Obfuscation of Fault Secured DSP Design through Hybrid Transformation

Anirban Sengupta¹, Shubha Neema¹, Pallabi Sarkar², Sri Harsha P¹, Saraju P Mohanty¹, Mrinal Kanti Naskar²
¹Computer Science & Engineering, Indian Institute of Technology Indore, India
²Electronics and Telecommunication Engineering, Jadavpur University, India

Email: asengupta@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 \( (A_{\text{OBF-FS}}, T_{\text{OBF-FS}}) \) 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, \( A_{\text{OBF-FS}} \) and \( T_{\text{OBF-FS}} \) are the area and delay consumed by an obfuscated fault secured DSP design respectively.

**Area Model:** Total area \( A_{\text{OBF-FS}} \) occupied by an obfuscated fault secured design is used from [9], can be expressed as:
\[
A_{\text{OBF-FS}} = \sum_{i=1}^{n} A(R_i) \cdot N(R_i) + A(\text{mux}) \cdot N(\text{mux}) + A(\text{reg}) \cdot N(\text{reg}) \quad (1)
\]
Where, \( A(R_i) \), \( A(\text{mux}) \) and \( A(\text{reg}) \) represent the area of \( i^{th} \) resource, multiplexers and one register respectively; \( N(R_i), N(\text{mux}) \) and \( N(\text{reg}) \) 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 \( T_{\text{OBF-FS}} \) of the obfuscated fault secured design is calculated as follows:
\[
T_{\text{OBF-FS}} = \text{No of Control Steps} \quad (2)
\]
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:
\[
C_t = \phi_1 A_{\text{OBF-FS}}^{\text{max}} + \phi_2 T_{\text{OBF-FS}}^{\text{max}} \quad (3)
\]
Where, \( C_t \) is the cost of solution; \( A_{\text{max}}^{\text{OBF-FS}} \) and \( T_{\text{max}}^{\text{OBF-FS}} \) indicates the maximal area and delay of obfuscated fault secured design respectively. \( \phi_1 \) and \( \phi_2 \) are user specified weightage for area and delay respectively, the magnitude ranging between [0, 1] (Note: Both \( \phi_1 \) and \( \phi_2 \) are kept 0.5 to offer similar priority during cost calculation)

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.1. Proposed Obfuscation for fault Secured DSP Designs](Image)

![Fig.2. Proposed Logic Transformation based Obfuscation of IIR](Image)
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.

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- cycles to 2-k- cycles). For proposed approach k- 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, allocate same operator in original and duplicate operation.
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.

**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).

$$\text{SoO} = \frac{\text{Number of unique nodes modified}}{\text{Number of nodes before obfuscation}}$$  \hspace{1cm} (4)
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. INFERENCES

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


