Protecting IP core during architectural synthesis using HLT-based obfuscation

A. Sengupta✉ and D. Roy

For protecting an intellectual property (IP) core, it must be harder to reverse engineer. Structural obfuscation can play an important role in achieving this goal. A novel structural obfuscation methodology during architectural synthesis using multiple compiler-based high-level transformations (HLTs) that yield functionally equivalent designs (data flow graphs) which are camouflaged in identity is proposed. The proposed obfuscation methodology is driven through a number of HLT techniques such as redundant operation elimination, logic transformation and tree height transformation. In addition to performing obfuscation, performing area–delay tradeoff during exploring low-cost obfuscated design is also possible using these HLT techniques in the proposed methodology. Owing to multiple stages of HLT incorporated in the proposed approach during obfuscation, it yields a highly robust design which on integration with particle swarm optimisation-based exploration framework produced low-cost obfuscated IP designs. Results of the proposed approach yielded an enhancement in strength of obfuscation of 20.19% and reduction in obfuscated design cost of 59.66% compared with a similar approach.

Introduction: With the mounting popularity of the reusable intellectual property (IP) cores, security threats such as reverse engineering, piracy and hardware Trojan infection have become a serious problem for electronic designs. It is estimated that 10% of the globally sold electronic products are counterfeited that leads to ~$100 billion of revenue loss [1]. Obfuscation is a process of transforming an original design into its functionally equivalent form that significantly enhances the reverse engineering complexity [2]. Although there has been prior literature which targets obfuscation-based IP core protection at lower design abstraction levels; however, there is absolutely no work that provides compiler driven high-level transformation (HLT)-based obfuscation for protection of reusable IP cores at architecture level. An obfuscation which incurs minimal design cost provides high robustness and retains correct functionality for obscuring the structure of a reusable IP core at architecture level is critical for the present day complex electronic designs.

More explicitly, no approach in the literature has proposed a compiler-based multi-stage HLT driven obfuscation for robust protection of IP core at architecture level. However, few approaches such as [3, 4] have applied single-stage obfuscation for digital signal processing (DSP) circuits. Unlike the proposed approach, which executes obfuscation on the data flow graph (DFG) representations of reusable IP core, [3, 4] perform on DSP circuits. Moreover, [3, 4] (being single-stage obfuscation technique) is not as robust as the proposed approach. Finally, [3, 4] incurs higher design cost than the proposed approach.

Problem formulation: For a given DFG and user provided constraints for area (A_c) and delay (L_c), explore the design space to determine a low-cost obfuscated design solution during architectural synthesis. The generated solution should minimise the obfuscated design cost [as shown in (1) below] while satisfying user area–delay constraints; 'A' and 'L' are area and delay of an obfuscated design solution, whereas 'A_{max}' and 'L_{max}' indicate maximum values of area and delay of an obfuscated design solution in the design space

$$C(\text{obf}) = \varphi_1 \frac{L - L_c}{L_{max}} + \varphi_2 \frac{A - A_c}{A_{max}} \quad (1)$$

Proposed low-cost obfuscation framework: In the proposed approach, structural obfuscation is achieved through multiple stages of compiler-based HLT which includes: (a) '*redundant operation elimination*' (*ROE*), (b) '*logic transformation*' (*LT*), (c) '*tree height transformation*' (*THT*). The primary reason to employ HLTs for the proposed obfuscation of reusable IP core during architectural synthesis is its ability to generate many camouflaged (but functionally equivalent) DFG designs that lead to ambiguity. The framework of the proposed low-cost obfuscation methodology is shown in Fig. 1. The proposed approach accepts as inputs an original design of the application in the form of DFG, module library and user constraints (area–delay) and generates a low-cost optimised obfuscated IP design as output. The low-cost obfuscated IP is obtained by processing through the particle swarm optimisation design space exploration (PSO-DSE) block [5]. The PSO-DSE (where each particle encoding indicates a resource configuration for implementing an obfuscated IP) receives the obfuscated DFG from the obfuscation block, evaluates the fitness, determines the local and global best solutions and finally yields a low-cost optimised obfuscated IP design solution.

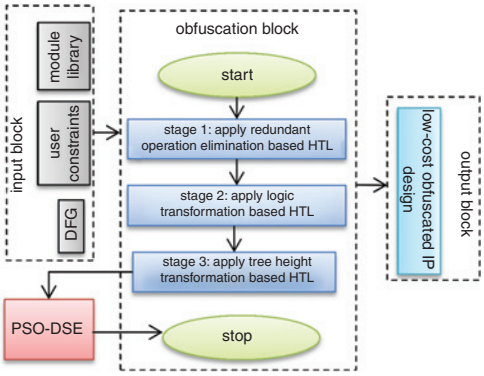

Fig. 1 *Proposed low-cost obfuscation framework*

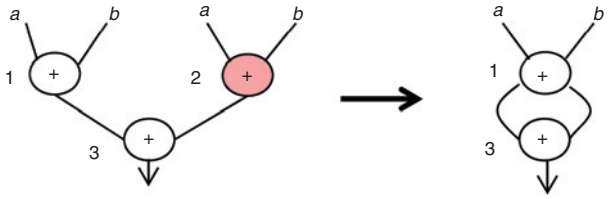

Fig. 2 *Example for ROE*

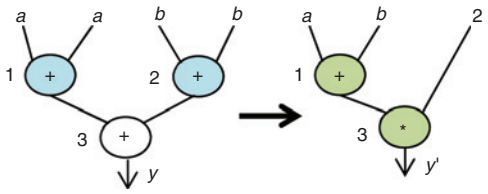

Fig. 3 *Example for logic transformation*

One of the HLTs which is applied on the input DFG to obfuscate is ROE. This technique scans top to bottom of the complete DFG, identifies the redundant operations (*i.e. operations with same input and computation type, like another operation in the graph*), eliminates them and performs necessary adjustments in the graph. For example, in Fig. 2 a redundant operation is node 2 (marked as red) in the original design which is eliminated through the proposed approach to structurally obfuscate the design. To maintain the correctness of the output, both the inputs of node 3 are taken from node 1 in the obfuscated design. Another HLT which is applied on the DFG to obfuscate is 'logic transformation' that is responsible for modifying a DFG with some different logically equivalent functions. It modifies the graph such that the graph looks obfuscated than the original yet obeys the correct functionality. For example, in Fig. 3 the graph on the right-hand side is the LT-driven obfuscated form of the original graph (on the left-hand side); however, both produce functionally equivalent outputs. Another HLT which is applied on the DFG to obfuscate is 'THT' that is responsible for increase or decreases in height of the DFG. For example in Fig. 4, structurally obfuscated form of the original graph (on the left-hand side) is obtained by breaking the critical path dependency into temporary sub-computations which are functionally equivalent. In the proposed approach, we have performed the three aforesaid HLTs in successive stages (see Fig. 1) to obtain higher robustness during obfuscation.

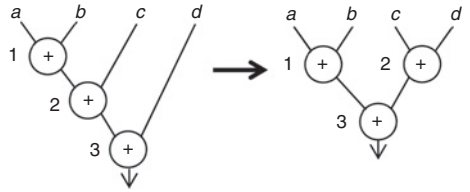

Fig. 4 *Example for THT*

Motivational example: Fig. 5 shows the original, non-obfuscated DFG of infinite impulse response (IIR) filter. Orange coloured nodes represent multiplier and purple coloured nodes represent adder in the graph. The integer value beside each node indicates the corresponding node number. As shown in Fig. 5, $x(n)$, $x(n-1)$ and $x(n-2)$ represent the input variables for the filter in time domain; $y(n)$ and $y(n-2)$ represent the current and the previous output of the filter, respectively, in time domain. A, B and C represent the constant values. The total number of nodes is 9 and the height of the tree is 5. Fig. 6 shows the functionally equivalent, structurally obfuscated design of IIR filter. All the nodes have been modified, except node 1. The total number of nodes is reduced from 9 to 7; tree height is reduced from 5 to 4 in obfuscated design compared with the original non-obfuscated design (as shown in Fig. 5). Fig. 7 shows the low-cost obfuscated IP design of IIR filter scheduled based on two adders and two multipliers (obtained through PSO-DSE block).

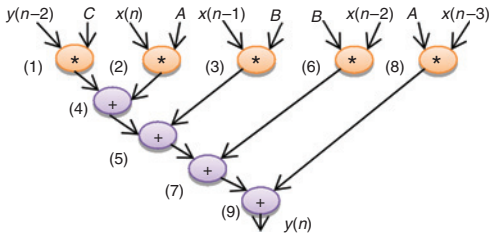

Fig. 5 *Original non-obfuscated DFG of IIR filter*

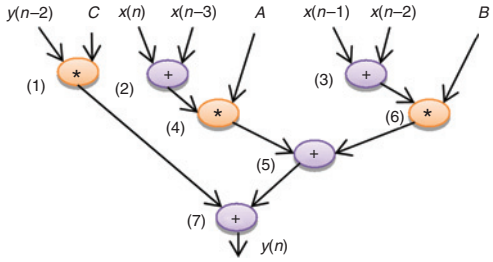

Fig. 6 *Obfuscated IIR filter*

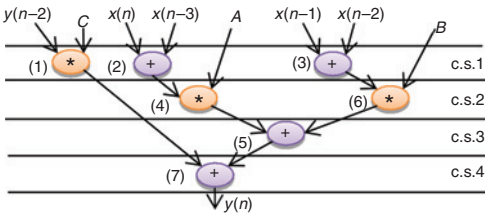

Fig. 7 *Low-cost obfuscated IIR filter IP design*

Experimental result: The proposed approach is implemented in Java and executed on Intel Core-i5-3210M CPU with 4 GB double data rate type three (DDR3) memory at 2.5 GHz. Design cost comparison of the proposed obfuscation-based approach with approach [4] is shown in Fig. 8. The result indicates that the proposed approach while abiding by user resource constraints, obtains lower obfuscated design cost (by 59.66%). On the contrary, [4] incurs more hardware and latency while providing obfuscation-based protection. The proposed approach obtains lower obfuscated design cost as PSO-based DSE and multiple HLTs are performed jointly. Furthermore for the proposed approach, strength of obfuscation (SoO) is higher (by 20.16%) than [4] which indicates stronger (robust) IP protection as shown in Fig. 9. This is because the proposed approach performs multi-stage HLT in succession for more camouflaging (robustness). SoO represents a normalised value between 0 and 1, determined using the following equation:

$$\text{SoO} = \frac{\sum_i^n a_i / a_i^T}{m} \tag{2}$$

where a_i is the number of modified nodes due to the ith HLT technique; a_i^{T} is the total number of nodes before applying the ith HLT technique; m is the total number of HLT techniques applied on a particular application. A node is considered a modified node when either of the following is true:

- A parent node or a primary input of a node of an obfuscated DFG is different than its original.
- The child of a node in an obfuscated DFG is different than its original.
- The resource type of a node in an obfuscated DFG is changed.
- A node of original DFG is non-existent in an obfuscated DFG.

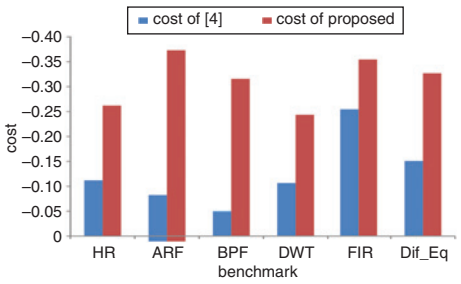

Fig. 8 *Comparison of proposed approach with [4] in terms of cost*

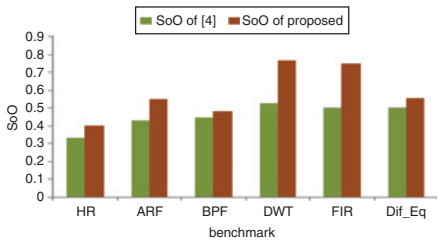

Fig. 9 *Comparison of proposed approach with [4] in terms of SoO*

Conclusion: This Letter proposes a novel multi-stage HLT driven obfuscation methodology during architectural synthesis that provides robust IP protection with low design cost.

Acknowledgment: We thank Indian Institute of Technology Indore for their support in executing this research.

Submitted: *11 April 2017* E-first: *25 May 2017*
doi: 10.1049/el.2017.1329
One or more of the Figures in this Letter are available in colour online.

A. Sengupta and D. Roy (*Computer Science & Engineering, Indian Institute of Technology Indore, Indore, India*)

✉ E-mail: asengupt@iiti.ac.in

References

1 Guajardo, J., Kumar, S.S., Schrijen, G.J., and Tuyls, P.: 'Brand and IP protection with physical unclonable functions'. IEEE Int. Symp. on Circuits and Systems, Seattle, WA, USA, May 2008, pp. 3186–3189
2 Chakraborty, R.S., and Bhunia, S.: 'HARPOON: an obfuscation-based SoC design methodology for hardware protection', *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, 2009, **28**, (10), pp. 1493–1502
3 Parhi, K.K.: 'Verifying equivalence of digital signal processing circuits'. 46th Asilomar Conf. on Signals, Systems and Computers (ASILOMAR), Pacific Grove, CA, USA, November 2012, pp. 99–103
4 Lao, Y., and Parhi, K.K.: 'Obfuscating DSP circuits via high-level transformations', *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, 2015, **23**, (5), pp. 819–830
5 Sengupta, A., and Bhadauria, S.: 'User power-delay budget driven PSO based design space exploration of optimal k-cycle transient fault secured datapath during high level synthesis'. Proc. 16th IEEE Int. Symp. on Quality Electronic Design (ISQED), CA, March 2015, pp. 289–292