Embedding Low Cost Optimal Watermark During High Level Synthesis for Reusable IP Core Protection

Anirban Sengupta
Computer Science and Engineering
Indian Institute of Technology, Indore, India
asengupt@iiti.ac.in

Saumya Bhadauria
Computer Science and Engineering
Indian Institute of Tech, Indore, India

Saraju P. Mohanty
Computer Science and Engineering
University of North Texas, Denton, USA
saraju.mohanty@unt.edu

***Abstract*— Intellectual property (IP) cores have emerged as a promising solution to the challenges of future design as well as mounting time to market pressure. However, due to increasing globalization of design supply chain, possibility of intervention and typical attacks is on the rise, which therefore mandates protection of IP cores from piracy/counterfeiting even at behavioral level. This paper presents a technique for generating low cost watermarking solution during high level synthesis (HLS) based on multi-variable signature encoding for security of reusable IP cores. The watermark generated by the proposed approach satisfies the following properties: (a) low embedding cost (b) robustness (c) low watermark creation time (d) strong proof of authorship (e) lower hardware overhead. Comparison with similar technique revealed that proposed approach obtains watermarked solution with lower embedding cost with less storage overhead and creation time.**

***Keywords*— IP Security; Watermark; High level Synthesis**

I. Introduction

To maximize design productivity and minimize design time, use of IP core, often supplied by a third party vendor, has become an industry de-facto standard. Owing to rising threats of security/piracy issues in the global supply chain, protection of IPs is a strong mandate. Embedding a robust watermark at a very high abstraction level (such as behavioral level) can serve as line of defense against typical attacks as well as in nullifying false claim of ownership, thereby protecting the value of a usable IP core. A watermark in general is a hidden signature of the owner embedded in a design. Watermarking on embedding should be capable of preserving the correct functionality of the design with minimal area overhead. The embedding cost of the watermarked solution should be as minimal as possible. This paper presents a technique for exploring low cost optimal watermarking solution based on multi-variable signature encoding embedded during HLS for security of reusable IP cores. Thus is a paradigm shift research in HLS which has been traditionally targeted for low-power [8]. The exploration backbone for our proposed approach for generation of low cost optimal watermarked solution is based on particle swarm optimization (PSO).

II. Related Prior Research

Embedding watermarking at behavioral level for IP protection has been tackled only in few works so far. For example in[1] and [2] , authors use only a combination of 0 and 1 to encode their signature in the form of adding additional edges in the colored interval graph during HLS. However, in such cases the signature is susceptible to attacks/compromise, if encoding rule of both the variable is known somehow. Moreover [1] and [2] are not capable to produce watermark with low embedding cost or less storage overhead. A related research at HLS which does not deal with watermarking for IP protection, rather designs for trusted IC is presented in [9]. Besides, watermarking for IP protection has also been applied at logic synthesis level [3, 4], physical level [5] and other higher level [10]. Apart from this, efforts were made to watermark analog and mixed signal designs [7]. However, no approach exist in the literature that generates a low cost optimal watermark based on robust multi-variable signature encoding at behavioural level for reusable IP core protection.

III. Proposed Methodology

A. Problem Formulation

Given a control data flow graph (CDFG), determine a low cost optimal watermarked solution (X_i) = N(R_1), N(R_2),…N(R_D),with minimum embedding cost(A_T , L_T) subjected to: $A_T \le A_{cons}$ and $L_T \le L_{cons}$ and IP security through watermarking; Where, A_T and L_T are the area and delay of watermarked solutions; A_{cons} and L_{cons} are user area and latency constraints; N(R_D) is the number of a resource type R_D.

B. Proposed Watermarking

For the purpose of embedding a watermark, additional constraints need to be imposed in the design during one of the HLS tasks. Watermarking constraints are applied in the register allocation step of HLS by adding additional constraints in the form of additional edges between the nodes of a colored interval graph. Adding these additional edges as watermarking constraints indicates that the storage variables of a colored interval graph are forced to execute through

Process for embedding watermarking the design

- Schedule the CDFG based on resource configuration provided.
- Assign storage variables in the scheduling and create the colored interval graph to find the minimum number of registers required for allocation.
- Based on colored interval graph, a controller is generated.
- Sort storage variables as per their number in increasing order
- Generate a desired signature in the form of random combination of a tuple comprising of *i, I, T, ! (note: these are chosen variables during encoding process).*
- Build a list *L[k]* of additional edge pairs corresponding to its encoded values by traversing the sorted nodes.
- Insert additional edges as watermark in colored interval graph.
- Modify controller design on the basis of created watermark

Fig.1. Process of Embedding Watermark

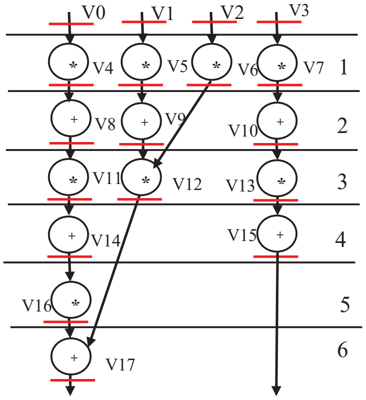

Fig. 2 Scheduling of MESA CDFG with 3 adders and 4 multipliers

Table I: Controller for register allocation before embedding watermark

Control Step(c.s)	Red (R)	Blue (B)	Green (G)	Yellow (Y)
0	v0	v1	v2	v3
1	v4	v5	v6	v7
2	v8	v9	v6	v10
3	v11	v12	v13	--
4	v14	v12	v15	--
5	v16	v12	v15	--
6	v17	--	v15	--

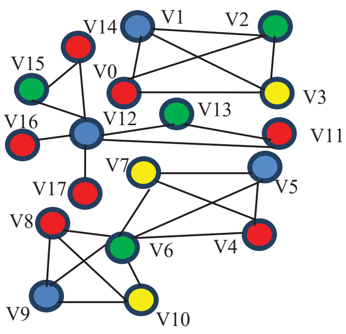

Fig. 3 Colored Interval Graph for the scheduling

Table II: Signature and its decoded meaning

Desired signature (7-digit)	Corresponding additional edges to add in the coloured interval graph
i	(2,3)
i	(2, 5)
I	(2, 4)
I	(2, 6)
T	(1, 2)
T	(1, 4)
!	(0, 1)

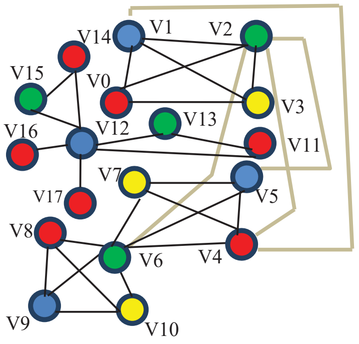

Fig. 4 Colored Interval Graph with additional edges (watermarking constraints) colored in grey

Table III: Modified Controller after embedding watermark (includes authors' hidden signature)

Control step(c.s)	Red (R)	Blue (B)	Green (G)	Yellow (Y)
0	v0	v1	v2	v3
1	v4	v5	v7	v6
2	v8	v9	v10	v6
3	v11	v12	v13	--
4	v14	v12	v15	--
5	v16	v12	v15	--
6	v17	--	v15	--

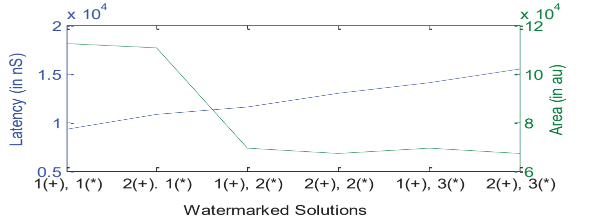

Fig. 5 Orthogonal condition of a watermarked solution during exploration

distinct registers. Our proposed scheme for signature creation comprises of four different variables: i, I , T, ! in contrast to two variables (0, 1) used by previous approaches. Each variable of our signature maps onto a certain edge pair. The following encoding mechanism is proposed:

- 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)

1) *Motivational Example for Embedding Watermark*

Fig.1 shows the proposed process for embedding watermark. Fig. 2 shows the scheduling of MESA CDFG based on a given sample resource configuration of (3 adders, 4 multipliers) as input. The respective storage variables (v0-v17) are indicated in each time step. Fig. 3 shows the corresponding colored interval graph created to find the minimum number of registers required for allocation. The respective controller of IP design is shown in Table I. First a desired signature: "i i I I T T !" is selected. Then the storage variables are sorted and the corresponding additional edges of a colored interval graph are decoded from the chosen signature as shown in Table II. The respective colored interval graph with the inclusion of additional edges (watermarking constraints) is shown in Fig. 4. As seen, four additional edges have been added as watermarking constraints on register allocation. Although in the signature there are seven additional edges to be added, however, by coincidence the remaining three edges were already added by default from before. Based on the new edges added the controller in Table I need to be modified to include these constraints. The modified controller with watermarking constraints embedded (comprising of owners signature) is shown in Table III. As evident, for this particular example embedding watermark did not result in storage overhead. However we note that depending on the strength of the signature, chances of overhead may vary. In our proposed approach, RSA encryption algorithm may also be used to encipher the signature data before embedding extra constraints in the design. This double layered protection of multi-variable signature encoding/embedding and encryption makes the watermark generated highly robust and difficult to tamper.

2) *Motivation for performing Design Space Exploration of an Optimal Watermark*

In case of IP protection using dynamic watermarking (i.e. embedding vendor specific watermark before performing synthesis), performing trade-off is extremely critical. This is because among the various competitive solutions present in the design space, selecting a low cost solution for embedding watermark is non-trivial. Further, choosing a solution without performing trade-off affects the latency and area of the final IP design. This is because every candidate design solution used as watermark impacts latency and area in a different way. For

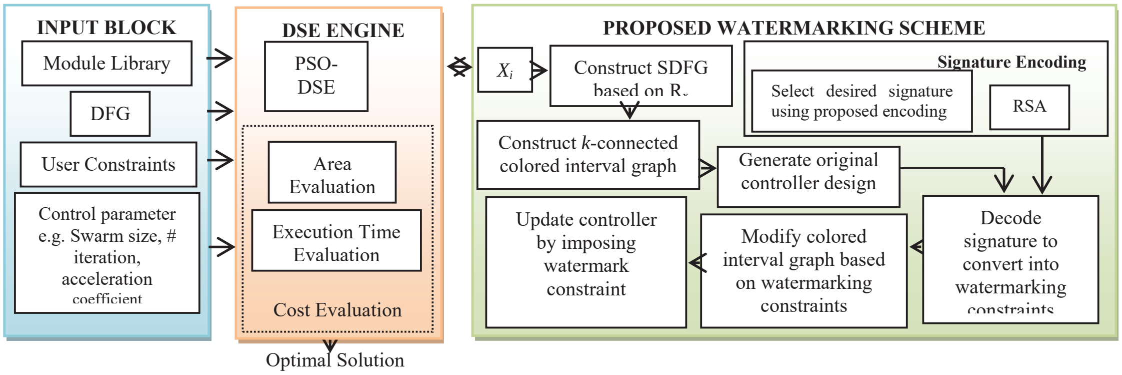
INPUT BLOCK
Module Library
DFG
User Constraints
Control parameter e.g. Swarm size, # iteration, acceleration coefficient
DSE ENGINE
PSO-DSE
Area Evaluation
Execution Time Evaluation
Cost Evaluation
Optimal Solution
PROPOSED WATERMARKING SCHEME
X_i
Construct SDFG based on R_x
Signature Encoding
Select desired signature using proposed encoding
RSA
Construct k-connected colored interval graph
Generate original controller design
Update controller by imposing watermark constraint
Modify colored interval graph based on watermarking constraints
Decode signature to convert into watermarking constraints

Fig.6. Proposed PSO driven design space exploration for optimal watermarking

example, Fig. 5 shows the orthogonal condition for BPF benchmark during exploration of watermarked solution for the potential design solutions. In our work, we have iteratively explored new solutions, applied dynamic watermarking (i.e. embedding watermark at pre-synthesis phase) using the solution in the IP design and evaluated the fitness of the resultant watermarked solution with respect to the latency and area constraints provided. This exploration process is repeated until an optimal watermarked solution (low cost) is found.

3) Proposed Framework for Exploration of Low Embedding Cost Optimal Watermark

The block diagram of proposed framework for generation of a low embedding cost optimal watermark is shown in Fig. 6. The exploration backbone is based on PSO and adopted from reference [6] which has following terminating criteria: a) exploration process reaches designer specified swarm size '*p*'. b) Global best is not improving over last 10 iterations. The particle position 'X_i' is given as Eqn. (1):

$$X_i = (N(R_1), (N(R_2),..(N(R_d)).. (N(R_D)) \qquad (1)$$

Each dimension *d* of a particle position X_i (except the last dimension) in PSO driven DSE is updated as Eqn. (2):

$$R^+_{d_i} = R_{d_i} + V^+_{d_i} \qquad (2)$$

The variable $V^+_{d_i}$ is updated by Eqn. (3):

$$V^+_{d_i} = \omega V_{d_i} + b_1 r_1 \left[R_{d_{lbi}} - R_{d_i} \right] + b_2 r_2 \left[R_{d_{gb}} - R_{d_i} \right] \qquad (3)$$

where, $R^+_{d_i}$ is new resource value of particle X_i in d^th dimension and R_{d_i} is resource value of particle X_i in d^{th} dimension. $V^+_{d_i}$ is new velocity of i^{th} particle in d^{th} dimension. $R_{d_{lbi}}$ is resource value of X_{lbi} (local best position) in d^{th} dimension, ω is inertia weight, $R_{d_{gb}}$ is resource value of X_{gb} in d^{th} dimension, b_1 and b_2 are acceleration coefficients which balances the effect of cognitive and social factor during exploration, r_1 and r_2 are random numbers for stochasticity.

4) Signature Detection

a) *Reverse Engineering:* During this phase relevant information of the received IP is collected in terms of structural property, specifications etc., and the controller is regenerated*;* b) *Signature Verification*: During this step, presence of owner's signature is verified in the regenerated controller design, by identifying the presence of additional watermarking constraints. This is performed by decrypting the enciphered message followed by decoding the signature. The block diagram of signature detection is shown in Fig. 7.

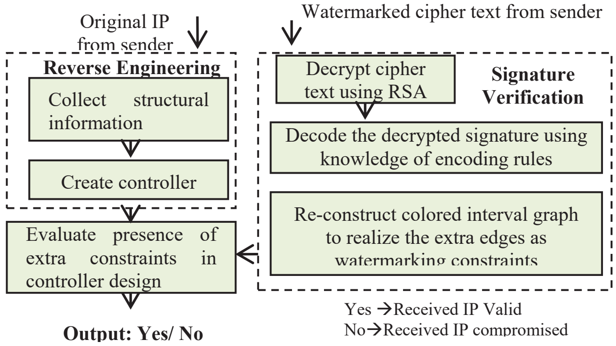
Original IP from sender
Watermarked cipher text from sender
Reverse Engineering
Collect structural information
Create controller
Evaluate presence of extra constraints in controller design
Output: Yes/ No
Decrypt cipher text using RSA
Signature Verification
Decode the decrypted signature using knowledge of encoding rules
Re-construct colored interval graph to realize the extra edges as watermarking constraints
Yes →Received IP Valid
No→Received IP compromised

Fig. 7: Signature Detection Process

C. Properties of Watermark generated

The watermark generated through the proposed approach satisfies all the following desired properties:

a) *Minimization of embedding cost:* The final watermarked solution generated is a product of PSO-driven exploration which considers minimization of hardware area and latency.

b) *Resiliency against attacks:* Our generated watermark is robust against typical attacks. This is because our watermark is based on multi-variable (4 variables) signature encoding, robust embedding process (i.e. distributing watermark constraints all over the design) and RSA encryption.

c) *Watermark Fault Tolerance:* In case part of the watermark is removed by an attacker, ownership still remains preserved because the watermarking constraints are distributed throughout the design.

d) *Watermark creation time and signature detection time:* In our proposed approach, the time taken to embed a watermark is very less. Besides, signature detection is straightforward for a genuine person who has full knowledge of encoding rules and encryption key, but is extremely tedious for an attacker.

IV. Experimental Results and Analysis

The proposed approach and [1] were both implemented in java and run on Intel Core-i5-3210M CPU with 3MB L3 cache memory, 4GB DDR3 memory at 2.5 GHz. [1] was chosen for comparison, because no other work exists in literature after this, with a similar objective of protecting IP core using watermark at behavioural level.

A. *Comparison of proposed watermarking with recent work*

As evident from Table IV, for same signature strength (i.e. watermarking constraints 'w') watermarked solution obtained by [1] is higher is embedding cost than proposed approach. This is because in [1], amongst the numerous competitive design solutions present, exploration of low cost optimal watermark was not performed. However, the proposed approach explores a low cost watermarked solution. Table IV reports the number of hardware units and registers required for implementing the watermarked solution. The proposed approach explores a low cost watermarked solution.

Table V reports the comparison of storage hardware (registers) in final watermarked solution with respect to the same signature strengths (watermark constraints) with reference [1]. The proposed approach results in watermarked solution with lesser storage overhead to minimize the embedding cost. This is due to the optimization of hardware area during exploration of an optimal watermarked solution.

Table VI reports the probability of coincidence measured as the probability of generating the same colored solution with the signature and it indicates the proof of authorship (strength/quality of the watermark) of the watermark generated. The function for evaluation of this metric defined as [1]: $P_c = (1 – 1/c)^w$ (4)

where, P_c = the probability of coincidence, c = number of colors used, w = number of watermarking constraints (strength of the signature in terms of number of digits used). It is observed that as the signature strength increases, the probability of coincidence decreases which indicates that with increase in signature strength, proof of authorship is stronger.

V. Conclusions and Future Directions of Research

This paper presented a novel solution to protection of IP core through a low cost robust watermarking technique. The proposed IP core protection watermarking technique can be extended to other HLS steps to assess its ability of providing security against typical attacks compared to register allocation.

References

[1] F. Koushanfar, I. Hong, and M. Potkonjak. 2005, "Behavioral Synthesis Techniques for Intellectual Property Protection," *ACM Trans. Des. Autom. Electron. Syst.*, Vol. 10, No. 3, July 2005, 523–545.

[2] I. Hong and M. Potkonjak, "Behavioral synthesis techniques for intellectual property protection," *in Proc. of the 36th annual ACM/IEEE Design Automation Conference*, 1999, pp.849–854.

[3] S. Meguerdichian and M. Potkonjak, "Watermarking while preserving the critical path," *in Proc. of 37th ACM/IEEE DAC*. 2000, pp.108–111.

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

[5] E. Charbon, "Hierarchical watermarking in IC design," in *Proc. of IEEE Custom Integrated Circuits Conf.*, 1998, pp. 295–298.

[6] A. Sengupta, V.K. Mishra , "Integrated Particle Swarm Optimization (i-PSO): An Adaptive Design Space Exploration Framework for Power-Performance Tradeoff in Architectural Synthesis", *Proc. of IEEE 15th Intl Symp. on Quality Electronic Design*, USA, 2014, pp.60 - 67.

[7] D.L. Irby, R.D. Newbould, J.D. Carothers, J. J. Rodriguez, and W.T. Holman, "Placement watermarking of standard-cell designs in Mixed-Signal Design," in *Proc. of the SSMSD*, 2001, pp. 116–120.

[8] S. P. Mohanty, N. Ranganathan, V. Krishna, "Datapath Scheduling Using Dynamic Frequency Clocking", in *Proceedings of the IEEE Computer Society Annual Symposium on VLSI*, 2002, pp. 58-63.

[9] A. Sengupta and S. Bhadauria, "Untrusted Third Party Digital IP cores: Power-Delay Trade-off Driven Exploration of Hardware Trojan Secured Datapath during High Level Synthesis", *Proc. of 25th IEEE/ACM Great Lake Symposium on VLSI,* 2015, 167 – 172.

[10] Castillo, E., Meyer-Baese, U., Garcia, A., Parrilla, L., et al., IPP@HDL: Efficient Intellectual Property Protection Scheme for IP Cores, *IEEE Trans in VLSI Syst.*, Vol 15, No 5., 2007.

Table IV: Comparison of proposed watermarked solution and cost with [1]
(Number of watermark constraint (w) = 15; p = 3; ω = [0.9 – 0.1]; b1, b2 = 2)
Note: Avg. runtime ~ 1.4 secs

Bench mark	**Proposed Watermarked Solution**		**Watermarked Solution for [1]**		**Cost of Watermarked Solution [f(A$_T$, L$_T$)]**	
	FU's	Registers	FU's	Registers	Proposed	[1]
DWT	1(+), 3(*)	6	2(+), 3(*)	5	-0.01	0.04
ARF	2(+), 4(*)	8	4(+), 2(*)	8	-0.21	0.02
MPEG	2(+), 5(*)	14	3(+), 7(*)	14	-0.44	-0.36
IDCT	4(+), 2(*)	8	4(+), 2(*)	8	0.08	0.08
MESA	3(+), 8(*)	48	9(+), 16(*)	48	-0.49	-0.38

Table VI: Measuring probability of coincidence (P_c) as strength of watermark
*Note: **S**(NW) = Number of storage hardware in non-watermarked solutions*

Benchmark	**number of storage variables**	**S(NW)**	**P_c**			
			number of watermarking constraints (w)			
			15	**30**	**60**	**120**
DWT	22	5	0.03	1.23×10^{-3}	1.53×10^{-6}	2.3×10^{-12}
ARF	36	8	0.13	0.01	3.3×10^{-4}	1.09×10^{-7}
IDCT	50	8	0.13	0.01	3.3×10^{-4}	1.09×10^{-7}
MESA	139	48	0.72	0.53	0.28	0.07
MPEG	42	14	0.32	0.10	0.01	1.37×10^{-4}

Table V: Comparison of storage hardware used in a watermarked solution for various signature strengths (watermarking constraint 'w')
Note: S(NW) = Number of storage hardware in non-watermarked solutions

Benchmark	**# of storage variables**	**S(NW)**	**# of storage hardware in watermarked solutions**							
			w = 15		**w = 30**		**w=60**		**w=120**	
			Proposed	**[1]**	**Proposed**	**[1]**	**Proposed**	**[1]**	**Proposed**	**[1]**
DWT	22	5	6	6	7	7	10	10	NA	NA
ARF	36	8	8	9	9	10	11	11	NA	NA
IDCT	50	8	9	9	9	10	10	11	16	16
MESA	139	48	48	48	48	48	48	48	50	50
MPEG	42	14	14	15	14	15	15	16	21	21