B. TECH. PROJECT REPORT

On

Fault Security of Digital Processors for Aerospace Electronic System

BY

P. Sri Harsha

DISCIPLINE OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

December 2017

Fault Security of Digital Processors for Aerospace Electronic System

A PROJECT REPORT

Submitted in partial fulfillment of the requirements for the award of the degrees

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Submitted by:

P. Sri Harsha

Guided by:

Dr. Anirban Sengupta

Computer Science and Engineering

Assistant Professor, IIT INDORE

INDIAN INSTITUTE OF TECHNOLOGY INDORE

December 2017

CANDIDATE'S DECLARATION

I hereby declare that the project entitled **"Fault Security of Digital Processors for Aerospace Electronic System"** submitted in partial fulfillment for the award of the degree of Bachelor of Technology in 'Computer Science and Engineering' completed under the supervision of **Dr. Anirban Sengupta, Computer Science and Engineering, Assistant Professor,** IIT Indore is an authentic work.

Further, I declare that I have not submitted this work for the award of any other degree elsewhere.

P. Sri Harsha (140001021)

CERTIFICATE by BTP Guide(s)

It is certified that the above statement made by the student is correct to the best of my knowledge.

Dr. Anirban Sengupta

Assistant Professor

Computer Science and Engineering

IIT Indore

Preface

This report on “Fault Security of Digital Processors for Aerospace Electronic System” is prepared under the guidance of Dr. Anirban Sengupta.

I have tried to present the detailed concept of obfuscation and transient fault. Through this report the explanation and all the transformation for obfuscation is shown diagrammatically. As well as the algorithm for making the design fault secure is present. For better understanding of my concept one complete obfuscated and fault design is added with all the steps from base case design i.e. design which is neither obfuscated nor secure. To conclude the report comparison of my obfuscation with previous work as well as comparison of cost of obfuscated and fault secure design with related work is included.

P. Sri Harsha
B.Tech. IV Year
Discipline of Computer Science and Engineering
IIT Indore

Acknowledgments

I wish to thank Dr. Anirban Sengupta for his kind support and valuable guidance.

It is his help and support, due to which I became able to complete the design and technical report.

Without his support this report would not have been possible.It was only possible because of his enthusiasm, beforehand schedule, knowledge and sincerity towards me and my work to produce the better result.I wouldn't have achieved my goals without his encouragement at each step.

P. Sri Harsha
B.Tech. IV Year
Discipline of Computer Science and Engineering
IIT Indore

Abstract

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 highlevel 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.Another HLT technique contains our customized ALU.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.

Fault security indicates ability to provide error detection or fetch correct output. Fault security assures possibility of using either hardware redundancy or time redundancy to optimize the overheads associated with fault security. However, optimal fault secured datapath structure based on user power–delay budget during high level synthesis (HLS) in the context k-cycle transient fault is considered an intractable problem. This is due to the fact that for every type of candidate design a feasible k-cycle fault secured datapath may not exist which satisfies the conflicting user constraints/budget. The project therefore presents the obfuscated and fault secure design.

Contents

List of Figures

List of Tables

1

Introduction

1.1 What is Obfuscation?

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 $\sim$\$100 billion of revenue loss . Obfuscation is a process of transforming an original design into its functionally equivalent form that significantly enhances the reverse engineering complexity. Our project 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.

1.2 What is Reliability?

DSE in high level synthesis includes searching an optimal datapath from a set of assorted design alternatives of equivalent functionality which offer higher performance and lower power expenditure with complete fault reliability. This DSE process in HLS aims at optimizing conflicting issues like area, power and performance along with certain orthogonal issues like runtime and quality of

results (QoR)However, optimizing only area, power and performance remains no longer sufficient now. This is applicable specifically for current generation of systems which demand designs (especially for space applications where radiation induced faults are highly possible) that require ability to detect errors occurring due to transient fault (such as single event upset). Transient faults are radiation induced faults which are non -permanent in nature. These non-recurring faults can be caused by energized particles, environmental noise or electromagnetic interference. The duration of such faults is in order of a few picoseconds. The occurrence of transient faults has increased due to recent advancements in technology where the packing of millions of transistors on a single chip have become feasible. The increase in density per unit area is negatively impacting the device and overall systems reliability by making it susceptible to transient fault or the Single Event Upset (SEU) especially in space applications. Therefore, to achieve complete reliability of the system, multi cycle transient fault security needs to be considered as design metric (or constraint) during multi-objective DSE in HLS.

2

Overview

For obfuscation we have come through compiler-based multi-stage HLT driver obfuscation. Multi-stage HLT process includes Redundant Elimination Operation (RTO), Logical Transformation Operation (LTO) and Tree Height Transformation Process (THT). Our proposed analysis also produces one different obfuscation by generating a customised ALU. With the RTO process, those nodes which are performing same operation on same inputs are eliminated. Next technique which obfuscates the CDFG is LTO where we assumed a restriction on few independent inputs and converting the resource type of those nodes from multiplication to addition. Unlike RTO and LTO there is no variation in number of nodes but there is a variation in height after THT which evaluates in parallel and results in same output. For ALU transformation to be applicable certain conditions are required, for same to happen addition has to be followed by multiplication where addition has independent inputs.

On the other hand, for the reliability of obfuscated design Double Modular Redundant (DMR) is applied on the scheduled obfuscated design which obeys resource constraint condition with some hardware allocation rules by taking fault strength as input for making the CDFG obfuscated and fault secure. For making it more secure and dropping off the delay we proposed new analysis multi-checkpoint.

Our complete proposed analysis is shown in Fig 2.1 where we have shown all the steps of our HLT techniques and Fault security.

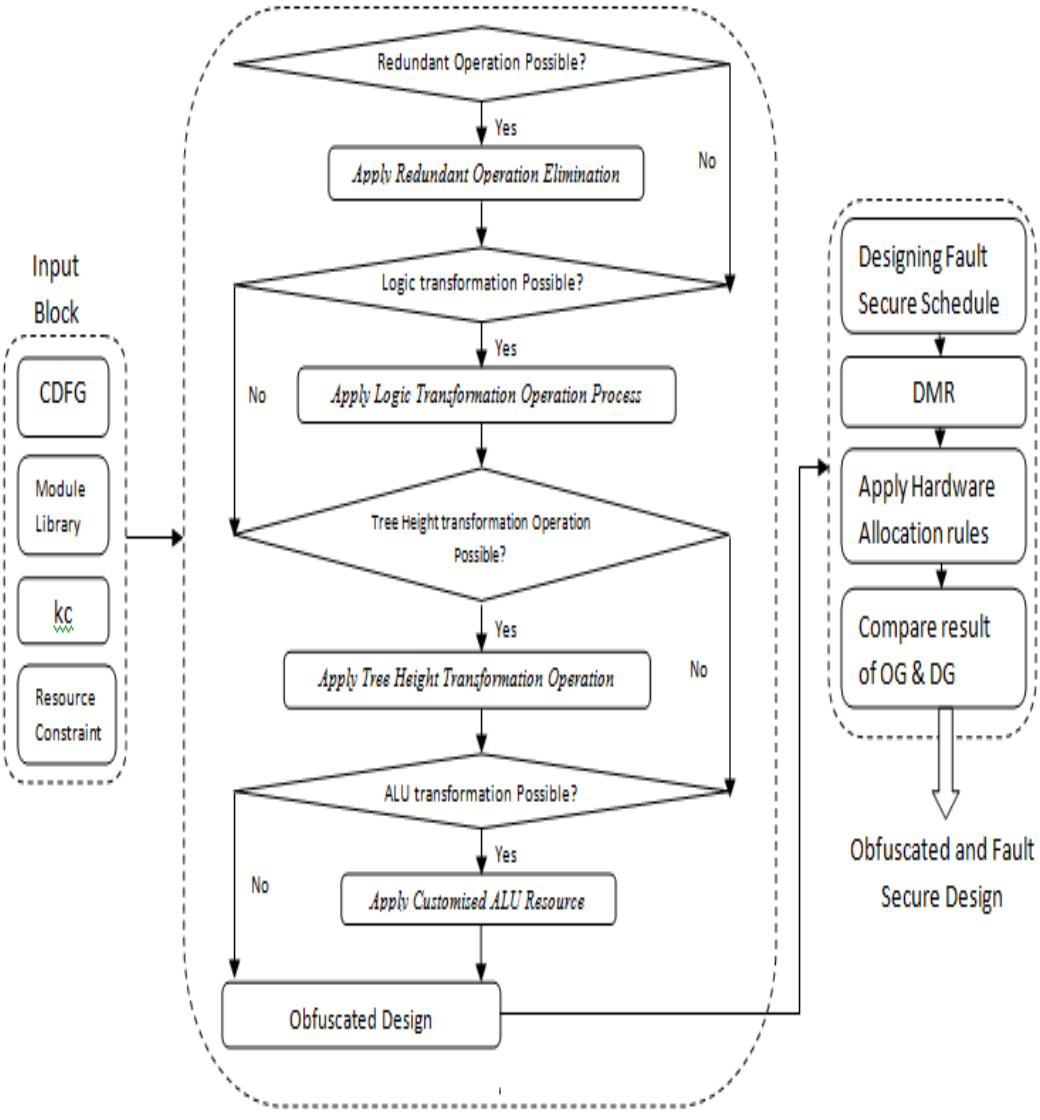

Figure 2.1: Overview of Proposed Analysis

3

High-level transformation

High-level transformations have been known for a long time and have been used in a wide range of applications, and have been used in synthesis of DSP systems. These techniques can be applied at the algorithm or the architecture level to achieve a tradeoff among different metrics of performance, such as area, speed, and power.
Some HLT techniques to obfuscate the CDFG used in this project are as follows:

3.1 Redundant Operation Elimination Process

An HLT technique which is applied to obfuscate the input CDFG by removing redundant nodes from the graph is redundant operation elimination. A node in the input graph is identified as a redundant node if there exist another node which has exactly same parents/inputs and same operation type. In our proposed approach we scan each node based on the node numbers in ascending order. If a pair of nodes is found which have same inputs and operation type then the node having higher node number is identified as a redundant node. These nodes are deleted from the graph and necessary adjustment is performed to maintain the correct functionality of the graph.

For example, in the original design shown in Fig3.1 the redundant operation is node 9, 11, 12, 13, 15 which is eliminated through the proposed approach as shown in Fig 3.2 to structurally obfuscate the design. To maintain the correct-

ness of the output the inputs of node 10, 14 and 16 is taken from node 8, 10 and 14 respectively in the roe-based obfuscated design (Fig 3.2).

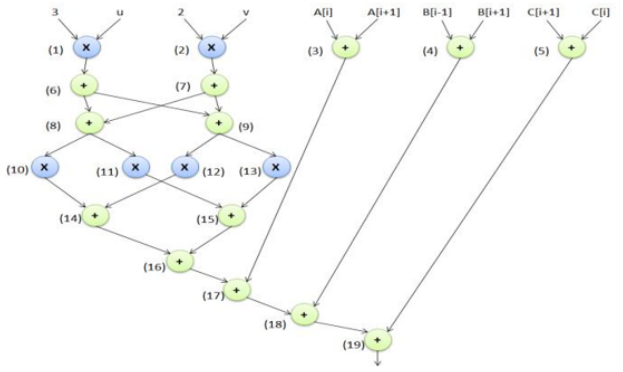

Figure 3.1: Original non-obfuscated CDFG

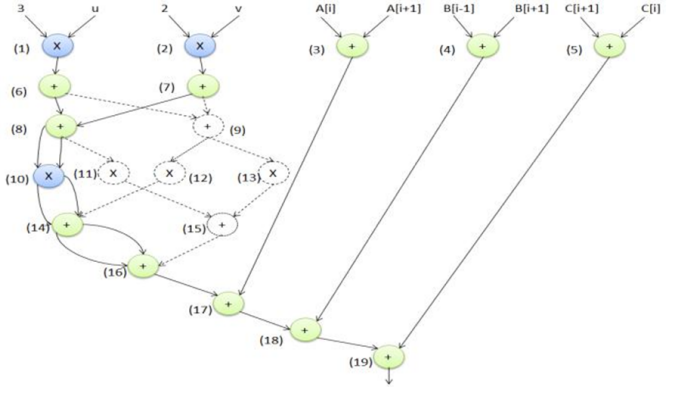

Figure 3.2: Redundant operation elimination based obfuscated design

3.2 Logic Transformation Process

Another HLT technique which is applied to obfuscate the input CDFG by modifying the graph with different logically equivalent function is logic transformation.This HLT technique change the resource type of some independent nodes from multiplication to addition. For those nodes wherever this technique is applied an assumption is made regarding taking inputs as variable and constant 3/4.New nodes are added as a result of LTO those are numbered consecutively after the maximum numbered node. It alters the nodes of the input graph such that the graph looks different than the original still satisfies the functionality correctly. Finally, LT based structurally obfuscated graph is produced as out-

put.

As shown in the Fig 3.3 and Fig 3.4 LTO is applied at the node 1 and 2 where

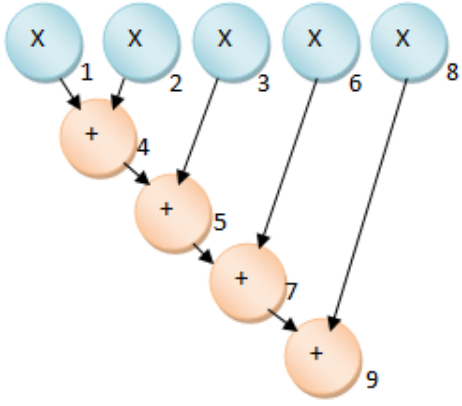

Figure 3.3: CDFG before Logical Transformation Process

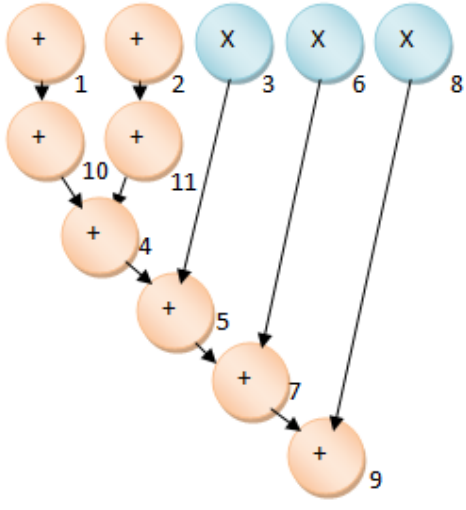

Figure 3.4: CDFG after Logical Transformation Process

we assumed the inputs are 4 and a variable. Thus , 4*v can be divided into (v+v)+(v+v). This results in alteration of nodes.

3.3 Tree Height Transformation Process

Another HLT technique which is responsible for obfuscating the input CDFG by increasing or decreasing the height of the graph is tree height transformation. It divides the critical path dependency into temporary sub-computations and evaluates in parallel, thereby generates structurally dissimilar yet functionally

equivalent graph. Finally, THT based structurally obfuscated graph is produced as output.

Fig 3.5 and Fig 3.6 shows the obufscation produced in the design after per-

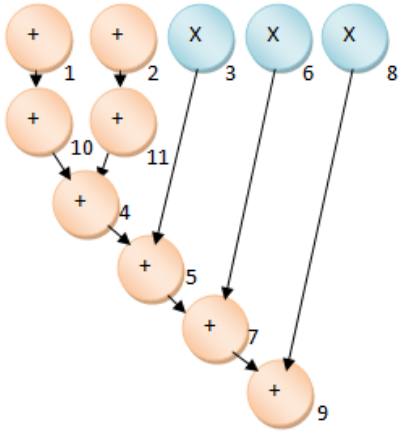

Figure 3.5: CDFG before Tree Height Transformation

forming THT. Three operations are performed parallel out of which operation whose input nodes are 3 and 6 are modified and results in the obufscation of the design.

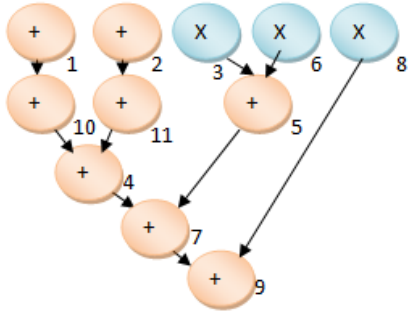

Figure 3.6: CDFG after Tree Height Transformation

3.4 ALU Transformation Process

Another HLT technique which is responsible for obfuscating the input CDFG by performing two operations in a single unit. It happens when there is a adder followed by multiplier whose inputs are independent cum output of the adder should only be the inputs of multiplier as shown in Fig.3.7. In this technique a new resource is produced which is the replacement of the adder and multiplier

where this obfuscation has been done and same can be termed as **Customized ALU** which is represnted by &.

Below shown figure Fig 3.9 is the outcome of the ALU process applied on the

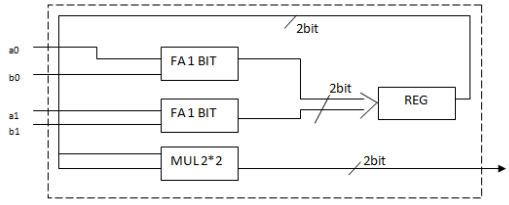

Figure 3.7: Internal block diagram of 2 bit ALU

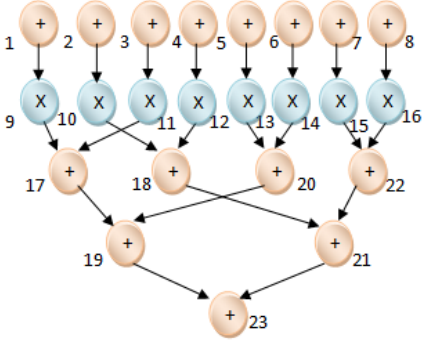

Figure 3.8: CDFG before ALU Transformation

Fig 3.8 where all the adders of the very first control step which obeys the above explain conditions are transformed into new resource &.

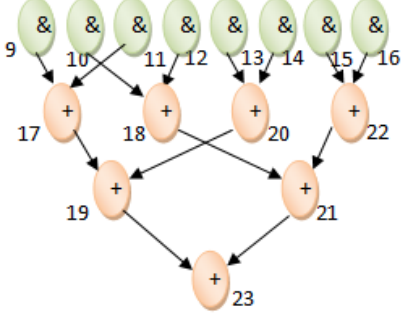

Figure 3.9: CDFG after ALU Transformation

4

Strength of Obfuscation

After performing all the possible High Level Transformation obfuscated design is obtained and its strength is calculated by the following formula :

$$\text{Strength of Obfuscation} = \frac{\text{No. of Unique Nodes modified}}{\text{Total no. of initial Nodes}}$$

A node is considered a modified node when either of the following condition is true :

1. A parent node of a node of an obfuscated DFG is different from original.
2. The child of a node in an obfuscated DFG is different that its original.
3. The resource type of a node in an obfuscated DFG is changed.
4. A node of original DFG is non-existent in an obfuscated DFG.

5

Background on register transfer level fault secured computation

A fault is any condition that causes a functional unit to malfunction. Assuming a single fault model, a system with fault-secure computation either gives out a correct result or signals an error. An operation is generally treated as a secured operation when the following conditions are satisfied: (a) datapath from primary inputs of CDFG to the outputs of this operation of original and duplicate do not share any hardware units. (b) The output of this operation and the output of its duplicate in recomputations are compared. An exception to condition (a) in case of transient faults may exist (i.e. operations of original and duplicate may share the same hardware unit).

A datapath is fault secure if all the operation nodes are covered by at least one

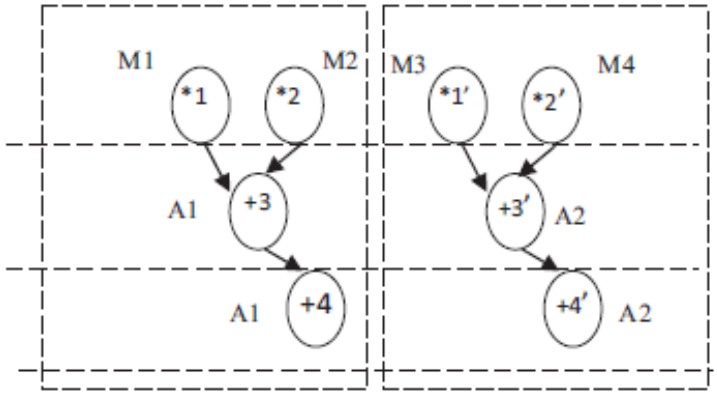

Fig. 1. Original (left) and duplicate (right) unit (DMR system).

Figure 5.1: DMR

secured sub-CDFG. In Fig 5.1 above, the original and duplicate unit of a sample CDFG is shown. Say, a checkpoint is introduced at the output of +4. The result from original computation and the result from duplicate computation (+40) will be compared at this point. This checkpoint secures its source operation +4 and groups all the predecessors of +4 into a sub-CDFG. (In this example, the sub-CDFG is the same as the whole CDFG). The structure presented in this figure is called 'Double Modular Redundant (DMR)' system which provides error detection (by obeying the condition in (a) discussed above) but at the expense of twice the resource (i.e. four multipliers and two adders), if a fault occurs at any operator.

6

kc-cycle transient faults

Due advancement of faster devices is a feature of future technologies that brings major concerns to the fault detection community. This is because as easily predictable, for those technologies, even particles with modest linear energy transfer (LET) values will produce transients lasting longer than the predicted cycle time of circuits. Therefore the technology evolution and LET of particle impact both plays a major role in inducing multi-cycle (k-cycle) transient fault (longer duration transient) in a device. The duration of the k-cycle transient fault is calculated by considering a worst case of the long pulse transient duration due to particle strike. Since the duration of cycle time is known, therefore as a design specification, the value of k-cycle transient fault is determined by considering a worst case duration of the long pulse transient that could affect the circuit. Therefore this worst case transient pulse duration (k-cycle) value is used as a design specification (or fault constraint) before initiating the exploration of optimal k-cycle transient fault secured datapath during HLS. Now let us take an example to demonstrate multi-cycle transient faults.

Given a sample scheduled data flow graph in Fig 6.1. It shows a scheduled data flow graph of an application which uses two multipliers (M1, M2) and one adder (A1). Under standard conditions, the circuit undergoes a traditional computation, thereby generating a feasible error free output. However, if a transient (non-permanent) fault occurs at any unit in the circuit due to particle strike, the corresponding output becomes erroneous, thereby affecting the entire circuit. For example, let us assume, if a 2-cycle fault occurs at Multiplier M1, when the

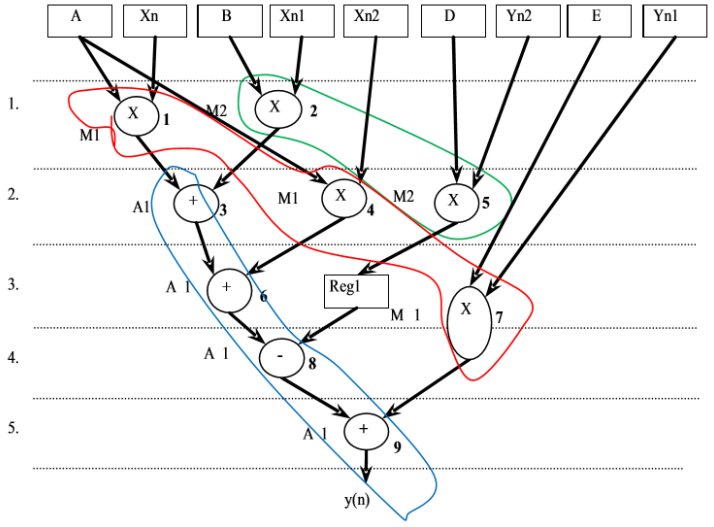

Figure 6.1: Scheduled sequencing graph with data registers

state of the system is in control step 1. As the Kc is 2 then the Multipler M1 will be faulty till 2 consecutive steps. The span of the error affecting similar operators depends upon on Kc (cycle duration). As, M1 is faulty in step 1 then operation 1 gives incorrect output and also, operation 4 gives incorrect output at step 2. But as soon as the system propagates to step 3, the effect of fault generated on M1 normalizes and the fault disappears. Hence, M1 operates correctly for the operation 7 at step 3. Such faults which occur once and then disappear which are termed as transient faults. Once this fault occurs on a logic element of a system, the fault is associated as transient fault of the operator.

7

Conditions for design of k-cycle fault secured DMR system

DMR system is shown in Fig 5.1.The proposed algorithm accepts the following as inputs the CDFG, fault security constraint (kc) indicating the strength of the fault and module library indicating the hardware units available for allocation. Therefore after feeding Kc and the hardware resources then we schedule the original units and duplicate units combining both we get DMR. The pair of units is concurrently scheduled on the basis of as soon as possible scheduling using the user supplied resource constraints and available dependency information of the nodes. After obtaining the scheduled DMR system, the hardware allocation of both the units is performed.

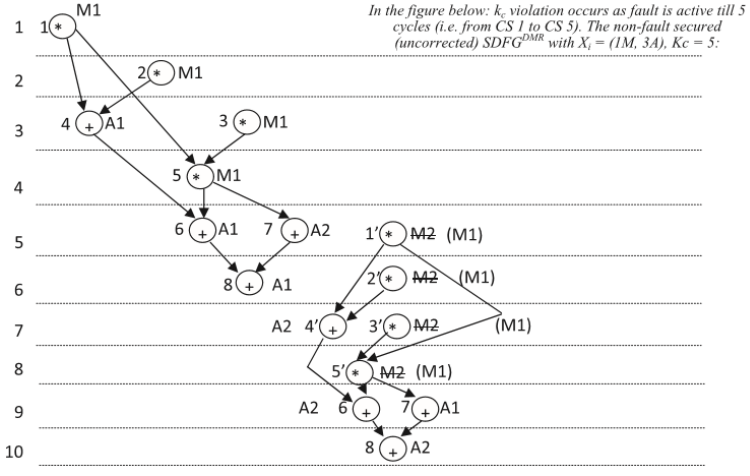

Figure 7.1: Uncorrected 5-cycle fault secured $SDFG^{DMR}$

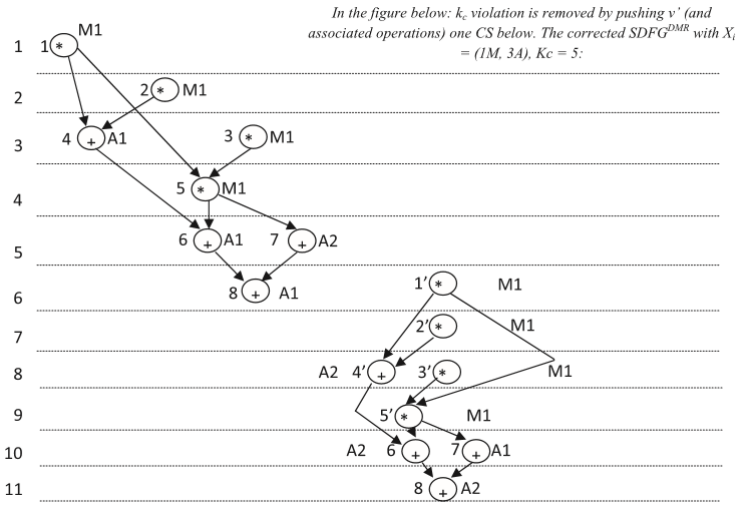

Figure 7.2: Corrected 5-cycle fault secured SDFG^{DMR}

Operations of the SDFGDMR system are allocated to hardware on the basis of following fault security conditions (schemes)

1. Allocate opn (v) $\in U^{OG}$ and opn (v') $\in U^{DP}$ to distinct operators (hardware units) based on availability.

2. If unavailable, then: Keep same assignment for v' (as v) in U^{DP} such that: $t(v') - t(v) \succ k_c$.

3. If above condition is false, then:
   Push v' (and its successors) $\in U^{DP}$ one CS below until above condition is true.

8

Complete Flow

Shown below is a complete example of **IIR Benchmark** whose all the phases are explained through diagrams.

Starting with base case design Fig 8.1 is the actual IIR benchmark which is neither obfuscated nor fault secure. It can be seen that REO transformation can not be applied here as there are no redundant nodes. Therefore, the next possible transformation LTO is applied, after applying LTO at nodes 1 and 2 new obfuscated design in obtained which is shown in very next figure (Fig 8.2). Modified nodes produced are 1 ,2 and 4 by applying the rules as explained in chapter 4.

For next transformation input is taken as the output of LTO transformation.

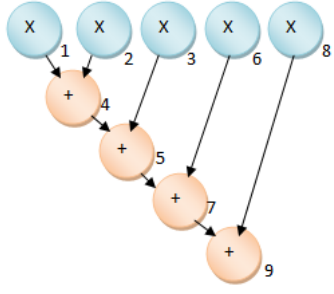

Figure 8.1: Original non-obfuscated CDFG

THT is applied at the Fig 8.2 whose resultant obfuscation is Fig 8.3 where node 3, 5, 6 and 7 are modified according to rules explained in chapter 4.

As after THT there are no adders on the first control step which satisfies the constraints of applying ALU , therefore, ALU transformation can not be applied on this particular benchmark.

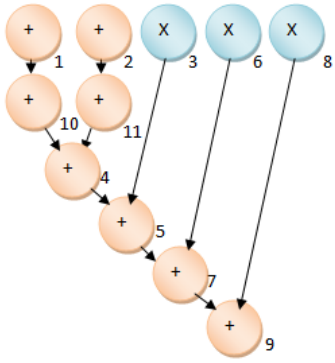

Figure 8.2: CDFG after Logical Transformation Process

After applying all the possible obfuscation on IIR benchmark, the resultant

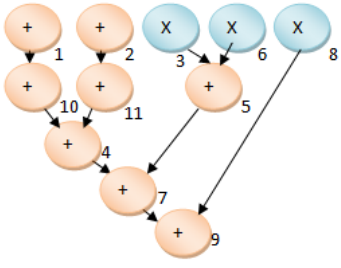

Figure 8.3: CDFG after Tree Height Transformation

obfuscated design is scheduled followed by DMR scheduling on the basis of Hardware allocation rules. Resources constraint used in the following example is **3A, 3M with k_c=2**.

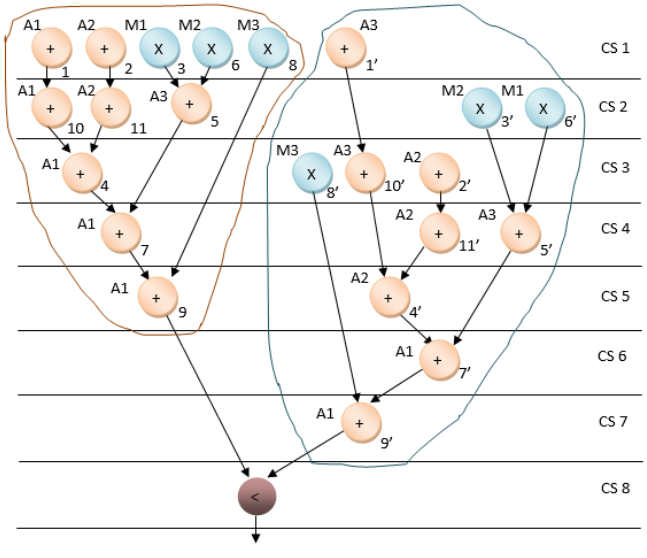

Figure 8.4: Obfuscated and Fault secure CDFG

9

Experimental Results

Table 1: Shows the individual number of nodes modified by each HLT transformation for 5 different benchmark.The Strength of Obfuscation (SoO) calculated by the formula presented in chapter 4 is compared with related work and outcome of comparison is shown in enhancement column in the table.

Table 2: This table shows the complete analysis of the area for extracted re-

Benchmark	Proposed approach					Related	Enhancement %
	REO (Unique nodes)	LTO (Unique nodes)	THT (Unique nodes)	ALU (Unique nodes)	SoO		
IIR	-	3	3	-	0.666667	0.333333	100
ARF	10	6	-	-	0.571429	0.428571	33.33
BPF	5	6	2	2	0.517241	0.448275	15.38
DWT	-	10	-	-	0.588235	0.529411	11.11
FIR	-	-	12	11	1	0.5	100

Table 1: Comparison of SoO of proposed approach with related work

sources which are given in the table, latency and cost by taking k_c as 2.

Table 3: Shows the complete details of non-obfuscated non-secure CDFG by taking into account that k_c is 0.

Table 4: This table compares our work with related work where only obfuscation was done. This table shows that at very less overhead of the cost obfuscation and reliability both can be achieved.

Benchmark	kc	Resources	FU area (um^2)	Mux area (um^2)	Reg area (um^2)	Area (um^2)	Delay (ns)	Cost
IIR	2	3A,3M,1C	300.418	25.559	6.29146	332.268	8	0.50893418
ARF	2	8A,8M,1C	772.278	62.6196	8.65075	843.548	11	0.6086040
BPF	2	5A,2M,1ALU,1C	357.827	75.3992	9.43718	442.663	13	0.5052612
DWT	2	5A,1M,1C	187.171	63.8976	7.07789	258.147	15	0.4900046
FIR	2	4A ,4ALU,1C	473.433	34.5047	3.14573	511.083	8	0.35022486

Table 2: Data based on Proposed Analysis

Benchmark	kc	Resources	FU area (um^2)	Mux area (um^2)	Area (um^2)	Delay (ns)	Cost
IIR	0	1A,3M	245.3674	6.38976	251.757	5	0.35439387
ARF	0	4A,8M	679.4784	33.2268	712.705	8	0.49761631
BPF	0	3A,4M	358.6134	34.5047	393.118	8	0.40932128
DWT	0	3A,4M	358.614	19.1693	377.783	10	0.57481188
FIR	0	4A,4M	377.4884	28.1149	405.603	9	0.34422066

Table 3: Base case data

Benchmark	PROPOSED ANALYSIS				RELATED WORK			
	CONF.	AREA (um^2)	LATENCY (ns)	COST	CONF.	AREA (um^2)	LATENCY (ns)	COST
IIR	3A,3M	332.268	8	0.50893418	3A,3M	295.7972	5	0.390663281
ARF	8A,8M	843.548	11	0.6086040	8A,8M	799.016	8	0.545458479
BPF	5A,2M, 1ALU	442.663	13	0.5052612	3A,4M	397.05	8	0.412526278
DWT	5A,1M	258.147	15	0.4900046	4A,4M	400.2944	10	0.601970046
FIR	4A,4ALU	511.083	8	0.35022486	4A,4M	399.508	9	0.342429132

Table 4: Comparison with related work

10

Conclusion

Through our work we tried to bring new innovation in this field. This kind of work has never been proposed in the literature before. We brought the obfuscation and reliability together at very less overhead in terms of cost and hence resulted in the better strength of obfuscation.

References

[1] 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.*

[2] Anirban Sengupta, Reza Sedaghat *Swarm Intelligence Driven Design Space Exploration of Optimal k-Cycle Transient Fault Secured Datapath during High Level Synthesis Based on User Power-Delay Budget.*Elsevier Journal on Microelectronics Reliability, Volume 55, Issue 6, May 2015, pp. 990-1004, March 2015.

[3] Anirban Sengupta , Dipanjan Roy *Protecting IP core during architectural synthesis using HLT-based obfuscation.* Electronics Letters (Volume: 53, Issue: 13, 6 22 2017)