DSP Design Protection in CE through Algorithmic Transformation Based Structural Obfuscation

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

***Abstract*—Structural obfuscation offers a means to effectively secure through obfuscation the contents of an intellectual property (IP) cores used in an electronic system-on-chip (SoC). In this work a novel structural obfuscation methodology for protecting a digital signal processor (DSP) IP core at the architectural synthesis design stage. The proposed approach specifically targets protection of IP cores that involve complex loops. Five different algorithmic level transformation techniques are employed: loop unrolling, loop invariant code motion, tree height reduction/increment, logic transformation and redundant operation removal. Each of these can yield camouflaged functionally equivalent designs. In addition, low cost obfuscated design is generated through proposed approach through the use of multi-stage algorithmic transformation and particle swarm optimization (PSO)-drive design space exploration (DSE). Results of proposed approach yielded an enhancement obfuscation of 22 % and reduction in obfuscated design cost of 55 % compared to similar prior art.**

***Index Terms*—Digital signal processing (DSP) core, high-level transformation, IP protection, structural obfuscation**

I. Introduction

SYSTEM-ON-CHIP (SoC) integrated circuits as employed in today's consumer electronic devices comprise of multiple major system modules such as memory (SRAM, Flash), custom processor/co-processor, A-to-D converter, DSP engines, A/V codecs, wireless modems, etc. Practically all modern consumer electronics devices ranging from smart phone, tablets, set-top box, smart TV, home gateways & routers, smart kitchen appliances and most recently smart-speakers are based on SoC ICs. In such devices, the main requirements for IP cores within the SoC are to have an optimized silicon area, thus lowering the manufacturing cost, and to operate at low power.

This work was financially supported by Council of Scientific and Industrial Research (CSIR) under sanctioned grant no. 22/730/17/EMR-II.

A. Sengupta and D. Roy are with the Discipline of Computer Science and Engineering, Indian Institute of Technology, Indore, 453552, India (e-mail: asengupt@iiti.ac.in; phd1501201007@iiti.ac.in).

S. P. Mohanty is with Department of Computer Science, University of North Texas, Denton, Texas (e-mail: Saraju.Mohanty@unt.edu).

P. Corcoran is with the College of Engineering & Informatics, National University of Ireland Galway (e-mail: peter.corcoran@nuigalway.ie).

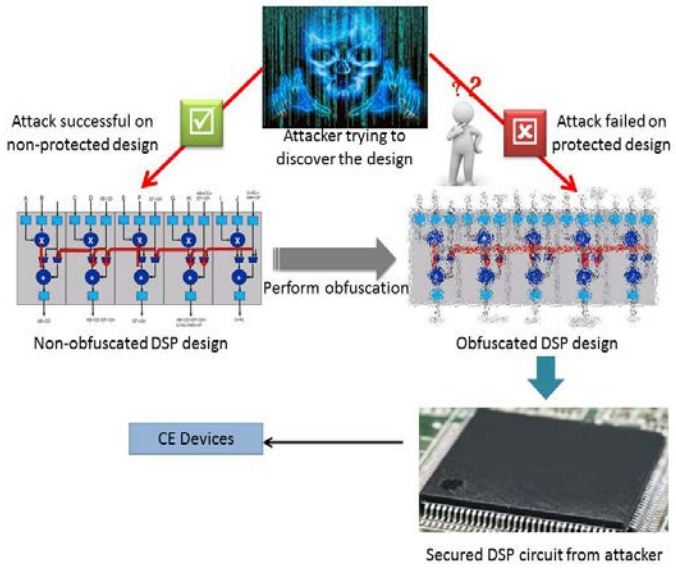

Fig. 1. A thematic representation of secured DSP circuit via obfuscation

Thus at the heart of every SoC is a DSP kernel which is an example of an IP core architecture that involves complex loops. Other CE architectures - e.g. camera pipelines, codecs, wireless modems, will share similar complex-loop structures. Future SoCs in CE devices are expected to perform sophisticated tasks such as browsing internet anywhere anytime on the globe, that would require usage of DSP engines most of the time [1] [2], [3], [4].

Algorithmic synthesis automates the design process of a DSP IP and generates register-transfer level (RTL) hardware description language (HDL) that implements the DSP designs behavior [5], [6]. However the sophistication of today's design tools that simplify algorithmic synthesis also provide sophisticated reverse engineering capabilities that enable the design of an IP core to be discerned with relative ease. Thus a significant emerging threat for IP core designers is piracy, where reverse engineering enables direct theft/copying of the IP for reuse without permission. It is most likely that the thief will 'silently' re-use the IP to save design/test cost [7].

A typical attack scenario and its protection mechanism for DSP application is shown in Fig. 1. Thus, key to a successful DSP datapath design in a portable CE is security against threats such as reverse engineering [2], [8], [9], [10], [11], [12], [13], [14]. The present paper introduces a novel multi-

stage high-level transformation driven obfuscation for DSP IPs used in CE devices.

The rest of this paper is organized as follows: In section II we elaborate novel contributions of this paper. Section III discusses the major related approaches, while Section IV describes our proposed low cost algorithmic (high-level) transformation (HLT) based obfuscation methodology. Demonstration of proposed approach is explained in Section V. Further, experimental results are presented in Section VI, followed by conclusion in Section VII.

II. Novel Contributions

The key contributions of this paper are:

- Novel obfuscation-based methodology at algorithmic (architectural) level of DSP architecture for loop-based control intensive applications.
- Introducing the use of several loop-based high-level transformations to enhance the obfuscation complexity.
- Methodological optimization to meet important design constraints such as hardware area, system latency and overall design cost.
- Proposes obfuscation-based methodology for loop-based control intensive applications which offers higher security in terms of Power of Obfuscation (PoO) than other similar approach.

Threat Model: The proposed work enhances the reverse engineering complexity for an adversary during RTL synthesis by hiding the structure of an IP design. Hence, provide protection against IP piracy and Trojan insertion.

III. Related Prior Research

Consumer electronics literature has several works dealing with various aspects of digital signal processor [15], [16], [17]. They focus on different aspects of consumer electronics system characteristics such as energy efficiency, high performance, and area efficiency. However, they miss the critical axis of CE system (i.e. the "security and IP protection"); thus do not comprehensively addressing the critical CE design issues in the current social network driven era in which the cyber-security is a critical design axis.

A. Obfuscation-based approach

Obfuscation is the process of transforming an original application or design into its functionally equivalent form to make the reverse engineering process significantly more challenging [11]. Obfuscation of a DSP IP can be achieved in two ways: a) logic obfuscation, b) structural obfuscation. Logic obfuscation hides the implementation of the DSP design by inserting additional component into it. Unlike logic obfuscation structural obfuscation hides the functionality of the DSP design through transformations, randomized placement of logic elements [18]. An implementation of software-defined digital Video broadcast (DVB)-T2 modulator and DVB-H receiver and gateway based is performed based on DSP processor in [2] and [3] respectively. None of these approaches hide the functionality of the DSP IP to minimize reverse engineering attacks. However, the proposed approach obfuscates a DSP design through a series of high-level transformation techniques at low design cost.

1) Source code-based obfuscation: To prevent an adversary from understanding the HDL code, the code is either transformed [19] into a more complex form or the source code is encrypted using some cryptographic techniques [20]. In [19], VHDL (Very High Speed Integrated Circuit Hardware Description Language) code obfuscation is performed using different transformation techniques which are harder to reverse engineer. In [20], encryption of the source code is achieved based on the Dead Code Insertion technique on a Crypter. However, in the proposed approach unlike [19] and [20] low-cost structural obfuscation of DSP design is achieved through multiple HTLs.

2) Logic obfuscation: According to [21] logic obfuscation can be classified into two types: a) sequential, b) combinational. In sequential logic obfuscation, additional invalid or blocking states are inserted into the finite state machine (FSM) of a design [22] [23]. The FSM is constructed in such a way that the design executes properly and reaches a valid state if the correct key is applied. In combinational logic obfuscation additional XOR/XNOR gates are introduced into the circuit to protect the IP core [24] [25] [26] [27], [28]. All these key-based obfuscation techniques increase the chance of higher design overhead due to insertion of additional components. Furthermore, [22] [23] [24] [25] [26] do not apply any optimization technique to achieve low-cost functionally obfuscated design. On the contrary, proposed approach protects DSP design by employing multiple transformation-based structural obfuscation.

3) Structural obfuscation: In [11], [29] structural obfuscation is performed for DSP circuits using. However [11] has not performed multi-stage HLTs such as: ROE, LT, THT, LICM, loop Unroll for DSP designs. Additionally, [11] has not explored low-cost obfuscated design out of various structural obfuscated design. Moreover, no folding factor calculation methodology is explained in [11] that leads to higher design overhead. Further, [29] has not handled loop-based CDFG applications for DSP. Therefore, no loop-based HLT techniques are applicable to obfuscate the design. Moreover, no equivalent DSP circuit of obfuscated design is generated during synthesis. Our proposed approach performs low-cost, compiler-driven, multi-stage HLT techniques for loop-based CDFG to achieve structural obfuscation during algorithmic synthesis. The optimal cost is achieved through PSO driven design space exploration process.

B. Authentication-based approach

In digital symmetrical [30] based IP protection mechanisms IP seller's signature nullifies the false claim of ownership, thus protects against IP infringement [31] and IP buyer's signature traces illegally resold/overbuilt copies of IP core by a non-trustworthy IP seller, and thus provides exclusive user rights. Computational forensic engineering (CFE) is a non-signature based IP protection mechanism. It tries to identify whether the generated IP is coming from a familiar source or not [32]. IP metering [23], is another non-signature based IP core protection mechanism which detects overbuilt/duplicate copies

of an IP core. In hardware metering [33], a uniquely generated ID is inserted programmatically into the IP core design, which helps differentiate between a genuinely manufactured IP and its duplicate/unauthorized copy. All these aforesaid mechanisms of IP protection are authentication-based approaches. The limitation of these techniques is that these passive protection methods are only capable of tracing the illegal copies of IP but cannot prevent from being stolen [22].

IV. Obfuscated IP Core Design Methodology

The proposed approach hides the functionality of the application (IP) from an adversary during algorithmic synthesis. The output of algorithmic synthesis is structurally obfuscated RTL description (datapath and controller designs) of a reusable IP core. The obfuscation of the design is achieved via high-level transformation techniques in multiple stages that are difficult to reverse engineer.

A. Overview of Proposed Methodology

As shown in Fig. 2 proposed approach takes the loop-based CDFG as input and obfuscate the original design using five high-level transformation techniques i.e. Redundant Operation Elimination (ROE), Logic Transformation (LT), Tree Height Transformation (THT), Loop Unrolling (LU) and Loop Invariant Code Motion (LICM). The obfuscated design is further provided as an input to perform particle swarm optimization (PSO) driven design space exploration (DSE). Finally, a structurally obfuscated low-cost IP is generated which satisfies the user-given area-delay constraints. The detail HLT based multistage obfuscation and PSO driven DSE process is explained in Section IV-C and IV-D respectively.

B. Problem Definition and Evaluation Models

1) Problem Definition: Given a CDFG for loop based application, explore the design space to determine an optimal structurally obfuscate design solution. The generated solution should minimize the overall design cost while satisfying conflicting user-given constraints. The problem can be formulated as follows:

Minimize: Obfuscated design cost (A_T^{OBF}, T_E^{OBF}), for optimal X_i.

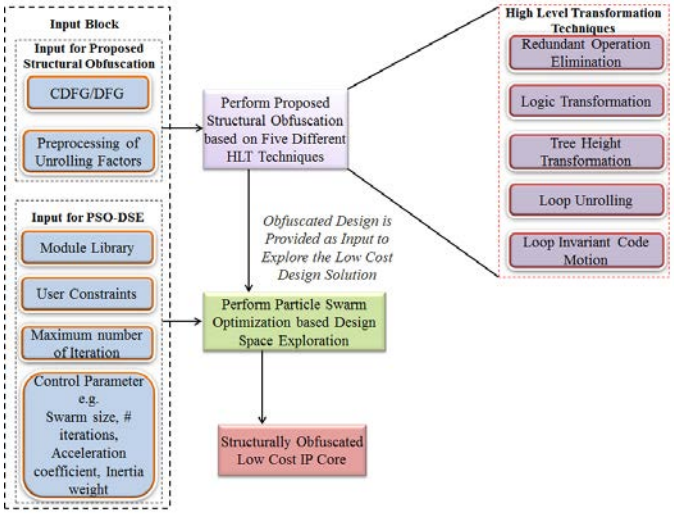

Fig. 2. Proposed low-cost obfuscation methodology

Subject to: $A_T^{OBF} \leq A_{cons}$ and $T_E^{OBF} \leq T_{cons}$ and IP protection through structural obfuscation. Where X_i is a resource set of a particle solution with unrolling can be represented as:

$$X_i = \{N(R_1), N(R_2), .. N(R_D), UF\} \quad (1)$$

Where, $N(R_D)$ is the number of resource type R_D; UF is the unrolling factor for loop-based application; A_T^{OBF} and T_E^{OBF} are the total hardware area and total execution time taken by the obfuscated design respectively. A_{cons} and T_{cons} are the user specified area and time constraints respectively.

2) Area Model: Total area A_T^{OBF} consumed by structurally secured obfuscated design is adopted from [30], can be expressed *as*:

$$A_T^{OBF} = \sum_{i=1}^{m} A(R_i) * N(R_i) + A(mux) * N(mux) + A(buffer) * N(buffer) \quad (2)$$

Total area is the sum of area occupied by the hardware resources, interconnecting units and storage units. Where, $A(R_i)$, A(mux) and A(buffer) are the area of i^{th} hardware resource, the mux/demux and the storage unit respectively; $N(R_i)$, N(mux) and N(buffer) are the number of unit of i^{th} hardware resource, mux/demux and storage unit respectively.

3) Delay Model: Total latency T_E^{OBF} of the obfuscated design is partially adopted from [34], can be shown:

$$T_E^{OBF} = (T_{body}^{OBF} * \left\lfloor \frac{I}{UF} \right\rfloor) + (I \quad Mod \quad UF) * T_{first}^{OBF} \quad (3)$$

In the above expression, T_{body}^{OBF} is the delay to execute the loop body of obfuscated CDFG once; I is the maximum number of iteration (loop count); T_{first}^{OBF} is the execution delay of first iteration of obfuscated CDFG.

4) Fitness Function: The fitness of each solution is calculated (considering hardware area consumption and execution delay) on the basis of following fitness function:

$$C_f(X_i) = \Phi_1 \frac{A_T^{OBF} - A_{cons}}{A_{\max}^{OBF}} + \Phi_2 \frac{T_E^{OBF} - T_{cons}}{T_{\max}^{OBF}} \quad (4)$$

Where, $C_f(X_i)$ is the cost of the solution X_i; A_{max}^{OBF} and T_{max}^{OBF} indicate the maximum area and execution delay of obfuscated design respectively; A_{cons} and T_{cons} are the user specified area and delay constraints respectively. $Ø_1$ and $Ø_2$ are user defined weights for area and delay respectively, the value lies between 0 to 1 (*Note: Both* $Ø_1$ and $Ø_2$ *are kept 0.5 to provide equal preference during exploration.*)

C. Process of Proposed Obfuscation Methodology

As shown in Fig. 3, proposed multi-stage HLT driven structural obfuscation methodology during algorithmic synthesis is achieved through five different compiler-based HLT techniques. They are (1) Redundant Operation Elimination (ROE) (2) Logic Transformation (LT) (3) Tree Height Transformation (THT) (4) Loop Unrolling (LU) (5) Loop Invariant Code Motion (LICM). Our approach takes the input of an application in the form of a loop-based CDFG and applies each of the aforementioned HLT to obfuscate it.

Fig. 4 shows an example (sample) non-obfuscated original design used for demonstration. The design is scheduled based on 3 adders and 4 multipliers taken as a user defined input. Fig. 5 shows the scheduled CDFG of the non-obfuscated

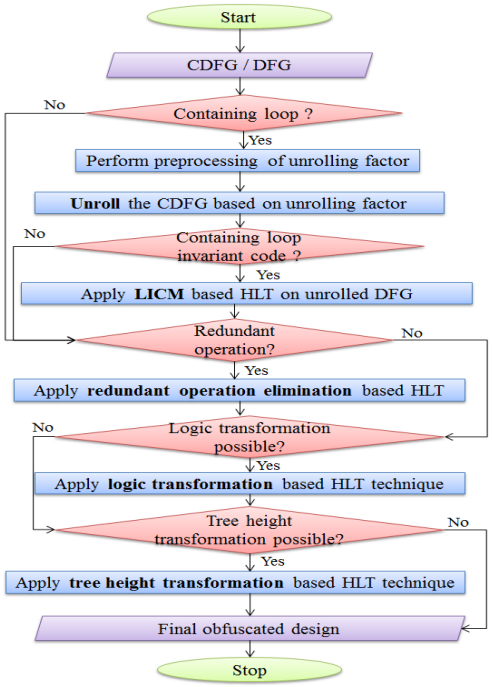

Fig. 3. Proposed multi-stage HLT based structural obfuscation methodology

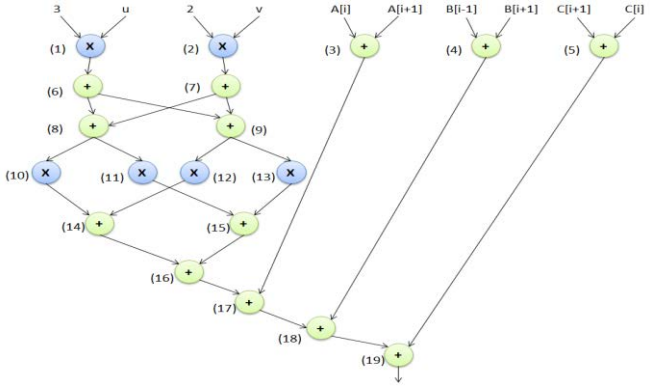

Fig. 4. Original non-obfuscated loop-based CDFG

design. Equivalent DSP circuit is shown in Fig. 6. It employs 6 4:1 muxes, 4 2:1 muxes, 10 input registers, 1 output register and 8 delay elements. The detailed process of each HLT technique is explained with a proper example in the following subsections.

Redundant Operation Elimination Process: An HLT technique which is applied to obfuscate the input CDFG by removing redundant nodes from the graph is redundant operation elimination. A node in the input graph is identified as a redundant node if there exist another node which has exactly same parents/inputs and same operation type. In our proposed approach we scan each node based on the node numbers in ascending order. If a pair of nodes is found which have same inputs and operation type then the node having higher node number is identified as a redundant node. These nodes are deleted from the graph and necessary adjustment is performed to maintain the correct functionality of the graph. Finally, ROE based structurally obfuscated graph is produced as output. For example, in the original design shown in Fig. 4 the redundant operation is node 9, 11, 12, 13, 15 which is eliminated through the proposed approach as shown in Fig. 7 to structurally obfuscate the design. To maintain the

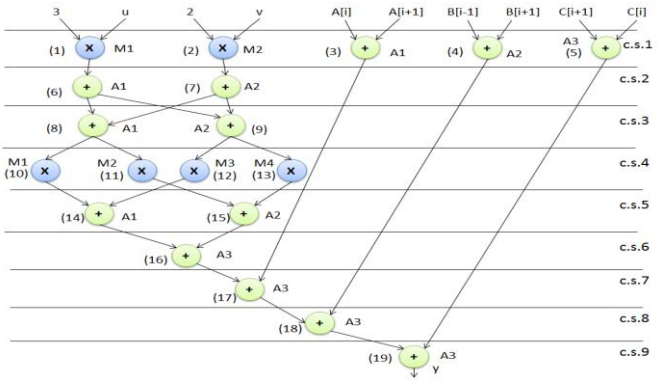

Fig. 5. Original non-obfuscated scheduled CDFG based on 3(+), and 4(*)

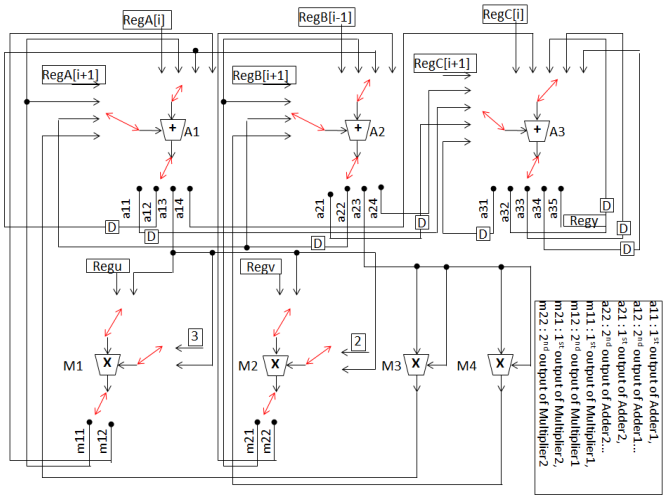

Fig. 6. Equivalent DSP circuit of non-obfuscated design

correctness of the output the inputs of nodes 14 and 16 are taken from node 10 and 14 respectively in the ROE-based obfuscated design (Fig. 7).

1) Logic Transformation Process: Another HLT technique which is applied to obfuscate the input CDFG by modifying the graph with different logically equivalent function is logic transformation. It alters the nodes of the input graph such that the graph looks different than the original still satisfies the functionality correctly. Finally, LT based structurally obfuscated graph is produced as output.

For example, Fig. 8 represents the logic transformation driven obfuscated form of the input graph (Fig. 7); newly added/modified nodes are marked with green colored node number and the modified dependencies are marked with green

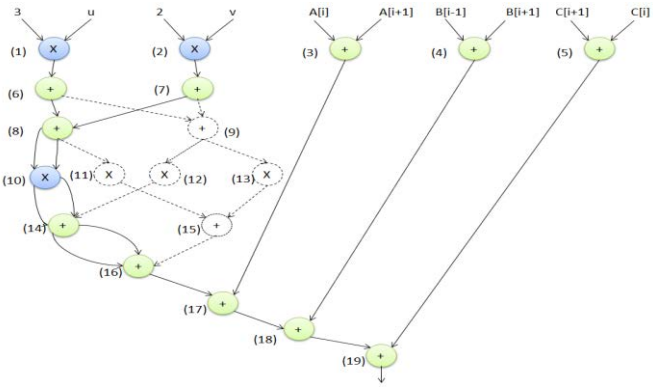

Fig. 7. Redundant operation elimination based obfuscated design

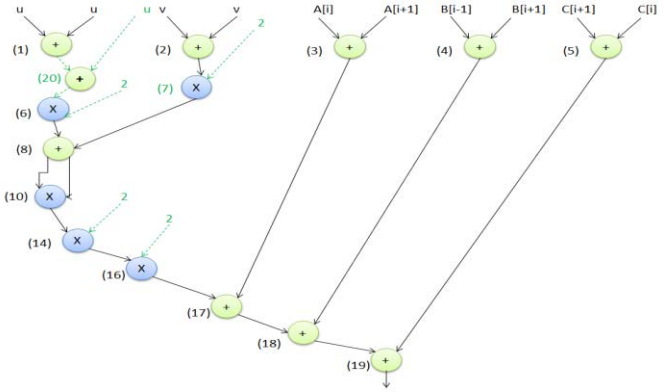

Fig. 8. Logic transformation based obfuscated design

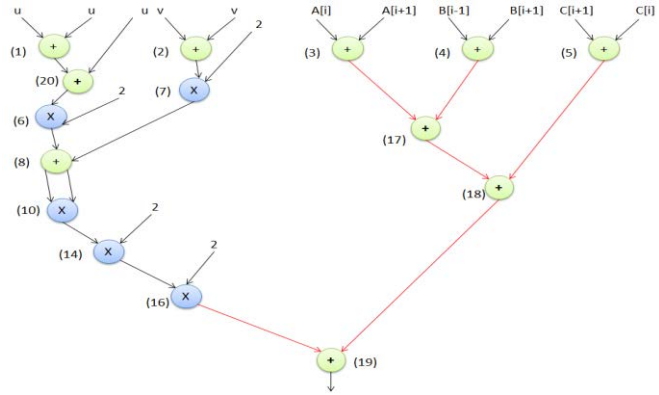

Fig. 9. Tree height transformation based obfuscated design

dotted line. However, as both the design is functionally equivalent there by produce same outputs.

2) Tree Height Transformation Process: Another HLT technique which is responsible for obfuscating the input CDFG by increasing or decreasing the height of the graph is tree height transformation. It divides the critical path dependency into temporary sub-computations and evaluates in parallel, thereby generates structurally dissimilar yet functionally equivalent graph. Finally, THT based structurally obfuscated graph is produced as output.

For example, in Fig. 9, THT-based structurally obfuscated form of the input graph (shown in Fig. 8) is shown. It reduces the height of the obfuscated graph from 10 to 8 compared to the original design. The computation of node 17 and 18 in obfuscated design is executed prior to the input design. The dependencies of the obfuscated graph are adjusted to maintain the correct functionality. The modified dependencies are marked with red lines.

3) Loop Unrolling: Loop transformation based HLT technique which is applied to obfuscate the input CDFG by unwinding the loop is loop-unrolling. Unrolling of Loops can be achieved by repeating the same loop body in multiple sequences to improve execution delay. Finally, loop unrolls based structurally obfuscated graph is produced as output.

For example, Fig. 10 unrolls the input loop-based CDFG two times. In the proposed approach the optimal unrolling factor is explored based on PSO driven DSE. Loop unrolling minimize the execution time simultaneously obfuscate the design.

4) Loop Invariant Code Motion: Another loop-based HLT technique which is used in our proposed approach to obfuscate

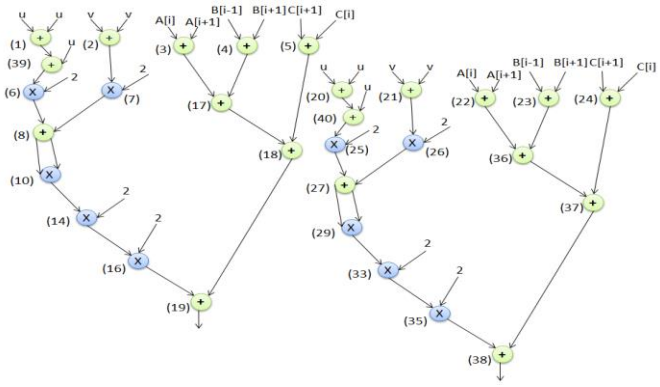

Fig. 10. Loop unroll based obfuscated design (unrolled twice)

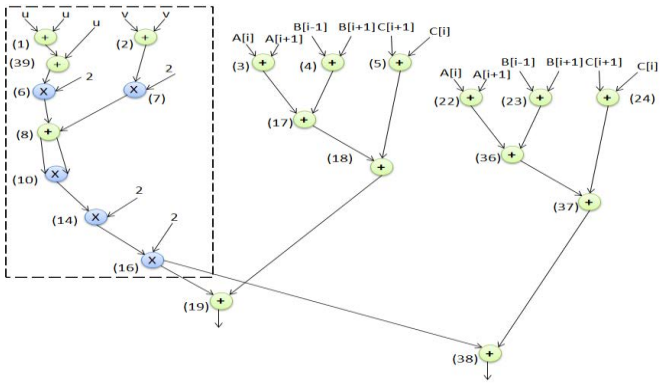

Fig. 11. Loop invariant code motion based obfuscated design (unrolled twice)

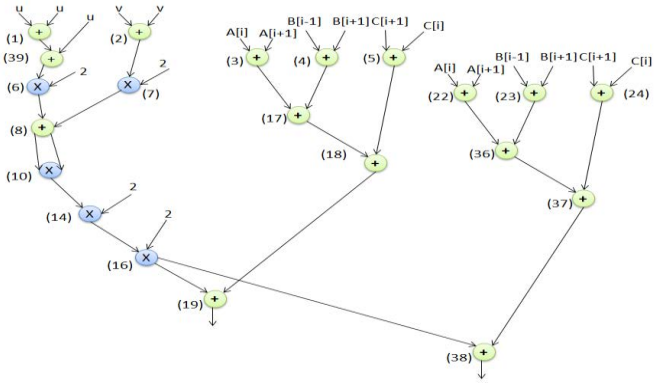

Fig. 13. Obfuscated design using multi-stage driven HLTs

the input CDFG by moving the loop independent nodes out of the loop body is loop invariant code motion. It moves out the nodes of the loop which would not make any differences if it performs inside the loop iteratively or outside the loop once. Thereby it speeds up the execution process while maintaining the correct functionality of the graph. Finally, LICM based obfuscated graph is generated as output.

For example, Fig. 11 shows the LICM-based structurally obfuscated form of the input graph (shown in Fig. 10). The dotted box shows the nodes which are not depend on the loop. According to the Fig. 11, for unrolling factor 2 loop invariant operations are executed once while loop depended operations are executed two times.

To obtain higher robustness during obfuscation, all the aforementioned high-level transformation techniques have performed in consecutive stages (refer Fig. 3).

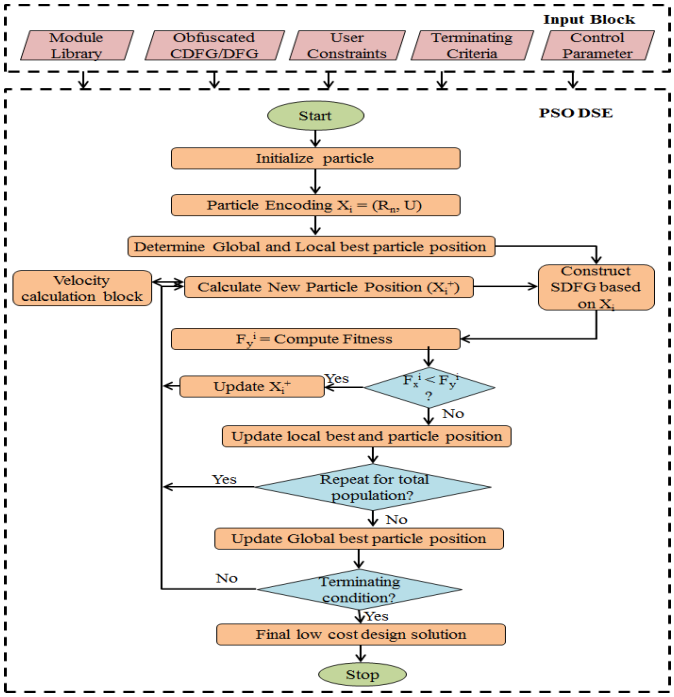

Fig. 12. PSO driven DSE process for optimal obfuscated design

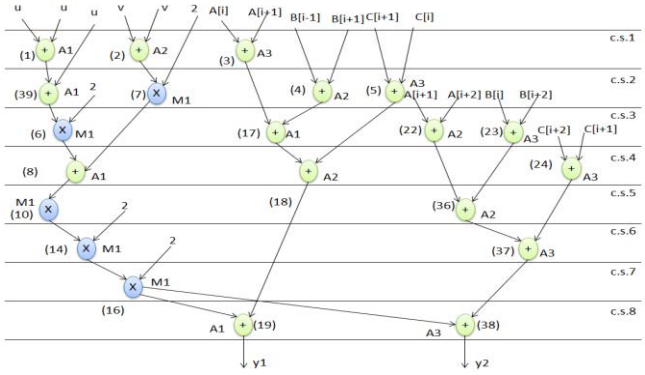

Fig. 14. Low-cost obfuscated IP design scheduled with 3(+) and 1(*)

D. Exploring PSO-driven Low-Cost Obfuscated Design

In our proposed approach low-cost obfuscated IP design is achieved through particle swarm optimization driven design space exploration. It accepts the obfuscated design (explained in Section IV-C) of the application in the form of loop-based CDFG, module library, terminating criteria of PSO, control parameters (like inertia weight, social factor, cognitive factor etc.) and user given area-delay constraints as inputs and generates a low-cost optimized obfuscated IP design as output. The detailed PSO-DSE process shown in Fig. 12 is explained in the following sub sections.

1) Background on Particle Swarm Optimization Methodology: PSO is a population-based stochastic optimization methodology where each single solution is known as a particle. The fitness of each particle is evaluated based on the fitness function to be optimized. The velocity of each particle directs the movement of the particle. The particles move through the search space by following the current global best 'gbest' and its own best location 'lbest'.

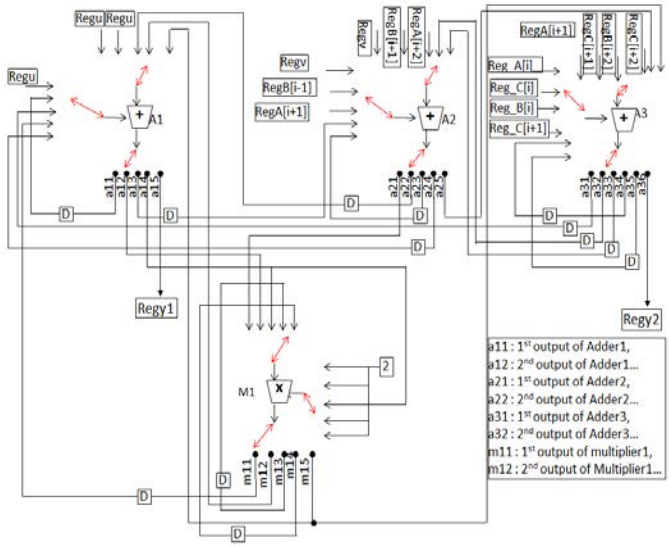

Fig. 15. Equivalent DSP circuit of obfuscated IP design

After finding a better 'gbest' or 'lbest' the i^{th} particle updates its velocity and position.

2) Movement of Particle using Velocity: In the PSO-DSE process [34], each dimension (d) of a particle velocity (V_{di}) is updated based on the following equation:

$$V_{d_i}^{+} = \omega V_{d_i} + b_1 r_1 \left[R_{d_{lbi}} - R_{d_i}\right] + b_2 r_2 \left[R_{d_{gb}} - R_{d_i}\right] \qquad (5)$$

Where, V_{di}^+ and V_{di} are new and current velocity of i^{th} particle in d^{th} dimension respectively; R_{di} is resource value/unrolling factor of i^{th} particle in d^{th} dimension; R_{dlbi} is local best position of i^{th} particle in d^{th} dimension; and R_{dgb} is global best of d^{th} dimension.

3) Terminating Criteria: The PSO based DSE process will terminate if any of the three conditions arise a) reached the maximum number of iteration (I_{max}), b) no improvement is observed in global best for δ number of iteration, c) velocity value becomes zero. In our approach the value of I_{max} and δ is taken as 100 and 10 respectively.

V. Demonstration of Proposed Methodology

For the demonstration of the proposed approach, an original, non-obfuscated CDFG (shown in Fig. 4) is taken as an input. Blue nodes represent multiplier and green nodes represent adder in the graph. The integer value beside each node indicates the corresponding node number. As shown in Fig.4, primary inputs of the graph. A, B and C represents the loop dependent variables u, v represent loop independent variables and the rest are the constant value. The total number of node in the graph is 19 before unrolling, the number of loops it consists is 8 and the height of the graph is 9. The obfuscated design using multi-stage driven HLTs is shown in Fig. 13. The loop-based CDFG is unrolled two times, the loop invariant nodes (marks as black dotted box) are kept outside the loop, node 9, 11, 12, 13, 15 is deleted due to redundancy, node 6, 7, 14, 16, 39 is modified due to logic transformation, height of the graph is reduced from 9 to 8 and the total number of nodes is increased from 19 to 21. This obfuscated design is

TABLE I
Results for low-cost optimal obfuscated solution through proposed approach for different swarm size

Benchmark	Particle Size	Design Solution	Obfuscated Design Area(μm^2)	Obfuscated Design Latency(ps)	Obfuscated Design Cost
2D Autoregression Lattice Filter(ARF)	3	4A, 2M	241.83	1900.80	-0.37
	5	4A, 2M	241.83	1900.80	-0.37
	7	4A, 2M	241.83	1900.80	-0.37
FIR (6-tap)	3	2A, 2M,1C, 8UF	214.99	1477.85	-0.36
	5	2A, 2M,1C, 8UF	214.99	1477.85	-0.36
	7	2A, 2M,1C, 8UF	214.99	1477.85	-0.36
AUTO-CORRELATION	3	4A, 8M,1C, 8UF	726.17	3557.15	-0.46
	5	7A, 8M,1C, 8UF	800.68	2940.45	-0.47
	7	7A, 8M,1C, 8UF	800.68	2940.45	-0.47
DIFFERENTIAL EQUATION	3	2A, 3M, 1C, 16UF	330.99	4498.80	-0.33
	5	2A, 3M, 1C, 16UF	330.99	4498.80	-0.33
	7	2A, 3M, 1C, 16UF	330.99	4498.80	-0.33
DHMC	3	4A, 4M, 1C, 6UF	419.07	8581.61	-0.63
	5	4A, 4M, 1C, 6UF	419.07	8581.61	-0.63
	7	4A, 4M, 1C, 6UF	419.07	8581.61	-0.63
Adaptive Filter(noise cancellation)	3	4A, 1S, 1M, 1C, 30UF	270.83	3120.01	-0.63
	5	4A, 1S, 1M, 1C, 30UF	270.83	3120.01	-0.63
	7	4A, 1S, 1M, 1C, 30UF	270.83	3120.01	-0.63
Adaptive Filter(Least mean square)	3	5A, 1S, 1M, 1C, 30UF	292.26	6833.58	-0.70
	5	5A, 1S, 1M, 1C, 30UF	292.26	6833.58	-0.70
	7	5A, 1S, 1M, 1C, 30UF	292.26	6833.58	-0.70

further used as input with other necessary inputs (refer to Fig. 12) to perform PSO driven DSE process to explore the optimal obfuscated IP design. Fig. 14 represents the final low-cost obfuscated IP design scheduled based on 3 adders and 1 multiplier. Equivalent obfuscated DSP circuit is shown in Fig. 15. It employs 8 8:1 muxes, 18 input registers, 2 output register and 13 delay element.

VI. Experimental Results

Our proposed approach provides a low-cost, robust DSP IP core protection through multi-stage based HLT driven obfuscation methodology during algorithmic synthesis.

A. Experimental Setup and Benchmark

The proposed approach, non-obfuscated design and [11] are implemented in Java 8 and executed on a computing platform with 4GB DDR3 primary memory and processor frequency of 3.20 GHz. A 15nm technology scale based on NanGate is used to evaluate both the area and the latency of an IP design [35]. During multi-stage obfuscation process different HLTs are applied on loop-based CDFGs in the following order: 1) Loop Unroll, 2) LICM, 3) ROE, 4) LT and 5) THT. During PSO-DSE process both $Ø_1$ and $Ø_2$ are kept 0.5 to provide equal preference as both silicon area and latency is equally essential for a DSP design of a CE device. For a broader range of optimization a designer may tune the weighing factors to unequal values. However, this would be applicable when a designer assigns more priority to silicon area than latency or vice-versa. Further, following optimal settings from [34] is used for PSO framework: ω (inertia weight) = linearly decreasing between 0.9 to 0.1; b1 and b2 (acceleration coefficient) = 2; r1 and r2 (random numbers) = 1; I_{max} = 100 or δ = 10 as stopping criterion; swarm size p = 3 or 5 or 7.

TABLE II
Measuring power of obfuscation for each HLT technique

Benchmark	p^{obf} for ROE	p^{obf} for LT	p^{obf} for THT	p^{obf} for UF	p^{obf} for LICM	Total p^{obf}
ARF	0.32	0.61	0	-	-	0.55
FIR	0	0	0.50	1	-	0.75
AUTO-CO	0	0	0.49	1	-	0.75
DIF-EQN	0	0.44	0	1	-	0.72
DHMC	0	0.38	0	1	-	0.69
Ada-NC	0	0.67	0.03	1	-	0.57
Ada-LMS	0	0.75	0	1	-	0.88

TABLE III
Comparison of proposed obfuscated design with non-obfuscated design

Benchmark	Original Design (non-obfuscated)			Proposed Obfuscated Design		
	Area (μm^2)	Latency (ps)	Cost	Area (μm^2)	Latency (ps)	Cost
ARF	241.8	2573.5	-0.26	241.8	1900.8	-0.37
FIR	215.0	1661.3	-0.31	215.0	1477.8	-0.35
AUTO-CO	736.4	3399.2	-0.48	736.4	2940.5	-0.46
DIF-EQN	333.0	7246.5	-0.18	333.0	4498.8	-0.33
DHMC	419.1	8872.1	-0.27	419.1	8581.6	-0.63
Ada-NC	270.8	3330.98	-0.62	270.8	3120.01	-0.63
Ada-LMS	292.3	12561.3	-0.61	292.3	6833.58	-0.70

B. Result of Proposed Approach in terms of Design Cost and Security Metric

Table I shows the impacts on obfuscated design area, latency and cost with the variation of swarm size. As seen from Table I except Auto Correlation benchmark no improvement in quality of solution is found. However, for Auto Correlation benchmark design cost is -0.4647 for p = 5 which is better than -0.4556, design cost for p = 3; design cost remain same for p = 5 and p = 7. Further, all the benchmarks satisfy the user provided area-delay constraints to optimize the obfuscated design cost according to (4), thereby achieve design cost value less than 0. Design cost is a mathematical metric that indicates combined normalized value of silicon area and latency of a DSP design (refer (4)). Design cost may refer to implementation cost, however, since (4) computes a normalized value hence it is unit less.

The robustness of the proposed obfuscation is measured by the structural mismatch between the original design and the proposed obfuscated design. Power of Obfuscation for single stage obfuscation (p^{obf}) and multi-stage obfuscation (P^{obf}) is expressed using following equations indicating normalized value between 0 to 1 is:

$$p_i^{obf} = \frac{n_i}{n_i^T} \quad (6)$$

$$p^{obf} = \frac{\sum_{i=1}^{5} p_i^{obf}}{N(HLT)} \quad (7)$$

where, n_i is the number of modified nodes due to i^{th} HLT technique; n_i^T is the total number of nodes before applying i^{th} HLT technique; N(HLT) is the total number of HLT techniques applied on a particular application. Higher the value of P^{obf}, stronger is the security of the design. A node is considered as a modified one if any of the following cases is true:

- A parent node or a primary input of a node of an obfuscated CDFG is different than its original.
- The child of a node in an obfuscated CDFG is different than its original.
- The operation type (addition, multiplication etc) of a node in an obfuscated CDFG is changed.
- A node of an original CDFG is non-existent in the corresponding obfuscated CDFG.

Table II refers the p_i^{obf} due to each HLT technique as well as the total P^{obf} after applying multi-stage HLTs. For example, for ARF benchmark, p_i^{obf} is 0.32, 0.61, 0 for ROE, LT and THT respectively while applying each HLT separately, whereas, the total normalized P^{obf} is 0.55 for multi-stage based obfuscation.

C. Comparative Perspectives and Discussions

Comparison between a non-obfuscated design with its obfuscated form in terms of design area, latency and cost is shown in Table III. For example, the design area, design latency and design cost of non-obfuscated FIR benchmark are 214.99 sq. micro meter(µm2), 1661.33 pico-seconds (ps) and -0.31 respectively, whereas, the design area, design latency and design cost of obfuscated FIR benchmark are 214.99 sq. micro meter(µm2), 1477.84 pico-seconds (ps) and -0.35 respectively. *Note*: sometimes an obfuscated design achieves lower latency than a non-obfuscated design since the proposed approach performs series of HLT optimizations (such as ROE, THT, LT, unrolling etc.) that can result into reduction of operations in the graph. Thus in such specific scenario, the graph after scheduling may result into lower latency. Table IV shows the comparison between proposed multi-stage based obfuscation design with [11] in terms of design area, latency and cost. For example, the design area, design latency and design cost of [11] for FIR benchmark are 399.50 sq. micro meter(μm^2), 1315.67 pico-seconds (ps) and -0.25 respectively, whereas, the propose obfuscated approach explores better design solution for FIR benchmark (reported earlier). Finally, the proposed approach achieves an average 55% reduction of design cost for standard benchmarks [36] [37], compared to [11]. This is obtained due to PSO-DSE based optimization in the proposed approach for area-delay trade-off.

TABLE IV
Comparison of proposed obfuscated design with [8] in terms of area, latency and design cost

Benchmark	Design of [8]			Proposed Obfuscated Design		
	Area (μm^2)	Latency (ps)	Cost	Area (μm^2)	Latency (ps)	Cost
ARF	788.8	1511.5	-0.08	241.8	1900.8	-0.37
FIR	399.5	1315.7	-0.25	215.0	1477.8	-0.35
AUTO-CO	3196.1	1144.9	-0.32	736.4	2940.5	-0.46
DIF-EQN	3590.1	2326.4	-0.15	333.0	4498.8	-0.33
DHMC	4818.9	2891.1	-0.28	419.1	8581.6	-0.63
Ada-NC	2959.0	2826.5	-0.28	270.8	3120.01	-0.63
Ada-LMS	6075.2	4385.8	-0.30	292.3	6833.58	-0.70

TABLE V
Comparison of proposed obfuscated design with [8] in terms of p^{obf}

Benchmark	P^{obf} for Proposed	P^{obf} for [8]	P^{obf} improvement %
ARF	0.55	0.43	22.36
FIR	0.75	0.50	33.33
AUTO-CO	0.75	0.50	32.99
DIF-EQN	0.72	0.50	30.62
DHMC	0.69	0.50	27.27
Ada-NC	0.57	0.33	41.18
Ada-LMS	0.88	0.50	42.86

The comparative results of the proposed approach with [11], in terms of P^{obf} are reported in Table V. As evidence from the result, proposed multi-stage based obfuscation methodology provides stronger IP protection as for all the tested benchmarks proposed approach has higher P^{obf} value. Further, the proposed approach achieves an average 22% more robust than [11].

VII. Conclusion

This paper presents a novel low-cost structural obfuscation based reusable IP core protection mechanism for loop-based CDFG during algorithmic synthesis. This is the first attempt to perform structural obfuscation through multi-stage compiler-based high-level transformation techniques. The proposed approach achieves lower design cost as well as higher robustness than [11]. Besides yielding low design cost and high IP protection the proposed approach also provides low implementation time and easy adaptability to any CAD tool.

Our future research direction includes reusable IP core protection using both structural and functional obfuscation during algorithmic synthesis. We intend add more transformation to achieve more robust obfuscated design.

References

[1] E. Castillo, U. Meyer-Baese, A. Garcia, L. Parrilla, and A. Lloris, "IPP@HDL: Efficient Intellectual Property Protection Scheme for IP Cores," *IEEE Trans. Very Large Scale Integration Sys.*, vol. 15, no. 5, pp. 578–591, May 2007.

[2] H. Yang, N. Basutkar, P. Xue, K. Kim, and Y. H. Park, "Software defined DVT-T2 demodulator using scalable DSP processors," *IEEE Trans. on Consumer Electronics*, vol. 59, no. 2, pp. 428–434, May 2013.

[3] P. J. Lobo, E. Juarez, F. Pescador, G. Maturana, and M. C. Rodrguez, "A DVB-H receiver and gateway implementation on a FPGA- and DSP based platform," *IEEE Trans. on Consumer Electronics*, vol. 57, no. 2, pp. 372–378, May 2011.

[4] S. P. Mohanty, "GPU-CPU Multi-Core For Real-Time Signal Processing," in *Proc. IEEE ICCE*, Las Vegas, 2009, pp. 55–56.

[5] M. Kruni, I. Povaan, M. Popovi, and J. Kovaevi, "Data flow CAD tool for firniware development and power consumption estimation in multicore hearing aids," in *Proc. IEEE ICCE,* Las Vegas, 2016, pp. 575–576.
[6] M. Li, P. Zhang, C. Zhu, H. Jia, X. Xie, J. Cong, and W. Gao, "High efficiency VLSI implementation of an edge-directed video up-scaler using high level synthesis," in *Proc. IEEE ICCE,* Las Vegas 2015, pp. 92–95.
[7] A. Sengupta, "Intellectual Property Cores: Protection designs for CE products," *IEEE Consumer Electronics Mag.*, vol. 5, no. 1, pp. 83– 88, Jan 2016.
[8] The Economic Impacts of Counterfeiting and Piracy, last modified: 02.03.2017. [Online]. Available: http://www.inta.org/Communications/Documents/Forms/AllItems.aspx.
[9] J. Guajardo, S. S. Kumar, G. J. Schrijen, and P. Tuyls, "Brand and IP protection with physical unclonable functions," in *Proc. IEEE ISCS*, Seattle, May 2008, pp. 3186–3189.
[10] K. K. Parhi, "Verifying equivalence of digital signal processing circuits," in *ASILOMAR,* Pacific Grove, Nov 2012, pp. 99–103.
[11] Y. Lao and K. K. Parhi, "Obfuscating DSP Circuits via High-Level Transformations," *IEEE Trans. Very Large Scale Integration Sys.*, vol. 23, no. 5, pp. 819–830, May 2015.
[12] S. Walz and Y. Schrder, "A privacy-preserving system architecture for applications raising the energy efficiency," in *Proc. IEEE ICCE,* Berlin, 2016, pp. 62–66.
[13] A. Sengupta, "Hardware Security of CE Devices," *IEEE Consumer Electronics Mag.*, vol. 6, no. 1, pp. 130–133, Jan 2017.
[14] S. P. Mohanty, *Nanoelectronic Mixed-Signal System Design*. McGraw-Hill Education, 2015, no. 0071825711.
[15] J. Kim, E. s. Jung, Y. t. Lee, and W. Ryu, "Home appliance control framework based on smart TV set-top box," *IEEE Trans. on Consumer Electronics*, vol. 61, no. 3, pp. 279–285, Aug 2015.
[16] S. Thavalengal and P. Corcoran, "User Authentication on Smartphones: Focusing on iris biometrics," *IEEE Consumer Electronics Mag.*, vol. 5, no. 2, pp. 87–93, April 2016.
[17] S. Lu, X. Huang, L. Cui, Z. Zhao, and D. Li, "Design and implementation of an ASIC-based sensor device for WSN applications," *IEEE Trans. on Consumer Electronics*, vol. 55, no. 4, pp. 1959–1967, Nov 2009.
[18] A. Vijayakumar, V. C. Patil, D. E. Holcomb, C. Paar, and S. Kundu, "Physical Design Obfuscation of Hardware: A Comprehensive Investigation of Device and Logic-Level Techniques," *IEEE Trans. on Information Forensics and Security*, vol. 12, no. 1, pp. 64–77, Jan 2017.
[19] M. Brzozowski and V. N. Yarmolik, "Obfuscation as Intellectual Rights Protection in VHDL Language," *in CISIM,* Minneapolis*,* June 2007, pp. 337–340.
[20] C. Barria, D. Cordero, C. Cubillos, and R. Osses, "Obfuscation procedure based in dead code insertion into crypter," *in ICCCC,* Oradea May 2016, pp. 23–29.
[21] J. Rajendran, M. Sam, O. Sinanoglu, and R. Karri, "Security Analysis of Integrated Circuit Camouflaging*," in Proc. ACM SIGSAC*, Berlin, 2013, pp. 709–720.
[22] R. S. Chakraborty and S. Bhunia, "HARPOON: An Obfuscation-Based SoC Design Methodology for Hardware Protection," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 28, no. 10, pp. 1493–1502, Oct 2009.
[23] Y. M. Alkabani and F. Koushanfar, "Active Hardware Metering for Intellectual Property Protection and Security," *in Proc. USENIX Association, Berkeley,* 2007, pp. 20:1–20:16.
[24] J. Zhang, "A Practical Logic Obfuscation Technique for Hardware Security," *IEEE Tran. Very Large Scale Integration Sys.* vol. 24, no. 3, pp. 1193–1197, March 2016.
[25] X. Wang, X. Jia, Q. Zhou, Y. Cai, J. Yang, M. Gao, and G. Qu, "Secure and low-overhead circuit obfuscation technique with multiplexers," *in GLSVLSI*, May 2016, pp. 133–136.
[26] J. A. Roy, F. Koushanfar, and I. L. Markov, "EPIC: Ending Piracy of Integrated Circuits," *in DATE*, Munich, March 2008, pp. 1069–1074.
[27] J. Rajendran, Y. Pino, O. Sinanoglu, and R. Karri, "Security analysis of logic obfuscation," *in DAC, 2012*, San Francisco, June 2012, pp. 83–89.
[28] A. Baumgarten, A. Tyagi, and J. Zambreno, "Preventing IC Piracy Using Reconfigurable Logic Barriers," *IEEE Design Test of Computers*, vol. 27, no. 1, pp. 66–75, Jan 2010.
[29] A. Sengupta and D. Roy, "Protecting an intellectual property core during architectural synthesis using high-level transformation based obfuscation," *IET Electronics Letters*, Vol: 53, Issue: 13, pp. 849 – 851, June 2017
[30] D. Roy and A. Sengupta, "Low Overhead Symmetrical Protection of Reusable IP Core Using Robust Fingerprinting and Watermarking During High Level Synthesis," *Future Gener. Comput*. Syst., vol. 71, no. C, pp. 89–101, Jun. 2017.
[31] A. Sengupta and S. Bhadauria, "Exploring Low Cost Optimal Watermark for Reusable IP Cores During High Level Synthesis*," IEEE Access*, vol. 4, pp. 2198–2215, 2016.
[32] J. L. Wong, D. Kirovski, and M. Potkonjak, "Computational forensic techniques for intellectual property protection," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 23, no. 6, pp. 987–994, June 2004.
[33] F. Koushanfar, *Hardware Metering: A Survey.* New York, NY: Springer New York, 2012, pp. 103–122.
[34] A. Sengupta, S. Bhadauria, and S. P. Mohanty, "TL-HLS: Methodology for Low Cost Hardware Trojan Security Aware Scheduling With Optimal Loop Unrolling Factor During High Level Synthesis," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 36, no. 4, pp. 655–668, April 2017.
[35] NanGate 15 nm open cell library. [Online]. Available: http: //www.nangate.com/?pageid=2328, last accessed on June 2017
[36] DSP benchmark suite. [Online]. Available: http://www.ece.ucsb.edu/EXPRESS/benchmark/, last accessed on June 2017
[37] BDTI DSP Kernel Benchmarks. [Online]. Available: https://www.bdti.com/Services/Benchmarks/DKB, last accessed on June 2017

Prof. Anirban Sengupta (M'09) is an Asst. Professor (Associate Professor appointment approved) in Discipline of Computer Science and Engineering at Indian Institute of Technology (I.I.T) Indore, where he directs the research lab on 'CAD for CE Device Security & Reliability'. He is IEEE Distinguished Lecturer of IEEE Consumer Electronics Society. He is an author of nearly130 Publications (which includes 11 Patents). He is currently serving as Associate Editor of IEEE Transactions on Aerospace and Electronic Systems (TAES), Guest Editor of IEEE Transactions on VLSI Systems (TVLSI), Executive Editor of IEEE Consumer Electronics Magazine & Associate Editor of IEEE Access Journal. He is also Technical Program Chair of 36th ICCE 2018, Las Vegas.

Dipanjan Roy (S'16) is a research scholar in Computer Science and Engineering at Indian Institute of Technology (I.I.T) Indore. 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.

Prof. Saraju P. Mohanty (M'04-SM'08) is a Professor at the Department of Computer Science and Engineering (CSE), University of North Texas (UNT). Prof. Mohanty is an author of 220 peer reviewed publications and 3 books. 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), ACM Journal on Emerging Technologies in Computing Systems (JETC). He is currently the Editor-in-Chief (EiC) of the IEEE Consumer Electronics Magazine.

Prof. Peter Corcoran (F' 10) is a Fellow of IEEE, the Founding Editor of IEEE Consumer Electronics Magazine and holds a Personal Chair in Electronic Engineering at the College of Engineering & Informatics at NUI Galway. His current research interests include biometrics, deep learning, embedded computer vision, and consumer electronics. He is co-author on 300+ technical publications and co-inventor on 300+ granted US patents.