IP vendor's selected **'i'**, **'I'**, **'T'**, and **'!'** digits. Further, these determined security constraints are embedded into the CIG of the design, followed by local alteration in register allocation if two adjacent register's colors are the same. To resolve this conflict, either colors of the register are swapped, or a new colored register is allocated.

- Finally, RAT of triple-phase watermarked hardware IP core is generated using HLS.

Proposed watermarking methodology (Contd.)

- Signature detection is a compulsory step when using watermark for resolving vendor ownerships conflicts.
- Here, signature detection is a two-step process, (i) *Inspection*, and (ii) *Signature verification*.

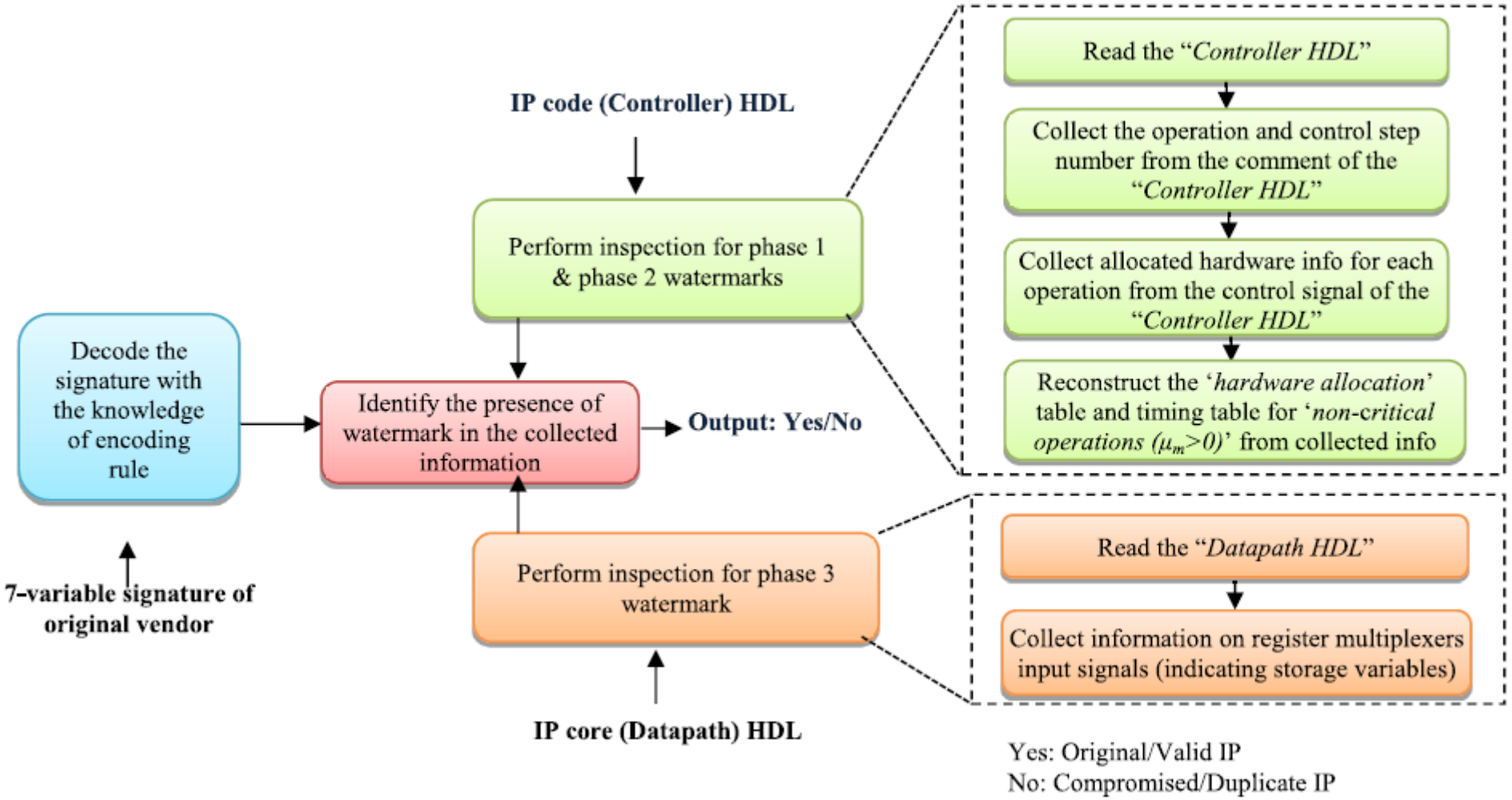

Fig. 3: Signature detection process.

Vendor signature and its decoded meaning (watermark constraints)

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

Motivational example : proposed watermarking process

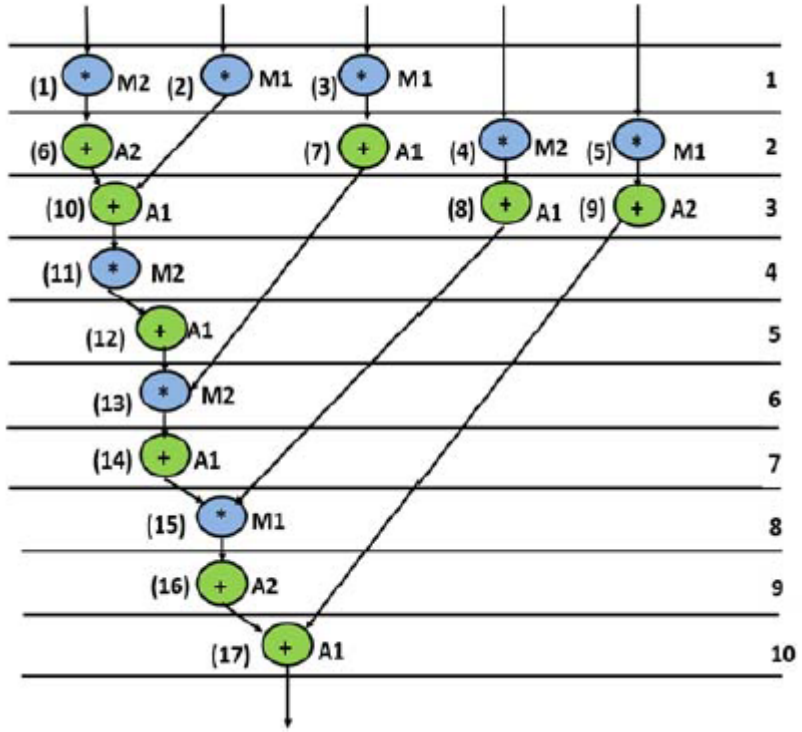

Fig. 4: Scheduled DFG (using three adders and three multipliers) of DWT with random FU allocation before embedding watermark.

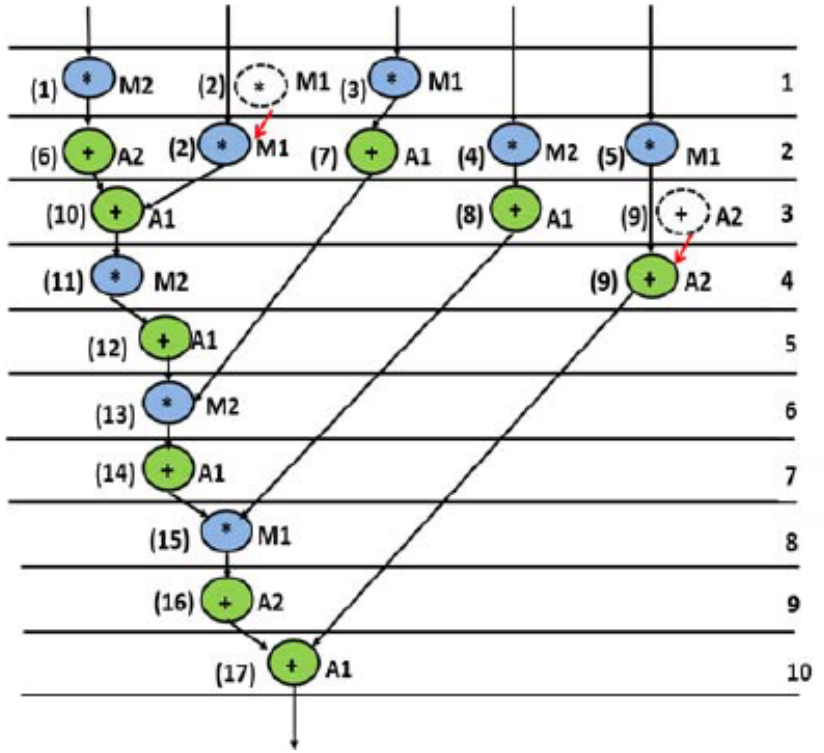

Fig. 5: Modified scheduled DFG after embedding phase 1 watermark (γ digits).

TABLE II
TIMING TABLE FOR NONCRITICAL 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 NONCRITICAL OPERATION ($\mu_m > 0$) (AFTER EMBEDDING WATERMARK IN PHASE 1)

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

Motivational example : proposed watermarking process (Contd.)

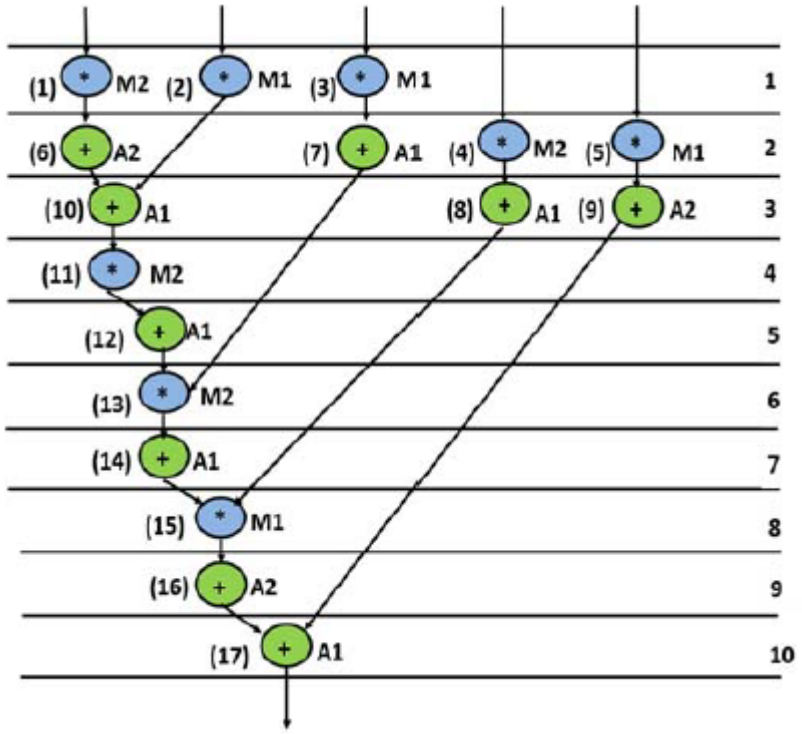

Fig. 4: Scheduled DFG (using three adders and three multipliers) of DWT with random FU allocation before embedding watermark.

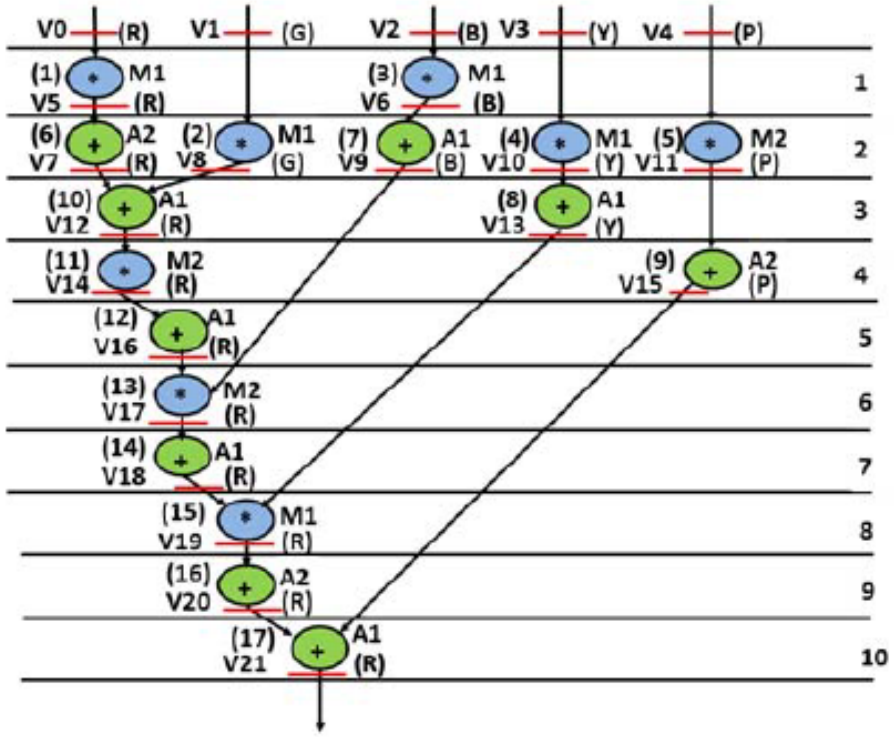

Fig. 6: Modified scheduled DFG after embedding phases 1 and 2 watermarks (α, β, and γ digits).

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	–

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	–

Motivational example : proposed watermarking process (Contd.)

TABLE VI
FINAL HARDWARE ALLOCATION TABLE (AFTER EMBEDDING WATERMARK IN PHASES 1, 2, AND 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	–	–	–	--						

Motivational example : proposed watermarking process (Contd.)

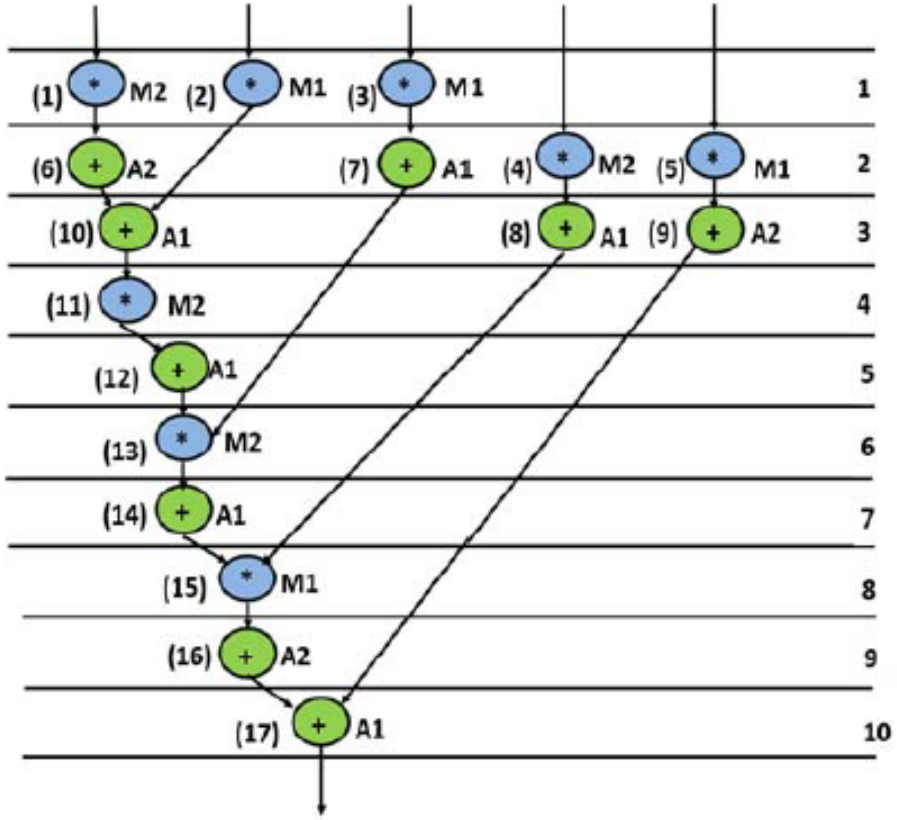

Fig. 4: Scheduled DFG (using three adders and three multipliers) of DWT with random FU allocation before embedding watermark.

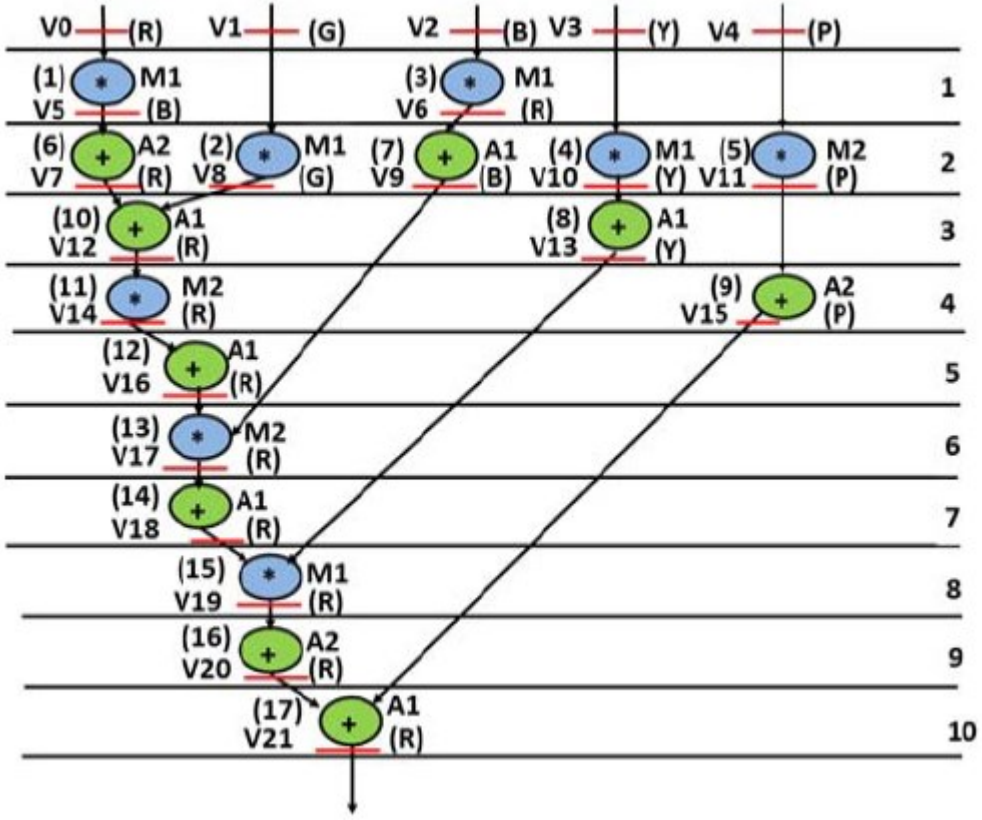

Fig. 7: Final scheduled DFG after embedding phases 1, 2, and 3 watermarks (α, β, γ, i, I, T, and ! digits).

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:
  - *Extracting Unintended Signature*

    Entity B may try to extract the signature, and he/she may randomly/arbitrarily claim any existing information of the design as his/her signature.

    An attacker may claim "all operations of CS 1 should be allocated to Vendor 1" (like Fig. 7) as his/her signature encoding rule, which may work for a single design, but will *prove to be nonmeaningful for other* watermarked designs.

  - *Inserting Unauthorized Signature*

    Entity B may insert his/her own signature into the original watermarked design of A and claim ownership.

    In such a conflict the actual owner A can prove his/her ownership as A's design only contains his/her watermark (corresponding to his/her signature), however, B's design contains watermark of both A and B.

  - *Tampering Original Signature in the Design*

    Here, B may apply some alterations to the original watermarked design of A, trying to create his/her 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.

Properties of generated watermark

- The properties of the generated watermark includes the following:
  - *Embedding cost*

    The proposed approach produces watermark that incurs low design overhead of area and latency. Further, register overhead is found to be minimal.

  - *Robustness*

    The proposed approach implants watermark in three different design phases of HLS. Thus, the generated watermark is extremely robust.

  - *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.

  - *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.

Experimental results

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

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	$1.9x10^{-23}$	$9.4x10^{-4}$	$9.4x10^{-4}$	$5.0x10^{19}$

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	$4.1*10^{67}$	$1.2*10^{24}$	$1.5*10^{48}$	$3.4*10^{43}$	$2.8*10^{19}$

[4] F. Koushanfar, I. Hong, and M. Potkonjak, "Behavioral synthesis techniques for intellectual property protection," ACM Trans. Design Autom. Electron. Syst., vol. 10, no. 3, pp. 523–545, 2005.

[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. 48th IEEE Int. Symp. Circuits Syst. (ISCAS), Montreal, QC, Canada, 2016, pp. 974–977.

Experimental results (Contd.)

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

Benchmarks	Resource Configuration	Area (μm^2)		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

Experimental results (Contd.)

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	Hardware configuration	Watermark Design Area (μm^2)			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

TABLE XI
REDUCTION PERCENTAGE (%) OF PROPOSED APPROACH COMPARED TO [5] FOR WATERMARK DESIGN AREA, LATENCY, AND 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, AND 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

[4] F. Koushanfar, I. Hong, and M. Potkonjak, "Behavioral synthesis techniques for intellectual property protection," ACM Trans. Design Autom. Electron. Syst., vol. 10, no. 3, pp. 523–545, 2005.

[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. 48th IEEE Int. Symp. Circuits Syst. (ISCAS), Montreal, QC, Canada, 2016, pp. 974–977.

Experimental results (Contd.)

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

- The proposed approach uses register allocation-based watermark (i, I, T, and !) partially, while the remainder signature digits are embedded through hardware allocation and scheduling. Since, register allocation-based watermark incurs register overhead in most cases, thus [4] and [5] consumes more area always than proposed approach.

- The proposed approach uses multivendor concept in hardware allocation phase (signature digits: α and β) of watermark compared to single vendor hardware allocation watermark in [4] and [5].
  - Delay of multiplier and adder from vendor U2 < Delay of multiplier and adder from vendor U1
  - On the contrary, for [4] and [5] component allocation of all operations is entirely done through a single vendor U1.
  - However, for proposed approach, seven additions and five multiplications are allocated to vendor U1, and two additions and three multiplications are allocated to vendor U2 based on α, β digits of watermark signature.

- During scheduling phase, the proposed approach embeds signature digits (γ) in the noncritical path of the design which may result into occasional or zero latency overhead. This contributes to lower design cost in proposed approach.

[4] F. Koushanfar, I. Hong, and M. Potkonjak, "Behavioral synthesis techniques for intellectual property protection," ACM Trans. Design Autom. Electron. Syst., vol. 10, no. 3, pp. 523–545, 2005.

[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. 48th IEEE Int. Symp. Circuits Syst. (ISCAS), Montreal, QC, Canada, 2016, pp. 974–977.

Experimental results (Contd.)

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 'δ' 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 'γ' 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

[4] F. Koushanfar, I. Hong, and M. Potkonjak, "Behavioral synthesis techniques for intellectual property protection," ACM Trans. Design Autom. Electron. Syst., vol. 10, no. 3, pp. 523–545, 2005.

[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. 48th IEEE Int. Symp. Circuits Syst. (ISCAS), Montreal, QC, Canada, 2016, pp. 974–977.

Conclusion

- The involvement of *7-digit multi-variable signature* and different *IP vendor-selected encoding mechanisms* for different phases (scheduling, hardware allocation, and register allocation) of watermarking makes the proposed approach highly robust.

- 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.

References

[1] S. P. Mohanty, U. Choppali, and E. Kougianos, “Everything you wanted to know about smart cities: The Internet of Things is the backbone,” IEEE Consum. Electron. Mag., vol. 5, no. 3, pp. 60–70, Jul. 2016.

[2] R. Maes, D. Schellekens, and I. Verbauwhede, “A pay-per-use licensing scheme for hardware IP cores in recent SRAM-based FPGAs,” IEEE Trans. Inf. Forensics Security, vol. 7, no. 1, pp. 98–108, Feb. 2012.

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

[4] F. Koushanfar, I. Hong, and M. Potkonjak, “Behavioral synthesis techniques for intellectual property protection,” ACM Trans. Design Autom. Electron. Syst., vol. 10, no. 3, pp. 523–545, 2005.

[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. 48th IEEE Int. Symp. Circuits Syst. (ISCAS), Montreal, QC, Canada, 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 Int. Conf. Comput.-Aided Design, San Jose, CA, USA, 2007, pp. 674–677.

[7] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, “Reversible data hiding,” IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 3, pp. 354–362, Mar. 2006.

[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.

[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), Munich, Germany, 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. Compilers Archit. Synthesis Embedded Syst. (CASES), Atlanta, GA, USA, 2008, pp. 227–234.

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

[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, pp. 71–92, 2012.

[17] Y. M. Alkabani and F. Koushanfar, “Active hardware metering for intellectual property protection and security,” in Proc. 16th USENIX Security Symp., Boston, MA, USA, 2007, Art. no. 20.

[18] R. S. Chakraborty and S. Bhunia, “HARPOON: An obfuscationbased SoC design methodology for hardware protection,” IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 28, no. 10, pp. 1493–1502, Oct. 2009.

[19] (2016). NanGate 15 nm Library. [Online]. Available: http://www.nangate.com/?pageid=2328

[20] A. Sengupta, “Protection of IP-core designs for CE products,” IEEE Consum. Electron. Mag., vol. 5, no. 1, pp. 83–89, Dec. 2015.

[21] A. Sengupta, “Hardware security of CE devices: Threat models and defence against IP trojans and IP piracy,” IEEE Consum. Electron. Mag., vol. 6, no. 1, pp. 130–133, Jan. 2017.

[22] A. Sengupta and D. Roy, “Antipiracy-aware IP chipset design for CE devices: A Robust watermarking approach [hardware matters],” IEEE Consum. Electron. Mag., vol. 6, no. 2, pp. 118–124, Apr. 2017.

[23] Anirban Sengupta, Saumya Bhadauria, "Exploring Low Cost Optimal Watermark for Reusable IP Cores during High Level Synthesis", IEEE Access, Volume:4, Issue: 99, pp. 2198 - 2215, May 2016

[24] Anirban Sengupta, Saumya Bhadauria, Saraju P Mohanty "TL-HLS: Methodology for Low Cost Hardware Trojan Security Aware Scheduling with Optimal Loop Unrolling Factor during High Level Synthesis", IEEE Transactions on Computer Aided Design of Integrated Circuits & Systems (TCAD), Volume: 36, Issue: 4, April 2017, pp. 655 – 668

[25] Anirban Sengupta, Dipanjan Roy, Saraju Mohanty, Peter Corcoran "DSP Design Protection in CE through Algorithmic Transformation Based Structural Obfuscation", IEEE Transactions on Consumer Electronics, Volume 63, Issue 4, November 2017, pp: 467 – 476

[26] Anirban Sengupta, Mahendra Rathor "IP Core Steganography for Protecting DSP Kernels used in CE Systems", IEEE Transactions on Consumer Electronics (TCE) , Volume: 65 , Issue: 4 , Nov. 2019, pp. 506 – 515

[27] Anirban Sengupta, Mahendra Rathor "Securing Hardware Accelerators for CE Systems using Biometric Fingerprinting", IEEE Transactions on Very Large Scale Integration Systems , Vol 28, Issue: 9, 2020, pp. 1979-1992

Thank You !!!