Design-For-Diversity for Improved Fault-Tolerance of TMR Systems on FPGAs

Rizwan A. Ashraf, Ouns Mouri, Rami Jadaa and Ronald F. DeMara
Department of Electrical Engineering and Computer Science
University of Central Florida
Orlando, FL 32816-2450, United States
rizwan.ashraf, omouri, rami.jadaa@knights.ucf.edu

***Abstract*—This paper investigates the ability to provide improved Reliability of TMR systems at comparable area and time cost using design diversity. Namely, we evaluate multiple implementations of the same functional design using a repository of methods: *Templates, Case-Based, Inverted-Output,* and *NAND/NOR-Based* methods. The design methods are tested on multiple benchmark circuits in different TMR setups for each of which design diversity and fault tolerance are examined. The results show that extensive design diversity can be achieved at design-time using one or a combination of these methods, and verifies the increased fault-tolerance of TMR-based systems with diverse designs in multiple failure modes at run-time. Moreover, results indicate that improved system fault-tolerance can be achieved using designs from different classes of design techniques, rather than using variations of the same design method without incurring a run-time expense.**

***Keywords*-TMR, FPGAs, fault-tolerance, reliability, design diversity**

I. Introduction

Electronic systems, especially those based on Field Programmable Gate Arrays (FPGAs), are prone to faults due to many factors, like operation in harsh environments as encountered in space or nuclear applications, where these devices may be affected by radiation effects and/or the aging process [10]. Many methods have been proposed to build fault-tolerant systems that can sustain single/multiple faults, and redundancy is one of the simplest and widely used methods [4]. Redundancy can take many forms, like cold/hot spares, or N-Modular-Redundancy (NMR). In the latter case, an N number of functionally-identical modules are operated on the same set of inputs simultaneously and a majority vote is used between the multiple outputs to produce the final output. For example, when $N = 2$, we have a concurrent redundant system that can detect faults instantly, but fails to isolate the faulty module. Whereas, in a Triple Modular Redundancy system (TMR) as shown in Figure 1, three functionally-identical modules can detect fault in a single module and mask it instantly through bitwise, or word-wise voting.

The modules used in TMR systems must be functionally-identical, i.e. they should have matching input/output responses. However, this does not impose exact physical implementation. This raises the concept of design diversity in redundant systems, in which the same functionality can be implemented using physically different, yet functionally identical designs. Granted, the meaning of "physically different" differs when referring to FPGAs than when referring to Application Specific Integrated Circuits (ASICs). In FPGAs, two modules are said to be physically different if most LUTs in the same relative location on both modules do not implement the same logical function. TMR systems based on a single-design have less immunity towards Common-Mode Failures (CMFs) that affect more than one module at the same time in the same manner, generally due to a common cause [1]. This may be due to a design oversight, power disturbance, or especially in sub-90nm technology, due to the aging phenomenon where components might be affected uniformly in a region, causing multiple failures in the same manner [10]. For example, Hot Carrier Effect (HCE), Time Dependent Dielectric Breakdown (TDDB) and Electro-migration (EM) can cause permanent faults. This can manifest Common-Mode Failure if identical datapaths are used in the TMR arrangement. CMFs are quite common in redundant systems using the same designs as shown in [7].

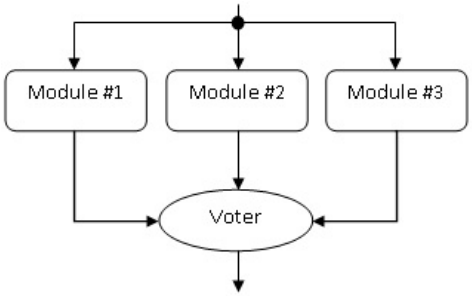

Figure 1. TMR System

Design diversity provides a solution to CMF. Techniques need to be investigated to generate diverse designs, which can be used in redundant systems as well as in other applications. For example, using a diverse population of individuals could achieve better performance in evolutionary-algorithm-based repair, as the population will offer many solutions, rather than creating new ones gradually through genetic operators [5]. Moreover, TMR systems implemented on FPGA can benefit from diversity by loading different designs generated online without the requirement of evolu-

978-0-7695-4551-6/11 $26.00 © 2011 IEEE
DOI 10.1109/ReConFig.2011.26

Table I
DESIGN DIVERSITY RELATED WORK

	Vigander [12]	**Keymeulen** [5]	**Sharma** [9]	**McClusky** [7]
Use of Diversity	TMR	Self-Repair	Fault-Isolation	Measuring reliability of redundant systems
Diversity Method Used	GA-based	GA-based	Place&Route	Not-Specified

tionary algorithms, and hence might offer autonomous repair by "jiggling" the modules with diverse but functionally-equivalent design configurations. The diversity techniques introduced and the concept of Diversity-TMR for improved reliability is also applicable to ASICs, but the ability for autonomous reconfiguration of ASICs will likely be limited and will certainly be application dependent.

This paper studies the synthesis of distinct designs using the Template-based (TB) method so that they can be used in redundancy-based fault-tolerant applications. Further, the Case-based (CB), Inverted-output (IO) and NAND/NOR-based design techniques are studied. The paper explicitly highlights how diversity can benefit a TMR system in different stuck-at faults in different failure modes (CMF and Random faults). It studies the possibility of generating a better TMR system using diverse designs generated from different classes of design techniques.

II. RELATED WORK

Some researchers made use of diversity in redundant systems, either implicitly, like in [3], [12], or studied diversity explicitly as in [2], [7], [8]. In [12], evolutionary algorithm based repair techniques are used to partially repair the faulty modules and these modules are used in a TMR system to mask the faults from individual modules as diverse modules fail in a different manner. Thus evolutionary algorithms can be potentially looked upon as an another design technique to generate diverse designs that can be used in a TMR setup. This is noted in [5], where evolutionary methods are used to create fault-tolerant designs of an analog multiplier and an XNOR function on FPTA using different techniques such as population-based, and fitness-based techniques.

In [9], diverse designs are generated through place & route technique to run Combinatorial Group Testing (CGT) methods for fault isolation. Place & Route is another design technique in which the same module can be considered physically different by different positioning and routing of the blocks. The authors of [2] explore the concept of design diversity based redundancy applied to mixed-signal circuit blocks.

All previous works provide an implicit example of the importance and usage of design diversity in different fields (TMR, GA-based refurbishment, Fault Isolation). Table I summarizes the work of each from the design diversity perspective. Our work uses the design diversity metric developed by McCluskey *et. al.* [7] to measure the improvement of the reliability of the TMR system when redundant modules are designed using radically different techniques belonging to different classes of design paradigms.

A mathematical model is presented in [8] to quantify design diversity among designs in a CED configuration, with emphasis on the importance of data integrity of the system, which is defined as its ability to either produce correct outputs or generate an error signal when incorrect outputs are produced. Let $d_{i,j}$ be the diversity with respect to a fault pair (f_i, f_j) where f_i and f_j are faults present in the diverse modules M1 and M2 respectively. Thus $d_{i,j}$ denotes the probability that the designs do not produce identical error patterns, in response to a given input sequence. This prompts the notion of *joint detectability*, $k_{i,j}$ which is defined as the number of input patterns that produce the same erroneous output pattern in both implementations of M1 and M2. If we assume that all input patterns are equally likely, then $d_{i,j}$ can be specified as:

$$d_{i,j} = 1 - \frac{k_{i,j}}{2^n} \tag{1}$$

where n is the number of inputs. Assuming all fault pairs are equally probable and there are m fault pairs (f_i, f_j), the design diversity metric, D for the design pair is:

$$D = \frac{1}{m} \sum_{i,j} d_{i,j} \tag{2}$$

The CMF is represented by $i = j$ whereas the random fault case can be expressed by $i \neq j$.

For example, for two designs M1 and M2, with the following responses given in Table II, due to the injection of fault pair (f_i, f_j), the value of $k_{i,j} = 2$, and hence $d_{i,j} = 1 - \frac{2}{4} = 0.5$. So if we injected this single fault pair, then $D = 0.5$ for these designs.

Table II
M1 AND M2 RESPONSES TO AN INJECTED FAULT PAIR

Inputs	**Fault-free Outputs**	**M1 Outputs**	**M2 Outputs**
00	01	00	10
01	10	10	10
10	00	10	10
11	11	10	10

III. DESIGN-FOR-DIVERSITY TECHNIQUES

A. Template-Based Method

To understand the Template-Based method as proposed in [3], consider the system in Figure 2. The system func-

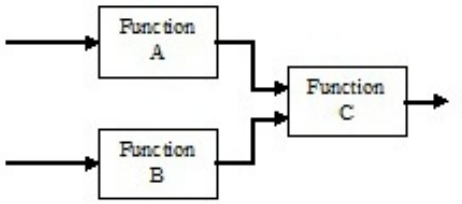

Figure 2. System Block Diagram

tion is implemented through blocks (templates) whose inputs/outputs are connected accordingly to perform the required functionality. Therefore, Figure 2 represents a general block diagram of the operation itself, implemented internally through blocks A, B, and C. The number of different designs we can obtain for the whole system by replacing each block with a possible design option (DO) is a function of the number of DOs available for each block. Therefore for any system represented by a block diagram of functional blocks (templates), the number of possible design options to achieve the same function is calculated by the following equation:

$$DO_{sys} = \prod_{i=A}^{C} DO_i \tag{3}$$

Where DO_i represents the design options available for each block i. For example, in the above system, if DO_A = 2, DO_B = 3 and DO_C=1, then there are 6 different design options for the full system as: $DO_{sys} = DO_A \times DO_B \times DO_C = 6$.

In short, the template-based technique can be applied to any system that is described by internal blocks each implementing a sub-functionality by replacing any block with a different design template. Therefore, by generating multiple designs for each block during design-time and storing the associated partial bit-files, the system can automatically use combinations of them online to create multiple diverse-designs at runtime.

B. NAND/NOR-Based Method

Combinational circuits can be implemented in multiple ways. NAND and NOR functions, which are known as the universal gates can implement any specified digital circuit. So it is possible to convert any given digital circuit into a NAND or NOR-only implementation. This makes way for design diversity as, given any digital circuit, it is true that there is always an alternative implementation which is in terms of NAND gates only (given the original implementation is not already in terms of NAND gates). This paper explores the applicability of this classical design technique for implementation in FPGA devices through the manipulation of the User Constraint File (UCF) with the goal of achieving design diversity. A circuit implemented through this technique tends to have a higher component count which might achieve better reliability at the expense of increased resource usage. Results of this approach for an FPGA implementation are presented in the results section.

C. Case-Based & Inverted-Output Methods

Case-Based Synthesis is an informal and very simple design technique that involves describing the function to be implemented in the form of a truth table. This truth table is then translated into a HDL case statement which is fed to the synthesizer for logic extraction. This process is straight-forward for small combinational circuits. However, the length of the case statement grows exponentially with the number of input bits. This can be overcome by automating the case statement generation process given a functional description of the circuit, or by dividing the complex system into smaller, more manageable sub-circuits.

Once the case statement described above is generated, an Inverted-Output description of the system is easily produced. This is done by inverting the outputs associated with each input case.

The synthesis of logic functions in their true and complemented forms during duplication was first proposed in [6], and, depending on the synthesizer used and the optimization parameters set, synthesis of the Inverted-Output description of a particular function will result in a different implementation from that obtained from the synthesis of a true case-based description.

A challenge arises when dealing with sequential circuits since the output does not depend solely on the inputs. In [11], it is shown that given the specification of a sequential logic circuit, and an encoding of its internal states, the problem of synthesizing the sequential circuit can be mapped to a combinational logic synthesis problem.

IV. Experimental Setup

A. Simulation Objectives, Tools and Workflow

A total of 6 experiments were performed on many TMR systems with different stuck-at failure-modes. The objectives of the experiments are to:

1) Find the diversity values among designs generated using the proposed design techniques.
2) Evaluate the sensitivity of different designs to the type of stuck-at fault (Zero or One) injected.
3) Compare the performance of many TMR systems in two fault modes: Common-Mode Fault (CMF) and Random Single Fault (RSF). The purpose is to find the best TMR system to provide the highest reliability in the presence of CMF and RSF faults.
4) Study the effect of using different design techniques vs. uniform design techniques in a TMR system in order to achieve an improved overall fault-tolerance.

All experiments were carried out using Xilinx ISE Design Suite 12.2 equipped with the ISim FPGA simulator. The target configuration was that of a Xilinx Virtex-4 FPGA device. The experiments involved multiple implementations of two benchmark circuits i.e. a three-bit Multiplier (combinational circuit) and a 8-state Finite State Machine as defined in dk17 benchmark circuit (sequential circuit) [13] as a case study. All implementations were generated using the *XST* synthesizer provided with the Design Suite, and Stuck-At faults at the inputs of the LUTs were injected directly into the post-Place&Route model file before simulation. An implementation, referred to as the base design (BASE), was generated using behavioral Verilog HDL code for both the designs. The synthesizer had complete control over the implementation process in this case.

In the following, we define Reliability as the probability that the output obtained from a certain design (or the TMR system) is not erroneous. For example, if the TMR system provides A erroneous outputs out of 2^n outputs (where n is the number of input bits) in the presence of a fault sequence (f_i, f_j, f_k), then the reliability R relative to the fault (f_i, f_j, f_k) is defined as:

$$R_{(f_i,f_j,f_k)} = 1 - \frac{A}{2^n} \tag{4}$$

If the number of fault sequences recorded is B, then the reliability can be expressed as:

$$R = 1 - \frac{A}{B \times 2^n} \tag{5}$$

B. Experiment Descriptions

1) Experiment 1: The Diversity Values among various designs are calculated in these experiments. Two instances of each design method are used to calculate the intra/inter-design diversity values to get an insight of the diversity of the designs generated by the same class of techniques.

2) Experiment 2: TMR with Diverse Designs (BASE, Da, Db) and a single CMF per module is used in these experiments. Da and Db are generated using the same design technique, though they are physically distinct. CMF study is conducted in line with McCluskey's *et. al.* [7] work and it assumes that the LUTs in the two designs are affected by a Stuck-At fault of the same type and at the same input pin location.

3) Experiment 3: TMR with Diverse Designs (BASE, Da, Db) and a single RSF per module is used for these experiments. This experiment utilizes the same setup as Experiment 2, but injects one random stuck-at fault at a random location in each module. In conjunction with experiment 2, this will indicate the benefit (if any) of using design diversity over replicated design in a TMR system.

4) Experiment 4: TMR with replicated design and a single random fault per module is used for these experiments. In this experiment, the TMR system is composed of three identical datapath modules. Random stuck-at faults are injected at random location in each module. The hypothesis of this experiment is that the modules will produce different outputs since the fault locations are not similar, and therefore the system might not behave significantly worse than the implementation of diverse modules.

5) Experiment 5: TMR with Diverse Designs (Inverted-Output, Template-Based, and NAND-Based) and a single CMF per module is used for these experiments. Single CMF were injected in this case per module. The diverse designs are expected to fail in different ways, thus most of the time correct output should be produced.

6) Experiment 6: TMR with Diverse Designs (Inverted-Output, Template-Based, and NAND-Based) and a single RSF per module is used for these experiments. This experiment uses the same setup as in Experiment 5, but faults are injected randomly at random locations in each module. The performance in this case is expected to be better than CMF case. Combined with the results of experiments 3 and 4, the presence of any advantage for using diverse designs for TMR should be confirmed (or refuted) in the presence of random faults.

Table III
DIVERSITY VALUE OBTAINED THROUGH COMPARISON WITH THE BASE DESIGN (3X3 MULTIPLIER)

	D1	**D2**	**CB**	**IO**	**NAND**	**NOR**
CMF	1	0.996	1	0.987	0.965	0.958
RSF	0.971	1	0.984	1	0.992	1
Average Diversity Value	0.986	0.998	0.992	0.994	0.979	0.979

V. RESULTS & ANALYSIS

A. Diversity Metric Value

Table III summarizes the diversity values obtained through CMF and RSF when compared to the BASE design for a 3x3 Multiplier. D1 & D2 are designs obtained through Template-based method, CB and IO are obtained through Case-based and Inverted Output method, and NAND & NOR are obtained through NAND and NOR-based method, respectively.

The diversity value of the different designs was calculated through the injection of $m_1 = 64$ CMF faults and $m_2 = 8$ RSF with a total of $(m_{total} = m_1 + m_2)$ 72 fault pairs. Table III shows high values of design diversity for the different designs when compared to the same BASE design, indicating the success of the proposed diversity techniques in creating diversity.

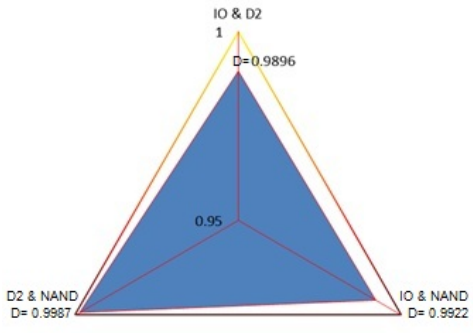

Figure 3. Inter-Design Diversities in CMF (3x3 Multiplier)

Intra-design diversity values are calculated among selective members of the same design class to get an insight of how diverse the designs generated by the same technique can be. Table IV shows the intra-design diversity values for the 3x3 Multiplier.

Table IV
INTRA-DESIGN DIVERSITY VALUES (3X3 MULTIPLIER)

	D1 & D2	CB & IO	NAND & NOR
CMF	0.9956	0.9868	0.9731
RSF	0.9961	0.9844	0.9883
Average Diversity Value	0.9959	0.9856	0.9807

Results indicate that all design techniques provide very diverse designs in response to fault pairs injected in different failure modes. D1 & D2 designs represent the extreme conditions template-based designs can generate, in which D2 is obtained by replacing all the Half and Full adders by different design templates than the ones used in D1. Therefore, their diversity values were high, and this sets the upper limit for the highest diversity value the Template-based technique can provide for the 3x3 Multiplier design.

To further comprehend the inter-design diversity of the designs generated from different classes, a CMF test was carried out. Results for the 3x3 Multiplier are reported in Figure 3.

B. TMR performance in different design techniques

To understand the effect of the generated designs on the overall TMR system reliability, the following TMR system topologies are utilized with a *bitwise-voter* output:

- *TMR_BASE:* Implemented by three modules using the same (BASE) design.
- *TMR_TB:* One module uses the BASE design and the other two modules use different designs (D1 and D2) obtained using *template-based* method.
- *TMR_DD:* Each module is implemented using different design technique, in which $module_1$ is implemented using *template-based*, $module_2$ implemented using *inverted output*, and $module_3$ implemented using *NAND* functions only.

Table V
TMR RELIABILITY (3X3 MULTIPLIER)

	TMR_BASE	TMR_TB	TMR_DD
CMF	0.9473	0.989	0.9674
RSF	0.9746	0.9316	0.9844

Table VI
TMR RELIABILITY (DK17 BENCHMARK)

	TMR_BASE	TMR_TB	TMR_DD
CMF	0.8307	0.9479	0.9922
RSF	0.8724	0.8880	0.9167

All TMR arrangements mentioned above are evaluated using actual physical (post-P&R) designs synthesized with the aforementioned Xilinx toolset on the Virtex-4 FPGA device. 64 test runs are made in CMF for TMR_BASE and TMR_TB systems, while 12 test runs are made for TMR_DD. All RSF tests were composed of 8 runs per TMR system. In all test runs, an exhaustive evaluation is done for all possible input patterns and erroneous outputs are recorded and counted. Note, the dk17 benchmark has 2 inputs and 8 possible states, thus it is evaluated for all 32 state transitions. Using equation 5, the reliability of each TMR system is then calculated and reported in Tables V and VI for 3x3 Multiplier and dk17 benchmark ($n = 5$) respectively.

Results indicate that TMR systems based on diversely-designed modules provide higher reliability in Common-Mode Failures. This is consistent with McCluskey's *et. al.* [7] conclusion about Common-Mode Failures in diverse design redundant systems, thus closing the design hypothesis with the objectives of this research.

Figures 4 and 5 show reliability values of different TMR systems in different failure modes (CMF and RSF) for 3x3 Multiplier and dk17 benchmark respectively. In these figures, TMR_IO is implemented by replicating the inverted output design. Similarly, TMR_NAND uses the NAND design while TMR_CB_IO uses the Case-Based and Inverted-Output designs besides the BASE design. The first three sets of bars represent a TMR system utilizing the same design replicated, while the last three represent a TMR system utilizing diverse designs from different design techniques. Results clearly show that a TMR system using diverse designs is superior in reliability to replicated design, irrelevant of the design technique used. This is consistent for both benchmarks evaluated in this work.

VI. CONCLUSION

The Template-based technique is studied and evaluated to automatically generate different designs during run-

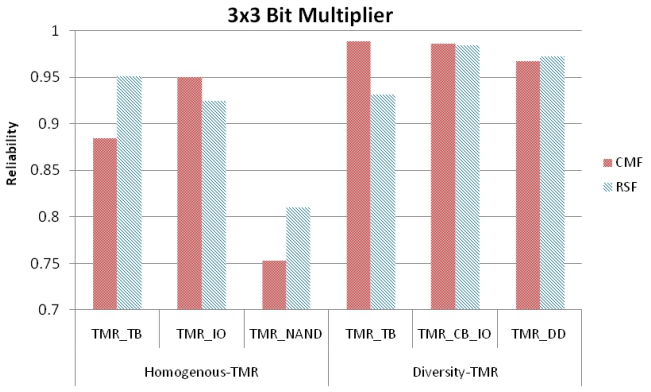

Figure 4. Reliability for different TMR systems (3x3 Multiplier)

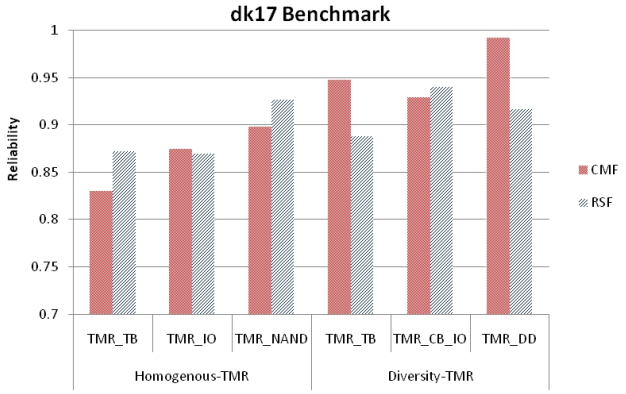

Figure 5. Reliability for different TMR systems (dk17 Benchmark)

time. Additionally, the Case-based, the Inverted-output and NAND/NOR-based design techniques are evaluated. The generated designs showed a high degree of design diversity, even though additional design-time effort is required. To measure diversity, McCluskey's diversity metric was used by injecting fault pairs and recording exact fault responses of the design pairs. In some cases, the metric has a counterintuitive behavior as same designs show high design diversities when fault pairs were injected at different locations, which hints that this metric might be valid for CMFs only.

An improved design diversity metric might help provide a more accurate result of how diverse the generated designs are, and hence it may shed more light on the performance of the system utilizing them.

Several TMR based systems were implemented using different topologies from different design techniques, and their reliability was studied accordingly. Results indicate that diverse-design-based TMR systems show higher reliability to CMF exposure, and using different design techniques offers improved reliability for randomly injected faults at minimal additional cost and effort.

REFERENCES

[1] A. Avizienis and J. Kelly. Fault tolerance by design diversity: Concepts and experiments. *Computer*, 17(8):67 –80, aug. 1984.

[2] G. d. M. Borges, L. F. Gonçalves, T. R. Balen, and M. S. Lubaszewski. Evaluating the effectiveness of a mixed-signal tmr scheme based on design diversity. In *Proceedings of the 23rd symposium on Integrated circuits and system design*, SBCCI '10, pages 134–139, New York, NY, USA, 2010. ACM.

[3] R. F. DeMara, K. Zhang, and C. A. Sharma. Autonomic fault-handling and refurbishment using throughput-driven assessment. *Applied Soft Computing*, 11(2):1588 – 1599, 2011. The Impact of Soft Computing for the Progress of Artificial Intelligence.

[4] G. W. Greenwood. Attaining fault tolerance through self-adaption: the strengths and weaknesses of evolvable hardware approaches. In *Proceedings of the 2008 IEEE world conference on Computational intelligence: research frontiers*, WCCI'08, pages 368–387, Berlin, Heidelberg, 2008. Springer-Verlag.

[5] D. Keymeulen, R. Zebulum, Y. Jin, and A. Stoica. Fault-tolerant evolvable hardware using field-programmable transistor arrays. *Reliability, IEEE Transactions on*, 49(3):305 –316, sep 2000.

[6] S. Mitra and E. McCluskey. Combinational logic synthesis for diversity in duplex systems. In *Test Conference, 2000. Proceedings. International*, pages 179 –188, 2000.

[7] S. Mitra, N. Saxena, and E. McCluskey. A design diversity metric and reliability analysis for redundant systems. In *Test Conference, 1999. Proceedings. International*, pages 662 – 671, 1999.

[8] S. Mitra, N. Saxena, and E. McCluskey. Efficient design diversity estimation for combinational circuits. *Computers, IEEE Transactions on*, 53(11):1483 –1492, nov. 2004.

[9] C. A. Sharma. *Sustainable Fault-Handling of Reconfigurable Logic Using Throughput-Driven Assessment*. PhD thesis, University of Central Florida, FL, USA, aug 2008.

[10] S. Srinivasan, R. Krishnan, P. Mangalagiri, Y. Xie, V. Narayanan, M. J. Irwin, and K. Sarpatwari. Toward increasing fpga lifetime. *IEEE Transactions on Dependable and Secure Computing*, 5:115–127, 2008.

[11] Y. Tohma and S. Aoyagi. Failure-tolerant sequential machines with past information. *Computers, IEEE Transactions on*, C-20(4):392 – 396, april 1971.

[12] S. Vigander. *Evolutionary Fault repair of Electronics in Space Applications*. PhD thesis, Norwegian University of Science and Technology, Trondheim, Norway, feb 2001.

[13] S. Yang. Logic synthesis and optimization benchmarks version 3. Technical report, Microelectronics Center of North Carolina, 1991.