Obfuscation of Fault Secured DSP Design through Hybrid Transformation

Anirban Sengupta[1], Shubha Neema[1], Pallabi Sarkar[2], Sri Harsha P[1], Saraju P Mohanty[3], Mrinal Kanti Naskar[2]
[1]Computer Science & Engineering, Indian Institute of Technology Indore, India
[2]Electronics and Telecommunication Engineering, Jadavpur University, India
[1]Computer Science & Engineering, University of North Texas, USA
Email: asengupt@iiti.ac.in; Saraju.Mohanty@unt.edu; mrinalnaskar@yahoo.com

Abstract— **A DSP circuit is considered to be secure, if its functionality is designed to be hidden from an adversary. In other words to make a DSP design secured, its hardware architecture should not look obvious in terms of its functionality. Structural obfuscation plays a critical role in realizing this objective. In the context of transient fault secured DSP circuits, one of the popular design approaches is using dual modular redundancy (DMR). This established design practice makes the functionality of common fault secured DSP circuits architecture easily identifiable to an adversary. In this paper we propose a novel obfuscation in the context of fault secure DSP circuit that uses hybrid transformations in successive layers to completely transform the hardware architecture of the design at register transfer (RT) and gate level without disturbing its functionality and incurring any design overhead. Results indicate without incurring any design cost overhead, the proposed obfuscation achieves significant structural transformation at the gate level such that the functionality becomes un-obvious to an adversary.**

Keywords—fault secure, DSP circuit, obfuscation, transformations

I. INTRODUCTION

Hardware security and Intellectual Property (IP) core protection is an emerging area of research for semiconductor community that focusses on protecting designs against standard threats such as reverse engineering, counterfeit, forgery, malicious hardware modification etc. Hardware security is broadly classified into two types: (a) authentication based approaches (b) obfuscation based approaches. Some of the approaches that fall under the first type are digital watermarking, IP metering, physical unclonable functions etc. The second type of hardware security approach i.e. obfuscation can again be further sub-divided into two types: (i) structural obfuscation (ii) functional obfuscation. Structural obfuscation transforms a design into one that is functionally equivalent to the original but is significantly more difficult to reverse engineer (RE), while the second one is active protection type that locks the design through a secret key. Obfuscation may include altering human readability of hardware description language or encrypting the source code. Converting a design into a form that makes it harder for an adversary to discover its functionality is difficult to reverse engineer i.e. when an IP core functionality is designed to be hidden for an adversary, it is difficult to RE [1, 3, 5].

In the context of transient fault secured DSP circuits, one of the popular design approaches is using dual modular redundancy (DMR) where a duplicate copy of the original unit is made followed by voting [2, 4]. This established design practice makes the functionality of common fault secured DSP circuits architecture easily identifiable to an adversary. The design process of fault secure DSP circuits usually don't employ any protection measures such as obfuscation in the design flow which thus makes it easier for an adversary to discover its functionality. The goal is to *protect* a **fault secure DSP IP core** such that its **functionality is designed to be hidden (not obvious)** to an adversary. In this paper we propose a novel obfuscation in the context of fault secure DSP circuit that uses hybrid transformations in successive layers to completely transform the hardware architecture of the design at RT and gate level without disturbing its functionality and incurring any design overhead.

II. PREVIOUS WORKS

Logic obfuscation uses additional XOR/XNOR gates in circuits to protect the IP core [8] [9] [11]. However this incurs overhead in design due to insertion of additional logic components/circuitry. Further, to effectively implement this approach, determining correct location of key gates is essential. Additionally above approaches have not dealt with obfuscation of fault secured DSP circuits. Further, in [1, 3] structural obfuscation is executed on DSP cores. Approach [1, 3] has not handled protection of fault secured DSP circuits as well as do not perform multiple obfuscations using ROE, THT, LTO and resource transformation. The techniques proposed in [1, 3] are not applicable to obfuscate fault secured design. Our proposed method performs low-cost, multi-stage transformations for DSP designs to realize structural obfuscation. None of the works in the literature performed obfuscating fault secured DSP design. The presented methodology obfuscates a fault secured DSP design through a sequence of transformations at low design cost and zero overhead.

III. PROPOSED APPROACH

A. *Overview of Proposed Methodology*

The inputs to the proposed approach (see Fig. 1) are a control data flow graph (CDFG) or C-code of a DSP core, resource constraint and transient fault strength. Structural obfuscation is achieved by applying the following transformations in sequence: (a) Redundant Operation Elimination (ROE) (b) Logical Transformation Operation (LTO) (c) Tree Height Transformation (THT) (d) resource transformation (RT). Once the obfuscated CDFG is obtained then subsequently it is fed into the fault secured DSP design block. The first step of the fault secured DSP design block is to convert into obfuscated Double Modular

Redundancy (DMR) design, followed by scheduling on the basis of the provided resource constraint. The next step is to apply fault secure hardware allocation rules on obfuscated scheduled DMR. Additionally multiple checkpoints are inserted into the obfuscated scheduled DMR to enhance fault security and perform delay optimization. Section C and D respectively discusses the details of obfuscation based transformation and fault secure design process.

B. Problem Definition and Models:

Generate an obfuscated fault secured DSP circuit at reduced design cost (A_T^{OBF+FS}, T_E^{OBF+FS}) subjected to $k_c = 2$ fault constraint, such that the resultant design is significantly structurally transformed than the original counterpart to hinder identification of functionality and reverse engineering process. Where, A_T^{OBF+FS} and T_E^{OBF+FS} are the area and delay consumed by an obfuscated fault secured DSP design respectively.

Area Model: Total area A_T^{OBF+FS} occupied by an obfuscated fault secured design is used from [9], can be expressed *as*:

$$A^{OBF+FS} = \sum_{i=1}^{n} A(R_i) * N(R_i) + A(mux) * N(mux) + A(reg) * N(reg) \quad (1)$$

Where, $A(R_i)$, $A(mux)$ and $A(reg)$ represent the area of i^{th} resource, multiplexers and one register respectively; $N(R_i)$, $N(mux)$ and $N(reg)$ represent the number of extracted i^{th} resource, mux and registers required respectively. Extracted resources of i^{th} resource is the maximum number of particular resource used in any control step after scheduling the obfuscated design on the basis of user-given resources.

Delay Model: Total delay T_E^{OBF+FS} of the obfuscated fault secured design is calculated as follows:

$$T_E^{OBF+FS} = No\ of\ Control\ Steps \quad (2)$$

Total number of control steps can be termed as the number of total steps required to complete the desired task in obfuscated fault secured design. Here, one control step is taken as 1000ps.

Fitness Function: The cost of solution is evaluated (assuming area occupied and delay) using following:

$$C_f = \emptyset_1 \frac{A^{OBF+FS}}{A^{OBF+FS}_{max}} + \emptyset_2 \frac{T^{OBF+FS}}{T^{OBF+FS}_{max}} \quad (3)$$

Where, C_f is the cost of the solution; A_{max}^{OBF+FS} and T_{max}^{OBF+FS} indicates the maximal area and delay of obfuscated fault secured design respectively. $\emptyset_1$ and $\emptyset_2$ are user specified weightage for area and delay respectively, the magnitude ranging between [0, 1] (*Note: Both* $\emptyset_1$ and $\emptyset_2$ *are kept 0.5 to offer similar priority during cost calculation*)

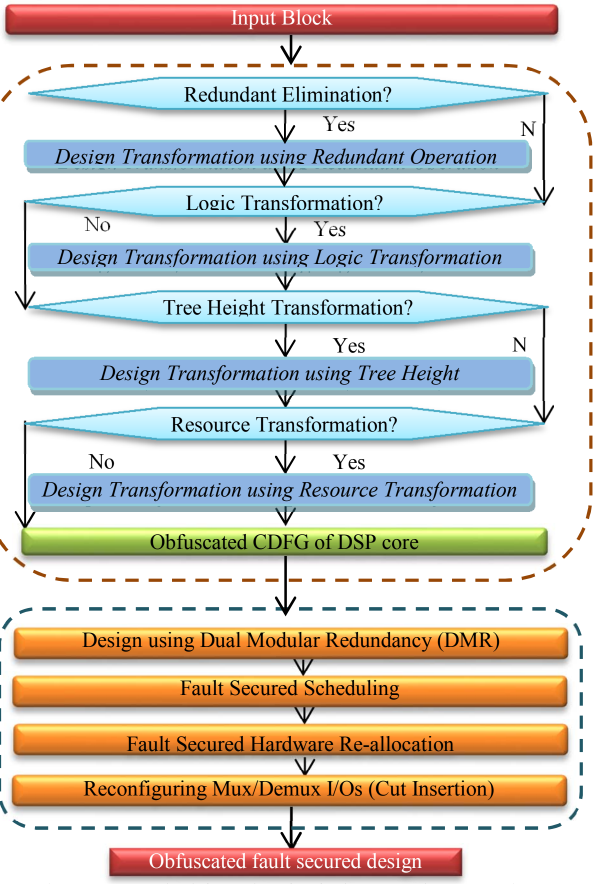
Input Block
Redundant Elimination?
Yes
N
Design Transformation using Redundant Operation
Logic Transformation?
No
Yes
Design Transformation using Logic Transformation
Tree Height Transformation?
Yes
N
Design Transformation using Tree Height
Resource Transformation?
No
Yes
Design Transformation using Resource Transformation
Obfuscated CDFG of DSP core
Design using Dual Modular Redundancy (DMR)
Fault Secured Scheduling
Fault Secured Hardware Re-allocation
Reconfiguring Mux/Demux I/Os (Cut Insertion)
Obfuscated fault secured design

Fig.1. Proposed Obfuscation for fault Secured DSP Designs

C. Methodology for Proposed Obfuscation

Fig. 1 represents the flow chart of our proposed structural obfuscation process; achieved via pursuing four different transformations sequentially. They are: (1) ROE (2) LTO (3) THT [3] (4) RT. Proposed obfuscation is driven by taking the CDFG as input and implementing each of the possible aforementioned transformation. Subsequent subsections contain the comprehensive explanation of all the individual process.

(a) Redundant Operation Elimination Process: Obfuscation through this process is accomplished by elimination of redundant nodes. A node is considered as redundant node when two or more nodes of same inputs and same resource type coexist in the design. Proposed approach eliminates the redundant node(s) by evaluating each node with other nodes in ascending order and if both the nodes entertain the aforementioned conditions, node with higher number is deleted.

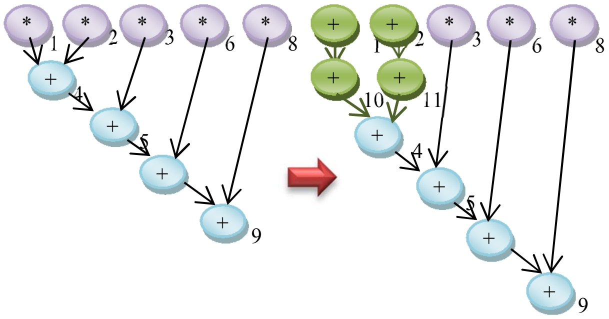

Fig.2. Proposed Logic Transformation based Obfuscation of IIR

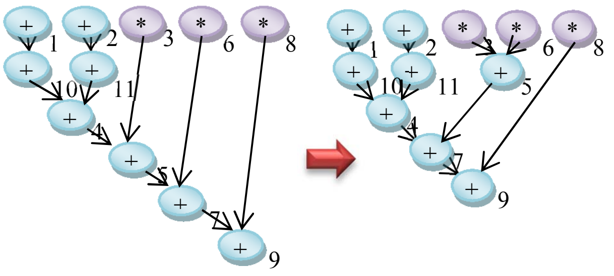

Fig. 3 Proposed THT based obfuscation

Deletion is carried over by required changes in the parent of deleted node's child resulting in keeping the functionality intact (representing change in inputs of resources).

(b) Logic Transformation Process: Another alteration process; applied to obfuscate the design in which nodes of the graph are logically altered but remains functionally equivalent. Proposed approach implements LTO only on those nodes whose inputs are independent. Execution of LTO leads to change in resource type as well as increment in number of total nodes.

Fig. 2 shows an instance representing original and logic transformation based obfuscated design; LTO is practiced on node 1 and 2 with the assumption of their one input as '4'. Newly added nodes (representing changed inputs to resources) are numbered in continuation of highest numbered node before obfuscation. Modified nodes are marked in green colour and the modified dependencies (representing changed muxes/demuxes interconnection) are marked with green line (one multiplication resource is replaced by 2 adders). Nonetheless, as both the designs are functionally alike henceforth yield same results.

(c) Tree Height Transformation Process: This transformation is driven by reduction in the height of the input CDFG. Height is reduced by disjoining the critical path into sub-sections and computing them in concurrency (representing change in inputs and outputs). Fig. 3 presents the IIR benchmark before and after THT. Parallel execution of node 5 is achieved thus resulted both in obfuscation as well as lower height but functionally flawless.

(d) Operation (Resource) Amalgamation: Unlike previous transformation here the process is driven through amalgamation of two resources; adder and multiplier. Proposed approach generates a new customized resource that executes "addition followed by a multiplication" (Fig. 5). However, aforementioned transformation is applicable only on those nodes whose inputs are independent and having adder as resource type whose output is only operative as both the inputs of a multiplier. Moreover, the customized resource is represented by **'#'** in this paper. Fig. 4 shows resource transformation based obfuscation of FIR benchmark.

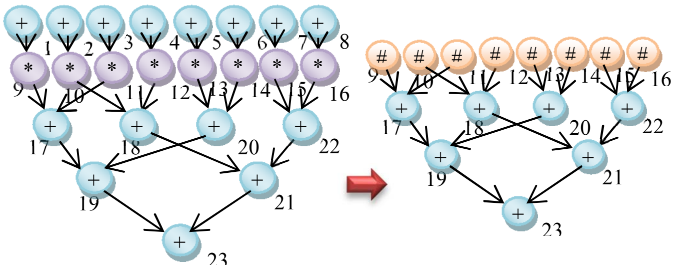

Fig.4. Proposed Resource Transformation based Obfuscation of FIR

D. Transient Fault Detection

The consumer electronics devices/applications need to be secure against transient fault of DSP design. Literature [2], [10] has discussed that single event transient fault occurs due to alpha particles. Same literature also emphasizes about high fault strength which happens to be up to a range of 2000ps, high energy particles causes' temporal effect more than one control step. However, researchers have assumed that transient fault strength due to high energy particle lies between 1000ps to 2000ps (1-k_c-cycles to 2-k_c-cycles). For proposed approach k_c-cycle is taken as 2 (based on real life scenario).

Double Modular Redundancy (DMR) of obfuscated design: The first step to fault security is generation of DMR version of final obfuscated graph. Obfuscated DMR represents original and duplicate units of obfuscated graphs.

Fault security of obfuscated design: Fault which affects the device for very short period of time (1000ps to 2000ps) is termed as Transient Fault, after this period the same resource again starts giving back the desired output. After scheduling the original obfuscated design on the basis of user-given constraints, resources are extracted. Next step is to schedule obfuscated DMR concerning extracted resources. This initial resource configuration is useful in estimating the initial design cost before fault security rules are incorporated. In next paragraph, we present the fault secured hardware allocation/re-scheduling rules, inspired from [2]. The initial obfuscated DMR scheduling obtained, may undergo re-scheduling in order to accommodate the fault security rules. Subsequently the final extracted resource configuration will be obtained after this.

Fault Secured Hardware Allocation/Re-Scheduling Rules

1. If delay (control step) difference of the original and duplicate sister operation is greater than k_c, allocate same operator in original and duplicate operation.

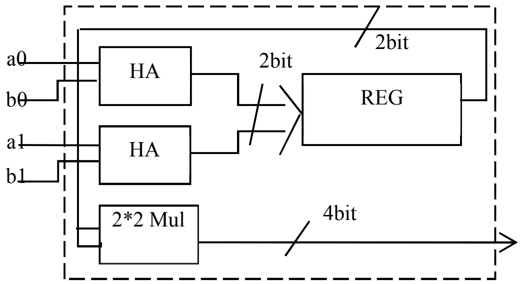

Fig.5. Custom design block used for resource transformation

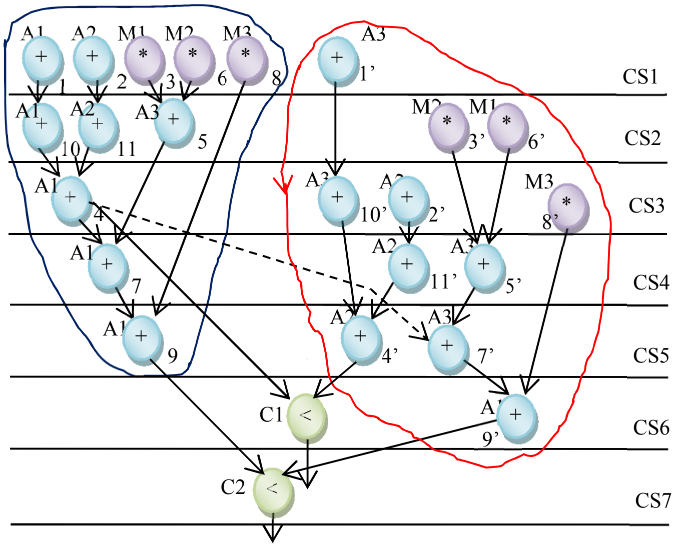

Fig. 6 Proposed DMR design of IIR with checkpoint

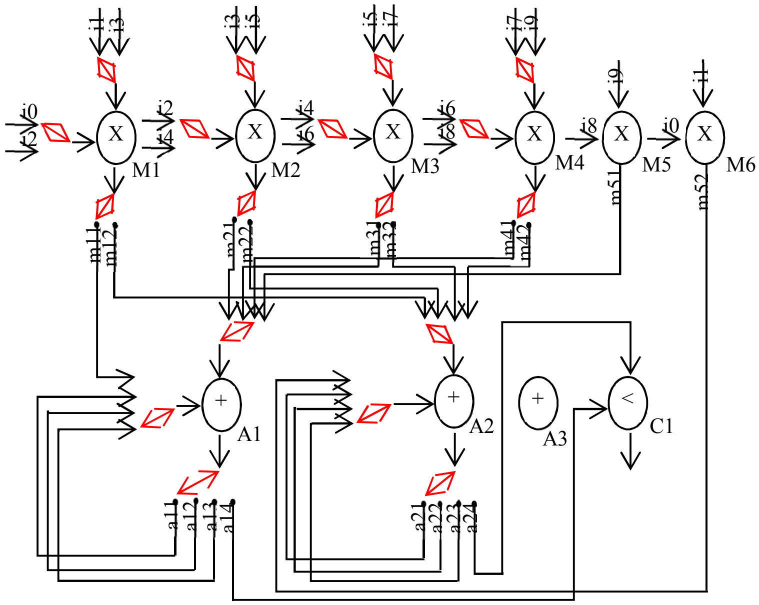

Fig. 7 Non-obfuscated fault secured IIR filter design using [4]

2. If rule 1 is not satisfied and delay difference of original and duplicate operation is not greater than or equal to k_c then allocate distinct hardware resource.
3. If either of the above rules is not satisfied and delay difference of original and duplicate operation is less than k_c, then keep shifting duplicate operation by 1 CS below until either of the above rules is satisfied.

Checkpoint: Delay design or area overhead usually occurs while execution of fault security. Literature [2] proposed one approach to overcome these overheads by insertion of cuts. Insertion of cuts also leads to enhanced fault security with some extra checkpoints. During insertion of cuts, some data dependency edges are cut in graph therefore, shifting some operations one CS up. According to our proposed approach, a cut is performed after generation of obfuscated DMR followed by employing the hardware allocation rules for transient fault security. Proposed approach scales down overhead by implementing maximum checkpoints, therefore, produces prominent security as well. Proposed approach uses different comparators for all the checkpoints; however, area overhead is compensated by significant reduction in latency. Fig. 6 and Fig. 8 represents the complete obfuscated and fault secured DSP design and circuit respectively of IIR benchmark with maximum feasible checkpoints concerning extracted resources and additional comparators. Further, circuit diagram of [4] is demonstrated in Fig. 7.

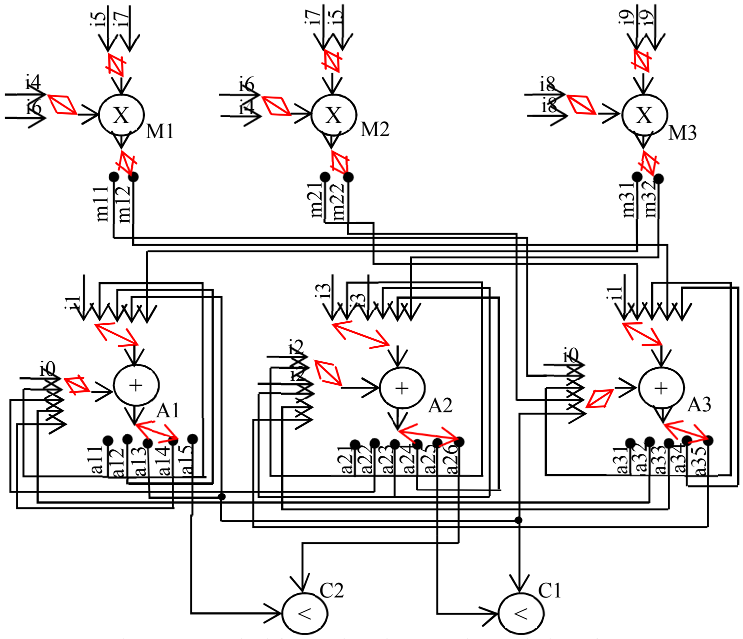

Fig. 8 Proposed Obfuscated Fault Secured DSP design of IIR

IV. EXPERIMENTAL RESULTS

A. Experimental Setup and Benchmark

The proposed approach, non-obfuscated, related works [4], [5] are implemented in object oriented programming language and executed at 1.90GHz for tested DSP cores [7]. Area calculations are performed on 15nm technology scale [6] in terms of NAND gates.

B. Comparison with [5] with respect to Strength of Obfuscation

The measure of inequality in the structure of the circuit after the obfuscation with respect to original circuit is given by Strength of Obfuscation (SoO). The SoO as shown in equation (4) is calculated by counting the unique nodes which are modified after performing aforementioned alterations (Fig.1).

$$\mathrm{SoO} = \frac{\sum_{i=1}^{m} \text{Number of unique nodes modified}}{\text{Number of nodes before obfuscation}} \qquad (4)$$

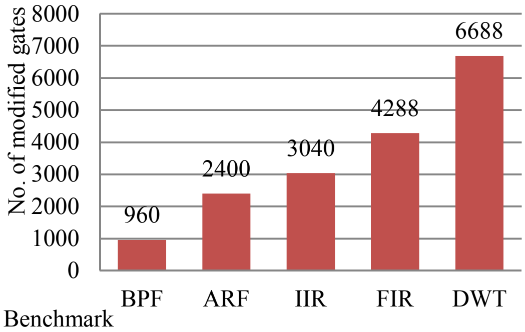

Fig. 9 Number of gates affected (structural change through proposed obfuscation)

Greater the magnitude of SoO, more robust is the security of the design, hence, hard to reverse engineer the circuit. A node is assumed to be transmuted if either of the condition is true from [5]. Furthermore, the enhancement achieved by proposed approach with respect to [5] and number of unique nodes modified after each transformation can be seen in Table I.

C. *Results of Proposed approach*

Occasionally an obfuscated fault secured design attains lesser latency than a non-obfuscated fault secured design since the presented methodology executes a sequence of transformations and optimizations (such as ROE, THT, LT, resource transformations etc.). Hence in some particular situation, the graph post-scheduling could yield lower latency. Table II contains details of proposed approach (with checkpointing) and [4]. Here, area calculation of proposed approach uses extraction of resource whereas [4] considers the user-given resources for measurement of area. Table II is an evidence of significant decline in the cost with obfuscation and multi-checkpointing as additional factor as compare to [4]. Besides, the number of gates affected with respect to [4] due to proposed obfuscation and fault security is shown in Fig. 9.

V. INFERENCE

This work introduced a new methodology for obfuscation of fault secured DSP circuits with zero overhead. The proposed approach achieves significant transformation of architecture without change in functionality.

ACKNOWLEDGEMENT

This Publication is an outcome of the R&D work undertaken in the project under the Visvesvaraya PhD Scheme of Ministry of Electronics & Information Technology, Government of India, being implemented by Digital India Corporation (formerly Media Lab Asia) and also financially supported by Council of Scientific and Industrial Research under sanctioned grant no. 22/730/17/EMR-II

REFERENCES

[1] Y. Lao and K. K. Parhi, “Obfuscating DSP Circuits via High-Level Transformations,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 5, pp. 819–830, May 2015.

[2] Anirban Sengupta, Deepak Kachave "Low Cost Fault Tolerance against kc-cycle and km-unit Transient for Loop Based Control Data Flow Graphs during Physically Aware High Level Synthesis", *Elsevier Journal on Microelectronics Reliability*, Volume 74, 2017, pp. 88-99

[3] A. Sengupta and D. Roy, “Protecting an intellectual property core during architectural synthesis using high-level transformation based obfuscation,” *Electronics Letters*, May 2017. [Online]. Available: http://digital-library.theiet.org/content/journals/10.1049/el.2017.1329.

[4] Inoue, T., Henmi, H., Yoshikawa, Y., &amp; Ichihara, H. High-Level Synthesis for Multi-Cycle transient fault Tolerant Datapaths. Proc. 17 th IEEE International On-Line Testing Symposium, 2011, pp 13-18.

[5] Anirban Sengupta, Dipanjan Roy, Saraju Mohanty, Peter Corcoran "DSP Design Protection in CE through Algorithmic Transformation Based Structural Obfuscation", IEEE Transactions on Consumer Electronics,

Table I. Comparative study of proposed obfuscation with [5] with respect to obfuscation strength

Benchmark	Proposed approach					[5]	Enhancement in Strength of Obfuscation (%)
	REO (Unique nodes)	LTO (Unique nodes)	THT (Unique nodes)	ALU (Unique nodes)	SoO		
IIR	-	3	3	-	0.666667	0.33333	100
ARF	10	6	-	-	0.571429	0.42857	33.33
BPF	5	6	2	2	0.517241	0.44827	15.38
DWT	-	10	-	-	0.588235	0.52941	11.11
FIR	-	-	12	11	1	0.5	100

Table II Comparison of proposed obfuscated fault secured approach with multi-checkpointing with [4]

Benchmark	Proposed approach (with multi-checkpointing)					[4]			
	# cuts	Resources	Area (um^2)	Latency (ns)	Cost	Resources	Area (um^2)	Latency (ns)	Cost
IIR	1	3A,3M,2C	354.681	7	0.497980	3A,6M,1C	545.195	7	0.654878
ARF	4	5A,6M,5C	718.505	10	0.526472	7A,7M,1C	747.013	10	0.542274
BPF	4	5A,2M,	525.73	11	0.550747	6A,4M,1C	545.686	10	0.555902
DWT	4	5A,1M,5C	340.722	12	0.553910	3A,3M,1C	350.946	13	0.578149
FIR	10	4A,4ALU,11C	686.556	6	0.351802	8A,8M,1C	837.945	11	0.521300

Volume 63, Issue 4, November 2017

[6] NanGate 15 nm open cell library. [Online]. Available: http: //www.nangate.com/?pageid=2328.

[7] DSP benchmark suite: http://www.ece.ucsb.edu/ EXPRESS/benchmark/

[8] J. Zhang, “A Practical Logic Obfuscation Technique for Hardware Security,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 3, pp. 1193–1197, March 2016.

[9] 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 2016 International Great Lakes Symposium on VLSI*, May 2016, pp. 133–136.

[10] Gaillard, Rémi. "Single event effects: Mechanisms and classification." *Soft Errors in Modern Electronic Systems*, pp. 27-54. Springer, 2011.

[11] J. A. Roy, F. Koushanfar, and I. L. Markov, “EPIC: Ending Piracy of Integrated Circuits,” *in 2008 Design, Automation and Test in Europe*, March 2008, pp. 1069–1074.