Low-Cost Obfuscated JPEG CODEC IP Core for Secure CE Hardware

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

***Abstract*—A novel approach for obfuscated JPEG compression/decompression (CODEC) IP core design methodology, suitable for use in re-usable IP core designs, is presented. This incorporates structural obfuscation for architecture or structure hiding from an adversary in order to maximize the complexity against reverse engineering (RE). Additionally, the proposed methodology performs the entire compression and decompression through a single dedicated hardware IP core. To obtain a lightweight (low cost version) of the proposed obfuscated JPEG CODEC IP, particle swarm optimization (PSO) driven design space exploration (DSE) is employed. Results of the proposed low cost, obfuscated JPEG CODEC IP core design when compared to un-protected (un-obfuscated) design yielded enhancement in strength of obfuscation of 76%, as well as reduction of 5% compared to un-optimized design.**

***Index Terms*—JPEG CODEC, structural obfuscation, grayscale image, IP protection**

I. Introduction

ENERGY efficiency and security are important constraints that a design engineer needs to balance carefully in an effective Consumer Electronics (CE) design [1]. These constraints are equally valid for networked or stand-alone designs and motivate an exploration of new design paradigms [3], [4] to optimize the security of an electronics design at minimal additional energy cost.

In modern designs there is a trend for re-usable IP cores. The use of such cores optimizes design productivity and minimizes design time. But standard IP core design process does not have ability to produce architectures that look functionally unobvious and thus makes reverse engineering harder. RE attacks could be employed by rogue elements in the design flow to implant malicious hardware logic in CE system-on-chip or to produce counterfeit designs of CE hardware.

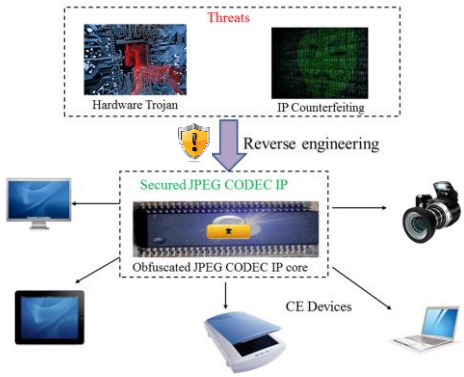

Fig. 1. Thematic representation of obfuscated JPEG CODEC IP

IP protection ensures protection of the hardware ownership or intellectual property rights (IPR) [6].

Digital signal processing (DSP) hardware processes some form of multimedia signal including image, audio, and video as a part of a CE system [7]. In the current paper, the focus is on image digital signal processing. However, in principle, the proposed design paradigm can be undertaken for audio and video. Image compression and decompression is a major functionality in most of these modern CE devices. In CE devices, a dedicated IP core can play a vital role in performing this process. Since, an IP vendor/designer invests substantial cost, workforce, research, and verification on its design, thus, it must be protected against cloning and/or infringement (refer Fig. 1). Insertion of malicious Trojan logic into the IP core through RE attack is another major problem that is faced by an IP vendor [6]. Successful RE attack enables an adversary (in the foundry) to identify the functionality of the design to counterfeit the netlist of the design and make several copies of it without the knowledge of the IP vendor/owner. Further, if an adversary is able to identify the functionality of the design from its functional structure, he/she can make malicious modification (i.e. insert Trojan) into the design and create malfunctioning. Both these threats can be avoided if reverse engineering process is made harder for an adversary. This is possible by structurally obfuscating the design such that the functionality of it becomes un-obvious to an adversary i.e. architecture designed to be hidden to an adversary. This strongly hinders the RE process, thus thwarting the threats.

Rest of the paper is organized as follows: The novel contributions of this paper are summarized in Section II. Section III explains related research works. The proposed methodology and its related theory for JPEG CODEC are

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

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

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

S. P. Mohanty is with Department of Computer Science, University of North Texas, Denton, Texas 1155 (e-mail: saraju.mohanty@unt.edu).

P. Corcoran is with the College of Engineering & Informatics, National University of Ireland Galway H91 TK33 (e-mail: peter.corcoran@nuigalway.ie).

explained in Sections IV and V respectively. Section VI provides implementation process of the resulting IP core design. Section VII presents the experimental results of the proposed approach. Section VIII presents conclusions and future research.

II. Novel Contributions of the Paper

Reusable IP cores play a vital role in the design of modern CE devices by maximizing design productivity and minimizing design time. In this paper we use the example of JPEG CODEC used in digital camera to explain how architecture can be transformed to enhance complexity against RE. Standard JPEG CODEC design process does not aim to obscure the functionality from an adversary by transforming the architecture. This does not make the reverse engineering process harder for an adversary. JPEG CODEC is considered secure, if its architecture is not obvious i.e. its functionality is not easily discoverable by inspecting its structure. JPEG CODEC has been selected as an example because it is a very vital core widely used in several CE applications. Keeping in view the recent surge of IP counterfeits and hardware Trojan, it is important to secure JPEG CODEC IP core against RE threats as failing to do so may lead to serious functional failures (due to Trojan) and/or significant loss to CE industry (due to IP counterfeit). There is no work in the literature that addresses this concern and it is high time that this concern be addressed. The contributions are:

- Proposes a novel obfuscation methodology for dedicated JPEG CODEC IP core that aims to obscure the functionality from an adversary by transforming the architecture. This would make the reverse engineering process harder for an adversary (threat model discussed in Section IV). Enhancement in strength of obfuscation [16] of 76% is obtained through proposed method.
- Proposes a novel design of obfuscated JPEG CODEC IP core hardware as a proof of concept of this methodology.
- Optimization of the cost of the obfuscated JPEG CODEC IP core design using particle swarm based design space exploration. Reduction in design cost of 5% on the current IP core design is achieved.

III. Related Prior Research

IP protection techniques can be broadly classified into two categories: (a) passive method and (b) active method. Passive IP protection approaches such as symmetrical IP core protection mechanisms [9], computational forensic engineering (CFE) [10], IP metering, hardware metering and IP vendor protection using signature [13] provide passive mode of protection which is only capable of tracing the clone copies of IP core but unable to prevent it from being stolen.

Active methods hide the functionality and implementation of an IP core as it passes through the different potentially untrustworthy phases of the design flow. Obfuscation is the process of transforming an original application or design into a functionally equivalent form to make the reverse engineering process significantly more complex [2, 5, 8, 12]. This can be done in two ways (a) key based, known as logic encryption [11]; (b) non-key-based, known as structural obfuscation. In logic encryption, the functionality is known to an attacker but the right combination of key is unknown; on the other hand structural obfuscation tries to hide the correct functionalities of the design [16], therefore protection through key is not required. Moreover, in [2, 5, 8] obfuscation is performed for sequential/combinational circuits at gate or layout level but not for DSP core at architectural level. In this paper, we have proposed low-cost high-level transformation based structurally obfuscated IP core design for JPEG CODEC.

A new recursive algorithm and two types of circuit architectures for the computation of 2D DCT is presented in [14]. The proposed algorithm is capable of computing the 2D DCT using the 1D DCT recursively. A theoretical analysis on the variation of local variance due to JPEG compression is presented in [14]. It is suggested that the local variance under JPEG compression can be used in image processing and analysis, such as image enhancement, image quality assessment, and image filtering. A wavelet transformation based grayscale image CODEC IP core is proposed in [15]. The architecture achieves efficient hardware utilization and lower hardware cost. Further, authors in [16] have proposed protection of DSP cores using compiler based transformations.

The current state of the art either do not offer resiliency against RE [14] [15] or may use key based logic locking [11] [12] for resiliency against RE. However, logic locking may incur design cost overhead in terms of area and latency. Secondly, deciding appropriate location for key insertion is not trivial and may require separate algorithm which may increase the complexity. Thirdly, key based logic locking is vulnerable to Boolean Satisfiability (SAT) attack, the protection against which may require complex block such as AES, which may again incur design overhead. However, enhancing complexity against RE attack through structural obfuscation does not incur the above limitations. Thus structural obfuscation although being a passive mode of protection offers substantial resiliency against RE at zero design overhead compared to the key based active mode of protection. Our example JPEG CODEC IP core used in this paper not only makes the architecture unobvious through high-level transformation based structural obfuscation but also explores a low-cost design solution through PSO.

IV. Proposed Methodology

This section discusses the proposed methodology for obfuscation of DSP core for CE systems as well as the low cost design space exploration system using PSO. We would also present the threat model and its solution.

A. Proposed Obfuscation

The proposed obfuscation methodology is capable to yield an obfuscated structure/architecture whose functionality is unobvious to an adversary. Through the proposed method an unsecured IP design, containing micro IP as well as the overall macro IP, is structurally obfuscated through Tree Height Transformation (THT). THT is a compiler driven optimization

that is useful for obfuscating an original DSP core design by transforming the height of the graph. It divides the critical path dependency into temporary sub-computations and evaluates in parallel, thereby generating functionally equivalent yet structurally dissimilar graph elements. Following are the generic detailed steps of the proposed obfuscation:

1. Formulate the transfer function or mathematical representation of a DSP core.
2. Perform expansion of the formulated transfer function or mathematical representation of a DSP core.
3. Derive the data flow graph (DFG)/control data flow graph (CDFG) corresponding to the DSP core.
4. Identify micro IPs within the macro IP corresponding to the above DFG/CDFG obtained.
5. Apply THT based structural obfuscation on each identified micro IPs as well as the macro IP of the corresponding DFG/CDFG.
6. Feed the THT-driven obfuscated DFG/CDFG of the DSP core into the High-Level Synthesis (HLS) engine.
7. Finally, a structurally obfuscated DSP core is obtained.

B. Proposed Design Space Exploration System for Low-cost Obfuscated DSP Core

The proposed approach integrates the obfuscation methodology with a Particle swarm optimization design space exploration (DSE) optimization framework.

PSO process: PSO is a population-based heuristic optimization that searches for an optimal solution iteratively. Each solution of the search-space is encoded as a particle and the fitness of each particle is evaluated based on the fitness function. 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. After finding a better gbest or lbest the i^{th} particle updates its velocity and position thus move towards the best solutions. More details are available in [6].

Benefits of PSO: a) Ability to escape local minima and converge on global optima in most cases, b) ability to introduce stochasticity into the exploration process, c) preserves exploration-exploitation balance during searching low-cost solution. PSO-DSE is performed on obfuscated DSP core DFG/CDFG, in order to achieve a low-cost IP core version. In other words, for obtaining low cost obfuscated IP hardware, obfuscated DSP core DFG/CDFG, module library, PSO control parameters and PSO terminating condition are fed as the inputs to the PSO-DSE block.

Initialization of particle: Each particle (P_i) is a solution of design resources that can be expressed as:

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

where, N(R_D) is the number of resource type R_D.

The particles are initialized based on uniform distribution over the search space and can be represented as follows [6]:

$$P_n = \left\{ \frac{(\min(R_1) + \max(R_1))}{2} \pm \alpha, .. \frac{(\min(R_D) + \max(R_D))}{2} \pm \alpha \right\} \quad (2)$$

where, 'α' is a random integer between min value and max value of a particular resource type.

Movement of particle using velocity: In the PSO-DSE process [6], each dimension (d) of a particle velocity (V_{di}) is updated based on the following equation:

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

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} and R_{dgb} are the local best position and global best of i^{th} particle in d^{th} dimension.

C. *Threat Model and Problem Formulation*

1) Threat Model: The proposed work by performing obfuscation based architectural change, converts the design architecture into a form whose functionality is not obvious to an adversary. Thus makes it harder to reverse engineer as it is harder for an adversary to discover the actual functionality.

2) Problem Formulation and Fitness Model: Design a low-cost, obfuscated, IP core. The fitness of each particle based on area-delay trade-off can be evaluated through the following:

$$C_f(P_i) = \Phi_1 \frac{A_T^{DSP}}{A_{\max}^{DSP}} + \Phi_2 \frac{D_T^{DSP}}{D_{\max}^{DSP}} \quad (4)$$

where, $C_f(P_i)$ is the cost of the particle P_i; A_T^{DSP} and D_T^{DSP} indicate the total area and total execution delay of the obfuscated DSP core IP design respectively; A_{max}^{DSP} and D_{max}^{DSP} indicate the maximum area and execution delay of the aforementioned design respectively; $Ø_1$ and $Ø_2$ are user defined weight parameter for area and delay respectively, where the values lie between 0 to 1 (in proposed approach, equal weightage is assigned to both $Ø_1$ and $Ø_2$).

V. PROPOSED METHODOLOGY ON JPEG CODEC IP CORE

In this section we explain the application of the proposed methodology on JPEG CODEC IP core. Firstly, the overview of JPEG process is provided. Next we demonstrate the proposed method on the design of low-cost obfuscated JPEG compression IP core, followed by demonstration on the design of low-cost obfuscated JPEG decompression IP core

A. Overview of JPEG Process

In JPEG compression/decompression, pre-processed image is taken as input. At first an NxN or NxM gray scale image is converted into an NxN or NxM matrix. Each integer value of the matrix represents the pixel intensity of a particular pair of co-ordinate (x,y) of the image. For 8-bit depth gray scale images the range is 0 to 255, where 0 indicates pure black and 255 indicates pure white. Next, the input matrix is then sub-divided into multiple non-overlapping 8x8 blocks of pixels.

$$T = \begin{bmatrix} C_4 & C_4 & C_4 & C_4 & C_4 & C_4 & C_4 & C_4 \\ C_1 & C_3 & C_5 & C_7 & -C_7 & -C_5 & -C_3 & -C_1 \\ C_2 & C_6 & -C_6 & -C_2 & -C_2 & -C_6 & -C_6 & C_2 \\ C_3 & -C_7 & -C_1 & -C_5 & C_5 & C_1 & C_7 & -C_3 \\ C_4 & -C_4 & -C_4 & C_4 & C_4 & -C_4 & -C_4 & C_4 \\ C_5 & -C_1 & C_7 & C_3 & -C_3 & -C_7 & C_1 & -C_5 \\ C_6 & -C_2 & C_2 & -C_6 & -C_6 & C_2 & -C_2 & C_6 \\ C_7 & -C_5 & C_3 & -C_1 & C_1 & -C_3 & C_5 & -C_7 \end{bmatrix}$$

Fig. 2. 2D-DCT coefficient matrix

The generic 2D-DCT coefficient matrix 'T' can be presented as shown in Fig. 2.As 2D-DCT can process an 8x8 block at a time, the input image is sub-divided into multiple non-overlapping 8x8 blocks of pixels. A generic pixel intensity of a 8x8 input image matrix 'M' can be presented in the form of m_{ij} as shown in Fig. 3 where, 'i' and 'j' represent the row and column number respectively of the pixel intensity of the 8x8 input image block.2D-DCT coefficients and standard quantization matrix are also fed as inputs. Finally, as DCT can handle pixel value within the range of -128 to 127, each block is leveled off by subtracting 128 from each pixel intensity. The quantized pixel intensity data can be represented as 8x8 2-dimensional matrix forms. Zigzag scanning is performed on this output data to convert it into 1-dimension array and then run-length encoding is applied to generate the bit stream data of the compressed image for finally storing it in a storage device. To decompress the image pixel intensities from the stored data, the stored bit stream representing compressed pixel data is first decoded through run-length decoding and then through inverse zigzag scanning, its equivalent 2D image pixel intensity matrix is reconstructed. To perform JPEG image decompression, inverse quantization is applied on the compressed image pixel block by multiplying each element of block with the corresponding element of the quantization matrix (Q) to obtain de-quantized image pixel intensities. Next inverse DCT is applied on the de-quantized compressed image block for decompression. Inverse DCT of compressed image block is achieved by applying 2D-DCT coefficient matrix on the de-quantized image block. 2D-DCT coefficients and standard quantization matrix are also fed as inputs.

B. Overview of Proposed Obfuscation Methodology in Compression

Using the proposed obfuscation steps in section IV.A and DSE engine process in section IV.B, the proposed low-cost, obfuscated JPEG compression IP core is designed. The design process includes multiple steps (as shown in Fig. 6). Initially an unprotected (unsecured) JPEG compression application in the form of a Data Flow Graph (DFG) is accepted as an input. Next, resiliency in the form of structural obfuscation is provided to the unsecured DFG to obtain an obfuscated version. This obfuscated DFG is process through an optimization framework to obtain a low-cost hardware configuration (detail explained in Section IV. B earlier).Thus this low-cost hardware configuration is used to design an obfuscated dedicated hardware for JPEG compression IP core (IP core 1). The proposed obfuscated JPEG compression IP core uses levelized pixel intensity as input to generate the compressed image pixel intensity as output. Finally, to generate the compressed pixel intensities through proposed IP core, 2D-DCT coefficients and standard quantization matrix are also fed as inputs.

$$M=\begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} & m_{15} & m_{16} & m_{17} & m_{18} \\ m_{21} & m_{22} & m_{23} & m_{24} & m_{25} & m_{26} & m_{27} & m_{28} \\ m_{31} & m_{32} & m_{33} & m_{34} & m_{35} & m_{36} & m_{37} & m_{38} \\ m_{41} & m_{42} & m_{43} & m_{44} & m_{45} & m_{46} & m_{47} & m_{48} \\ m_{51} & m_{52} & m_{53} & m_{54} & m_{55} & m_{56} & m_{57} & m_{58} \\ m_{61} & m_{62} & m_{63} & m_{64} & m_{65} & m_{66} & m_{67} & m_{68} \\ m_{71} & m_{72} & m_{73} & m_{74} & m_{75} & m_{76} & m_{77} & m_{78} \\ m_{81} & m_{82} & m_{83} & m_{84} & m_{85} & m_{86} & m_{87} & m_{88} \end{bmatrix}$$

Fig. 3. Generic 8x8 matrix of input image

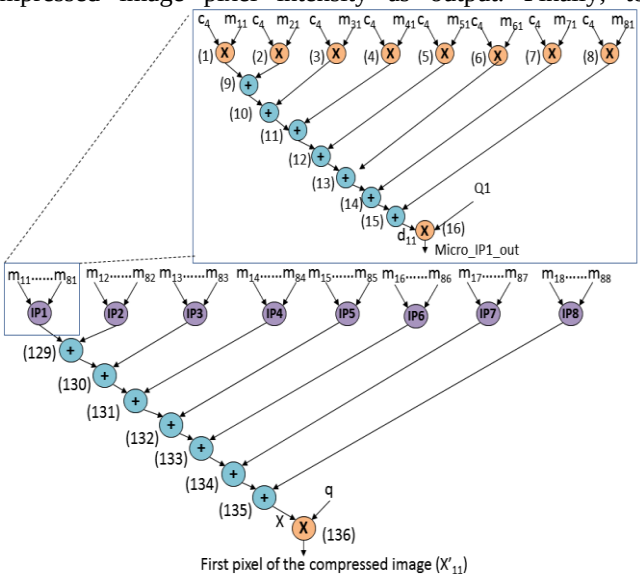

Fig. 4. Non-obfuscated data flow graph of JPEG image compression for calculating first pixel of the compressed image (X'_{11})

C. Generating Non-obfuscated JPEG CODEC IP Core in terms of Data Flow Graph

To perform DCT on an input image block, 2D-DCT coefficient matrix (T) is hit on the input block (M) through on the following expression:

$$X = D * T^{trans} \quad (5)$$

Where, 'D' is calculated through (6)

$$D = T * M \quad (6)$$

Elements of 'D' matrix (d_{11},d_{12},..d_{88}) indicates column wise transformed elements of input block. Further, elements of 'X' matrix (X_{11}, X_{12},...X_{88}) indicates both row and column wise transformed elements of input block. Thus, 'X' is the corresponding discrete cosine transformed block of input image block M; T^{trans} is the transpose of 2D-DCT coefficient matrix 'T'.

To convert the matrix relationship into a hardware function for dedicated IP core design, the pixels (m_{ij}) of the input image block (M) is transformed into compressed image pixel (X_{ij}) (using (5)). For example, X_{11} which is the first pixel of the compressed image is modeled as follows:

$$X_{11} = (c_4*d_{11} + c_4*d_{12} + c_4*d_{13} + c_4*d_{14} + c_4*d_{15} + c_4*d_{16} + c_4*d_{17} + c_4*d_{18}) \quad (7)$$

Where, d_{11}, d_{12},…d_{18} is calculated as follows:

$$d_{11} = c_4*m_{11} + c_4*m_{21} + c_4*m_{31} + c_4*m_{41} + c_4*m_{51} + c_4*m_{61} + c_4*m_{71} + c_4*m_{81} \quad (8)$$

$$d_{12} = c_4*m_{12} + c_4*m_{22} + c_4*m_{32} + c_4*m_{42} + c_4*m_{52} + c_4*m_{62} + c_4*m_{72} + c_4*m_{82} \quad (9)$$

Similarly,

$$d_{18} = c_4*m_{18} + c_4*m_{28} + c_4*m_{38} + c_4*m_{48} + c_4*m_{58} + c_4*m_{68} + c_4*m_{78} + c_4*m_{88} \quad (10)$$

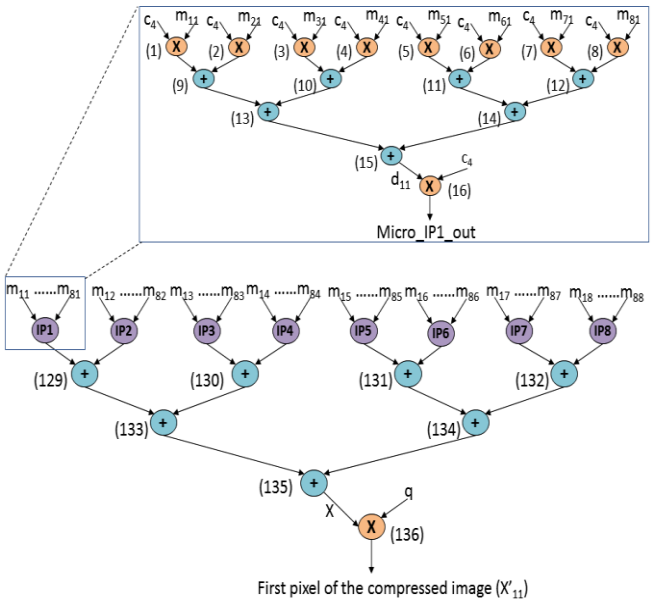

Fig. 5. Obfuscated data flow graph of JPEG image compression for calculating first pixel of the compressed image (X'_{11})

Similarly, other pixels of the block (M) are transformed where the input pixels remain same but the 2D-DCT coefficients become different. Thus the structure of the equation remain same, however only the inputs will be different while computing different transformed image pixel intensities. An equivalent DFG corresponding to (7) denoting an unsecured/unprotected (un-obfuscated) JPEG image compression is shown in Fig. 4, Each macro IP is designed using eight structurally equivalent micro IPs (name IP1-IP8); and each micro IP executes d_{ij} using (8)-(10). The un-obfuscated macro IP with one zoom in micro IP is also shown in Fig. 4. Each micro IP operation, addition operation and multiplication operation are indicated by purple, blue and orange node respectively. As shown in Fig. 4 the output of operation 135 generates the pixel intensity of the transformed image (X).

Post DCT transformation, pixel intensities of 8x8 image block (X) obtained are then compressed through quantization. To achieve different quality level (ranging from 1 to 100), different quantization matrix is used. Quality level 100 indicates less compression but higher quality image and quality level 0 indicates higher compression but less quality image. This is achieved through operation 136 performing the quantization on the transformed image pixel intensities (X) based on the corresponding element 'q' of the quantization matrix (Q). This produces the final output as quantized/compressed image pixel intensities (X'). Next, this un-obfuscated macro IP will be secured through high-level transformation based structural obfuscation in proposed approach which has been explained in next sub-section.

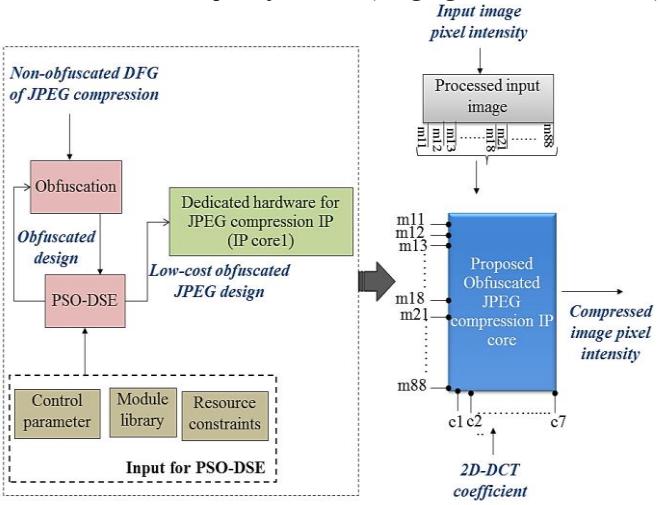

Fig. 6. Obfuscated JPEG compression IP core

D. Generating Obfuscated JPEG Compression IP Core

After performing THT-based structural obfuscation on JPEG DFG, the number of nodes in the original structure that gets affected is 12 (out of 16 nodes shown in Fig. 4) for micro IP DFG. The nodes that have got affected are node number 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14 and 15, as for the aforesaid nodes either the input/ output connectivity to the resources changes or the number of resources per control step changes, resulting into different datapath architecture and controller logic. Thus overall number of nodes affected in the macro IP is 102 nodes. This is because for every micro IP 12 nodes are affected and there are total 8 micro IPs in the entire obfuscated graph. The additional 6 nodes that are affected are node number 130, 131, 132, 133, 134 and 135, as for the aforesaid nodes either the input/ output connectivity to the resources changes or the number of resources per control step changes, resulting into a different datapath architecture and controller logic. The obfuscated macro IP with one zoom in obfuscated micro IP is shown in Fig. 5 where, each micro IP, adder and multiplier is represented through purple, blue and orange node respectively. Next phase of proposed design methodology is to generate a low-cost design for obfuscated macro IP core for JPEG compression via PSO-DSE process using HLS framework [6].

E. Overview of Proposed Design Obfuscation Methodology in Decompression

The proposed low-cost, obfuscated JPEG decompression IP core is designed through multiple steps (as shown in Fig. 7). Similar to the proposed obfuscated JPEG compression IP core design process, the proposed low cost obfuscated JPEG decompression IP core design process also accepts an unprotected (un-obfuscated) JPEG decompression DFG as input. Similarly, as performed in compression of JPEG IP core, resiliency in the form of structural obfuscation is provided to the unsecured DFG to obtain an obfuscated decompression DFG. This obfuscated DFG is process through an optimization framework to obtain a low-cost hardware configuration. Thus this low-cost hardware configuration is used to design an obfuscated dedicated hardware for JPEG decompression IP core (IP core 2) is obtained. Finally, for generating the decompressed pixel intensities through proposed IP core 2, 2D-DCT coefficients and standard quantization matrix are fed as inputs. The detailed explanation of each step for proposed obfuscated JPEG decompression IP core hardware is explained in next sub-sections.

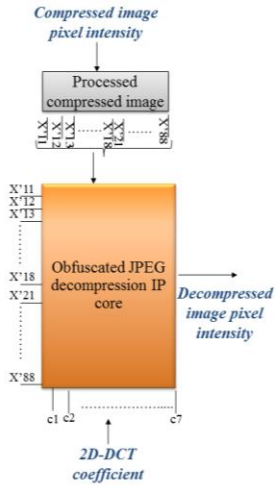

Fig. 7. Obfuscated JPEG decompression IP core

F. Generating Obfuscated JPEG CODEC IP Core for Decompression in terms of Data Flow Graph

Similar to the JPEG image compression process, in case of performing decompression the compressed image is also segmented into multiple 8x8 blocks. Inverse DCT of compressed image block is achieved by applying 2D-DCT coefficient matrix on the de-quantized image block (X'') based on the following expression:

$$O = E * T \quad (11)$$

Where, 'E' is calculated through (12)

$$E = T^{trans} * X'' \quad (12)$$

Elements of 'E' matrix (e_{11}, e_{12}, ...e_{88}) indicates column wise transformed elements of de-quantized image block. Further, elements of 'O' matrix (o_{11}, o_{12},...o_{88}) indicates both row and column wise transformed elements of de-quantized image block. Thus 'O' is the corresponding inverse DCT block of de-quantized image block X''. To convert the matrix relationship into a hardware function for dedicated IP core design, the pixels (X''_{ij}) of the de-quantized image block (X'') is transformed into decompressed image pixel (o_{ij}) (using (11)) as shown in (12). For example for o_{11} which is the first pixel of the decompressed image is presented as follows:

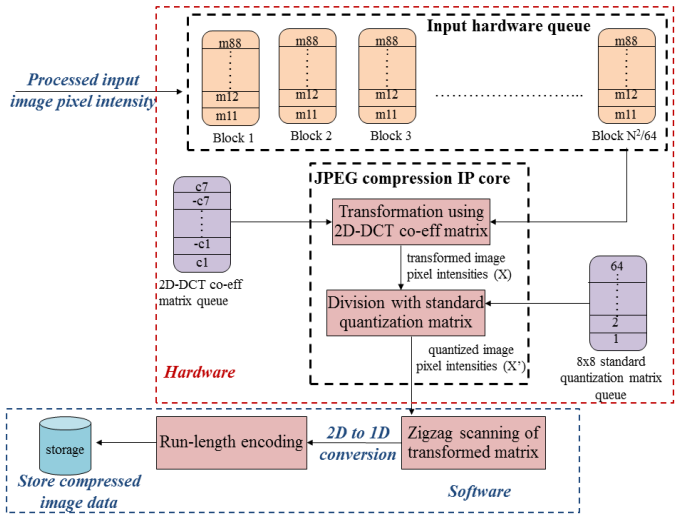

Fig. 8. Proposed hardware and software design flow using JPEG compression IP

$$o_{11} = (c_4*e_{11} + c_1*e_{12} + c_2*e_{13} + c_3*e_{14} + c_4*e_{15} + c_5*e_{16} + c_6*e_{17} + c_7*e_{18}) \quad (13)$$

Where, e_{11}, e_{12},…e_{18} is calculated as follows:

$$e_{11} = (c_4*X''_{11} + c_1*X''_{21} + c_2*X''_{31} + c_3*X''_{41} + c_4*X''_{51} + c_5*X''_{61} + c_6*X''_{71} + c_7*X''_{81}) \quad (14)$$

$$e_{12} = (c_4*X''_{12} + c_1*X''_{22} + c_2*X''_{32} + c_3*X''_{42} + c_4*X''_{52} + c_5*X''_{62} + c_6*X''_{72} + c_7*X''_{82}) \quad (15)$$

Similarly,

$$e_{18} = (c_4*X''_{18} + c_1*X''_{28} + c_2*X''_{38} + c_3*X''_{48} + c_4*X''_{58} + c_5*X''_{68} + c_6*X''_{78} + c_7*X''_{88}) \quad (16)$$

Similarly, other pixels of the de-quantized image block (X'') are transformed through inverse DCT matrix where the input pixels remain the same however the 2D-DCT coefficients become different. It is to be noted that the structure and pattern of forward DCT (5) and inverse DCT (11) are same, but only the inputs are different during computation of decompressed image pixel intensity.

An equivalent DFG corresponding to (13) denoting an unsecured/unprotected (un-obfuscated) JPEG image decompression can be obtained. As performed for JPEG compression DFG, obfuscation through tree height transformation can be performed on the un-obfuscated JPEG decompression DFG. Each micro IPs, as well as the complete macro IP, is structurally obfuscated through Tree Height Transformation. Similar to JPEG image compression DFG, an obfuscated JPEG decompression macro IP is designed using eight structurally equivalent micro IPs (name IP1-IP8), where each micro IP executes a part of (13). *Note: for the sake of brevity, JPEG decompression DFG has not been included.* Finally, de-levelization is performed on obfuscated DFG output pixel intensity of decompressed image by adding 128.

Next phase of proposed design methodology is to generate a low-cost design for obfuscated macro IP core of JPEG decompression through PSO-DSE process using HLS framework [6]. More details about PSO-DSE are available in the next Section.

VI. Proposed Implementation of JPEG CODEC IP Core

This section explains the proposed implementation.

A. Proposed Design of Obfuscated JPEG Compression IP

Fig. 8 shows the complete design setup for proposed JPEG compression process using proposed IP core. The IP core used is capable to accept an 8x8 block of a processed gray-scale image pixel intensity stored in a hardware queue along with 2D-DCT coefficients and standard quantization matrix. It performs transformation using 2D-DCT coefficient, quantize and round off the transformed result and finally generates the pixel intensities of the compressed image data (X').

As explained in Fig. 6 earlier, design resource configuration of 3 adders and 3 multipliers obtained through PSO-DSE process is used for designing proposed low-cost obfuscated JPEG IP core. The obfuscated DFG shown in Fig. 5 is scheduled in different control step (c.s.) based on As Soon As Possible (ASAP) algorithm using this resource configuration. Binding is performed on the scheduled DFG to map 136 operations of JPEG compression DFG to 3 adders and 3 multipliers. The c.s of each operation and their corresponding mapped hardware is

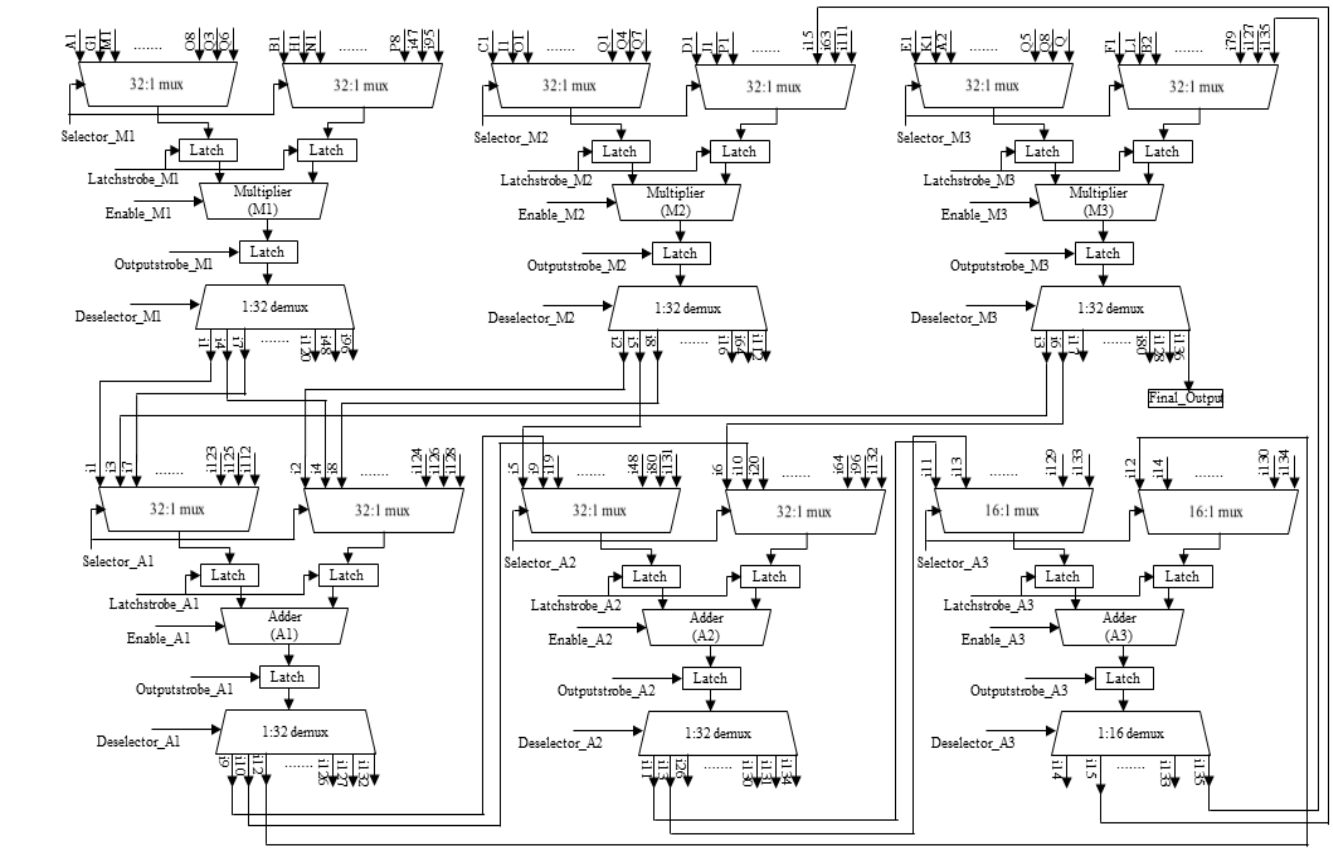

Fig. 9. DSP representation of the proposed obfuscated JPEG compression IP core

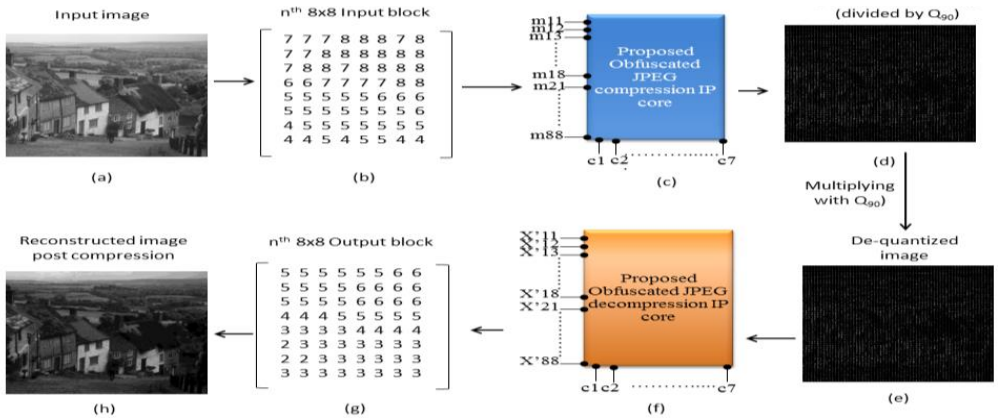

Fig. 10. Depiction of end to end process of JPEG image compression

shown in Table I. After performing scheduling and binding the development of JPEG compression IP core system is obtained through HLS framework [6]. The system block diagram comprises into two parts, data-path and controller. Three 16-bit adders (Adder_A1, Adder_A2 and Adder_A3) and three 16-bit multipliers (Multiplier_M1, Multiplier_M2 and Multiplier_M3) along with ten 32:1 and two 16:1 multiplexer, five 1:32 and one 1:16 demultiplexer, 40 registers and 18 latches are used to design the complete data-path of the proposed IP core. All the resources are of 16-bits because both inputs and outputs are in 16 bits IEEE 754 half precision floating-point format. The datapath block diagram is shown Fig. 9.

B. Proposed Design of Obfuscated JPEG Decompression IP

Low-cost design resource configuration obtained through PSO-DSE process is used for designing proposed low-cost obfuscated JPEG decompression IP core (Fig.7). In the proposed approach, a low cost design resource configuration comprising of 3 adders and 3 multipliers is obtained through PSO-DSE to design a low-cost obfuscated JPEG decompression IP core. The obfuscated DFG (similar to Fig. 5) is scheduled in different c.s. based on ASAP algorithm based on the aforementioned resource configuration. Binding is performed on the scheduled DFG to map 136 operations of JPEG decompression DFG to 3 adders and 3 multipliers. After performing scheduling and binding, the development of data-path and controller of JPEG decompression IP core is performed. Along with three 16-bit adders and three 16-bit multipliers, ten 32:1 and two 16:1 multiplexer, five 1:32 and one 1:16 demultiplexer, 40 registers and 18 latches are used to design the complete data-path of the proposed JPEG decompression IP core. The designed decompression IP core is capable of accepting an 8x8 block of compressed image pixel intensity along with 2D-DCT coefficients and standard quantization matrix. It is capable of performing inverse

TABLE I
Scheduling and binding of operations for obfuscated JPEG IP core

C.S.	Opn. assigned to M1	Opn. assigned to M2	Opn. assigned to M3	Opn. assigned to A1	Opn. assigned to A2	Opn. assigned to A3
1	1	2	3	-	-	-
2	4	5	6	9	-	-
3	7	8	17	10	11	-
4	18	19	20	12	13	-
5	21	22	23	25	26	14
6	24	33	34	27	29	15
7	35	36	37	28	41	-
8	38	39	40	42	30	-
9	49	50	51	43	44	45
10	52	53	54	57	46	31
11	55	56	65	58	59	47
12	66	67	68	60	61	-
13	69	70	71	73	74	62
14	72	81	82	75	77	63
15	83	84	85	76	89	-
16	86	87	88	90	78	-
17	97	98	99	91	92	93
18	100	101	102	105	94	79
19	103	104	113	106	107	95
20	114	115	116	108	109	-
21	117	118	119	121	122	110
22	120	16	32	123	125	111
23	48	64	80	124	129	-
24	96	112	-	126	130	-
25	-	-	-	127	131	133
26	-	-	128	-	-	-
27	-	-	-	132	-	-
28	-	-	-	-	134	-
29	-	-	-	-	-	135
30	-	-	136	-	-	-

quantization followed by inverse transformation using 2D-DCT coefficient, to finally generate a levelized decompressed image pixel intensity value as output.

C. Demonstration of End to End Process

For demonstration, a two-dimensional 512x512 gray scale image shown in Fig. 10(a) is taken as the input in the form of a matrix. This input matrix is then sub-divided into multiple non-overlapping 8x8 blocks. Fig. 10(b) represents the n^{th} 8x8 block of the input image which is compressed through our proposed compression IP core (shown in Fig. 10(c)). The output of proposed JPEG compression IP core produces DCT quantized image pixel intensity values. The DCT quantization image of the corresponding input image is shown in Fig. 10(d), where quantization is performed based on quantization matrix Q_{90}. To reconstruct de-quantized image, the DCT quantized image (after multiplying with Q_{90} quantization matrix) as shown in Fig. 10(d) is again sub-divided into multiple non-overlapping 8x8 blocks (shown in Fig. 10(e)). Each block of de-quantized image is then decompressed through our proposed JPEG decompression IP core, shown in Fig. 10(f) to generate 8x8 output blocks. Fig. 10(g) represents the n^{th} 8x8 block of the output image. After combing all 8x8 output blocks of decompressed image, the output reconstructed image of size 512x512 is generated, as shown in Fig. 10(h).

VII. Experimental Results

Standard 512x512 gray scale test images [17] and NASA images [18] are used as image dataset to generate compressed/decompressed images through the proposed IP cores. Both the proposed IP cores are implemented in standard synthesis tool. The FPGA device utilization summary of proposed IP cores is reported in Table II. The comparison of proposed low-cost obfuscated JPEG CODEC IP core with an obfuscated JPEG CODEC IP core (without optimization) in terms of design area, latency and cost is shown in Table III (note: design cost is calculated using eqn 4 which is a weighted function of normalized area and normalized delay for obfuscated design. Thus it does not have any unit). Non-optimized obfuscated JPEG CODEC IP core indicates obfuscated JPEG CODEC design generated without involving PSO-DSE (i.e. without optimization). Thus it incurs more design overhead due to lack of optimization framework during designing. The NanGate library is used to evaluate both the area and the latency of IP core design [19].

It can be observed from Table III that the proposed JPEG CODEC IP core achieves reduction of greater than 5% in design cost compared to the non-optimized obfuscated JPEG. This is due to the tree height transformation applied on the JPEG CODEC data flow graph followed by integration of particle swarm optimization DSE framework. Tree height transformation in proposed methodology drastically reduces the length of the critical path of the DFG thus minimizing schedule delay. This impacts reduction of cost. Subsequently this transformed DFG is fed into PSO-DSE which iteratively prunes the design space and explores an optimal low cost design resource. As these are novel solutions to reduce the design cost and never applied during JPEG CODEC IP core design before, hence these two layers of optimizations are not performed by synthesis tools. Since the proposed CODEC achieves reduction of design cost/overhead thus it has been called 'low-cost JPEG CODEC IP core'. Further, Table IV reports the comparison between proposed low-cost obfuscated JPEG CODEC IP core with a non-obfuscated JPEG CODEC IP core in terms of design area, latency, cost and Strength of Obfuscation (SoO). The SoO metric is given as [16]:

$$SoO = a_i/a^T_i \quad (17)$$

where a_i is the number of modified nodes of the DFG due to proposed obfuscation using THT; a_i^T is the total number of nodes before applying obfuscation using THT technique. The SoO metric indicates how strong an obfuscated JPEG CODEC design is concealed in terms of structural identity. The more the design is obfuscated, higher is the complexity in discovering the functionality through the architecture, thus minimizing chances of RE. As shown in Table IV, SoO of 76% is achieved through the proposed approach compared to standard non-obfuscated JPEG CODEC IP. Further, as shown in this table, the change (reduction) in gates due to proposed obfuscation is 10,064. This indicates massive structural transformation at gate level of JPEG CODEC IP core architecture due to transformation in functional resources, multiplexers, demultiplexers and registers. This massive transformation makes the architecture/structure of JPEG IP core un-obvious to an adversary in terms of functionality. An adversary would find it difficult to discover the actual functionality of the design structure. Total six images are selected from datasets [17], [18] to report the compression efficiency obtained through proposed JPEG CODEC IP core. Table V reports the storage size, reduction percentage, Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR) of compressed image for quantization value 90 (Q_{90}). JPEG properties have only been reported in table V to validate

TABLE II
Device utilization summary of proposed IP cores w.r.t. FPGA

Device utilization summary	Total used	Total available	Used %
Logic elements	12148	33216	37
4 input function	5018	11072	45
3 input function	6608	11072	60
≤ 2 input function	494	11072	4
Total register	1826	34593	5
Total pins	322	475	68

TABLE III
Comparison between non-optimized obfuscated JPEG CODEC IP core with proposed low-cost obfuscated IP core in terms of area, latency and cost

Design metrics	Obfuscated JPEG IP core without optimization	Proposed low-cost obfuscated IP core
Resource configuration	4A, 4M	3A, 3M
Design area (μm^2)	397.94	298.45
Design latency (ps)	5860.40	7603.22
Design cost	0.3884	0.3671

TABLE IV
Comparison between non-obfuscated JPEG CODEC IP core with proposed low-cost obfuscated JPEG CODEC IP core

Design metrics	Non-obfuscated JPEG IP core	Proposed low-cost obfuscated IP core	Structure changed due to obfuscation	Improved SoO of proposed design (%)
Resource configuration	4A, 8M, 12 (8:1)Mux, 12 (16:1)Mux, 6 (1:8)Demux, 6 (1:16)Demux, 25 Reg	3A, 3M, 10 (32:1)Mux, 2 (16:1)Mux, 5(1:32) Demux, 1(1:16) Demux, 13 Reg	**10064 Gates**	-
Design area (μm^2)	483.66	298.45	-	-
Design latency (ps)	6533.08	7603.22	-	-
Design cost	0.4582	0.3671	-	-
SoO	0	0.7574	-	**76%**

TABLE V
Storage size, reduction percentage, MSE and PSNR of compressed image for Q_{90} (Images 1 to 6 have been extracted from [17] and [18])

Images	Original size (bits)	Compressed size (bits)	Compression efficiency (%)	MSE	PSNR
Image 1	1048576	236568	77.44	3.84	17.68
Image 2	1048576	200000	80.93	3.50	18.08
Image 3	1048576	170640	83.73	2.00	20.51
Image 4	1048576	194144	81.49	2.63	19.32
Image 5	1048576	210760	79.90	3.25	18.40
Image 6	1048576	216136	79.39	2.20	20.09

(establish) that the designed obfuscated JPEG IP core achieves the desired capability of strong compression efficiency.

Though approaches [11] and [13] offer IP core protection but [11] is key-based functional obfuscation for combinational circuits (does not employ structural obfuscation and target DSP cores); while [13] is electronics signature based protection for protecting authors' rights through detection of cloned copies. However it does not aim to complicate RE as well as does not target structural obfuscation of DSP core. Since [11] and [13] are very different from the proposed method, thus it has not been experimentally compared.

VIII. Conclusion and Future Work

This paper proposed a low-cost obfuscation for JPEG CODEC IP core. Future works aims to amplify the SoO of proposed approach by including stronger structural obfuscation in the form of multi-level transformation in different abstraction levels. We also aim to integrate functional obfuscation with structural obfuscation during DSP core synthesis.

References

[1] R. K. Krishnamurthy, T. Humble, S. C. Cheung, J. Lyke, S. P. Mohanty, and M. Casto, "Energy and Cybersecurity Constraints on Consumer Electronics," http://www.icce.org/expert-panels/, Jan 13, 2018.

[2] J. Dofe and Q. Yu, "Novel Dynamic State-Deflection Method for Gate-Level Design Obfuscation," in *IEEE Trans. on CAD*, vol. 37, no. 2, pp. 273-285, Feb. 2018.

[3] S. P. Mohanty, "Everything you Wanted to Know about Internet of Things, IEEE CE Society DL, 2017.

[4] Q. Tang, M. Groba, E. Juarez, C. Sanz, and F. Pescador, "Real-Time Power-Consumption Control System for Multimedia Mobile Devices," *IEEE Trans. on CE*, vol. 62, no. 4, pp. 362– 370, 2016.

[5] L. Li and H. Zhou, "Structural transformation for best-possible obfuscation of sequential circuits," *Int. Sym. on HOST*, 2013, pp. 55-60.

[6] 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.*, vol. 36, no. 4, pp. 655–668, 2017.

[7] F. Pescador, G. Maturana, M. J. Garrido, E. Juarez, and C. Sanz, "An H.264 Video Decoder based on a Latest Generation DSP," *IEEE Trans. on CE*, vol. 55, no. 1, pp. 205–212, Feb 2009.

[8] B. Shakya, N. Asadizanjani, D. Forte and M. Tehranipoor, "Chip editor: Leveraging circuit edit for logic obfuscation and trusted fabrication," *IEEE/ACM Int. Conf. on ICCAD*, Austin, 2016, pp. 1-8.

[9] D. Roy and A. Sengupta, "Low Overhead Symmetrical Protection of Reusable IP Core using Robust Fingerprinting and Watermarking During High Level Synthesis," *Future Generation Computer Systems*, vol. 71, pp. 89 – 101, 2017.

[10] J. L. Wong, D. Kirovski, and M. Potkonjak, "Computational Forensic Techniques for Intellectual Property Protection*," IEEE Trans. on CAD of Integrated Circuits and Sys.*, vol. 23, no. 6, pp. 987–994, Jun 2004.

[11] R. S. Chakraborty and S. Bhunia, "RTL Hardware IP Protection Using Key-Based Control and Data Flow Obfuscation," in *Proc. 23rd Int. Conf. on VLSI Design*, 2010, pp. 405-410.

[12] Y. Lao and K. K. Parhi, "Obfuscating DSP Circuits via High-Level Transformations," *IEEE Trans. VLSI*, vol. 23, No. 5, pp. 819–830, 2015.

[13] E. Castillo, U. Meyer-Baese, A. Garcia, L. Parrilla and A. Lloris, "IPP@HDL: Efficient Intellectual Property Protection Scheme for IP Cores," *IEEE Trans. on VLSI Systems*, vol. 15, No. 5, pp. 578-591, 2007.

[14] J. Yang, G. Zhu, and Y. Q. Shi, "Analyzing the Effect of JPEG Compression on Local Variance of Image Intensity," *IEEE Trans. on Image Processing*, vol. 25, no. 6, pp. 2647–2656, Jun 2016.

[15] A. Sengupta, D. Roy, S. P. Mohanty, and P. Corcoran, "A Framework for Hardware Efficient Reusable IP Core for Grayscale Image CODEC," *IEEE Access Journal*, vol. 6, 2018, pp. 871-882.

[16] A. Sengupta, D. Roy, S. P. Mohanty, and P. Corcoran, "DSP Design Protection in CE through Algorithmic Transformation Based Structural Obfuscation," *IEEE Trans. on CE*, vol. 63, no. 4, Nov 2017.

[17] "Dataset of standard 512512 grayscale test images," http://decsai.ugr.es/cvg/CG/base.htm, last Accessed on 23 Nov 2017.

[18] "NASA image and video library," https://images.nasa.gov/n#/.

[19] "NanGate 15 nm open cell library," http://www.nangate.com/?pageid =2328, 2017.

Anirban Sengupta (M'09, SM'17) is a Faculty (Associate Professor appointment approved) in ComputerScience and Engineering at Indian Institute of Technology Indore. He is IEEE Distinguished Lecturer of IEEE CE Society. He is an author of more than 150 publications. He is currently Senior Editor of IEEE Consumer Electronics Magazine.

Dipanjan Roy (S'16) is a research scholar in Computer Science and Engineering at Indian Institute of Technology (I.I.T) Indore. He worked as a software development engineer in "Amazon Development Center, Bangalore.

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 250 peer reviewed publications and 3 books. He is IEEE Distinguished Lecturer of IEEE CE Society. He is the Editor-in-Chief of IEEE Consumer Electronics Magazine.

Peter Corcoran (F' 10) is 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. He is co-author on 300+ technical publications and co-inventor on 300+ granted US patents.