TRIPLE MODULAR REDUNDANCY WITH STANDBY (TMRSB) SUPPORTING DYNAMIC RESOURCE RECONFIGURATION

Kening Zhang
Department of Electrical and Computer Engineering
University of Central Florida
Orlando, Fl 32826
kzhang@mail.ucf.edu

Guy Bedette
Applied Technology Directorate NASA KSC
MS KT-C
Kennedy Space Center, FL 32899
(321) 861-2291
Guy.Bedette@nasa.gov

Ronald F. DeMara
Department of Electrical and Computer Engineering
University of Central Florida
Orlando, Fl 32826
407-823-5916
DeMara@mail.ucf.edu

Abstract - ***A fault tolerance model called Triple Modular Redundancy with Standby (TMRSB) is developed which combines the two popular fault tolerance techniques of Triple Modular Redundancy (TMR) and Standby (SB) fault tolerance. In TMRSB systems, each module of a TMR arrangement has access to several independent standby configurations. When a fault is detected in a module's active configuration, the physical resources within that module are re-mapped to restore the desired fault-free functionality by reconfiguring the resource pool to one of the standby configurations. A mathematic model for TMRSB systems is developed for Field Programmable Gate Array (FPGA) logic devices. Simulation of the model was also performed using the BlockSim reliability software tool which takes into account the reconfiguration time overheads and an imperfect switching mechanism. With component time-to-failure following an exponential distribution throughout long mission duration, the range of operation over which TMRSB is superior to a Standby System and a TMR system is shown.***

1. INTRODUCTION

Despite continued improvements in reliability at the component level, system-level fault tolerance strategies retain an essential role for mission-critical applications. Fault tolerance strategies utilizing redundant components include a variety of architectures that can provide higher system reliability. Many previous fault tolerance approaches such as *Triple Modular Redundancy* (TMR), Simplex/TMR and Standby systems have been extensively covered in literature [2] [11]. Recently reconfigurable logic devices have become available and achieved rapid popularity, especially in embedded systems including mission critical applications. In this paper, we develop and analyze an approach to exploit these new hardware capabilities to realize a hybrid model of TMR and SB strategies using an amorphous resource pool by employing dynamic reconfiguration. Dynamic reconfiguration provides a large number of "virtual" standby modules without requiring distinct physical components to realize each standby instance.

The conventional *N modular Redundancy* (NMR) [8] system provides a powerful approach of improving reliability and fault tolerance capacity of digital systems. In NMR models, there are *N* functional modules, *N=2m-1* for some integer *m>1,* which are implemented identically. Each module performs concurrent computation of identical tasks. Their outputs are provided to a majority voter to obtain a fault-free result whenever at least *m* modules and the voter are functioning correctly. Among NMR approaches, TMR [2] has been one of the most popular fault-tolerance schemes using spatial redundancy for practical system implementations. In Figure 1, the three functionally identical modules M_1, M_2, and M_3 are deployed in parallel and the outputs converge at the majority voter to obtain the validated output for the system.

Another fault tolerance strategy is the *Standby (SB)* model. The SB model represents the case in which a primary component (or system) has one or more identical backup components in an *off-line* state. When the original active component fails, a switch mechanism selects one of the *standby* backup components and configures it as the active component. Ideally, this allows the system to continue to operate correctly with a temporary execution degradation affected only by switching overhead.

When considering the range of possible backup component states, three types of standby systems can be identified. A *Hot Standby System* maintains the primary and backup components running simultaneously with the backup tracking the primary system in real time. A *Cold Standby System* is a method in which the secondary component is only called upon when the primary component fails. A third option is the *Warm* standby system, which periodically mirrors the primary component which means that there are times when both components do not contain the exact same data. As shown in Figure 2, the standby configuration can be in *Hot, Cold,* and *Warm* states depending on the specific system design.

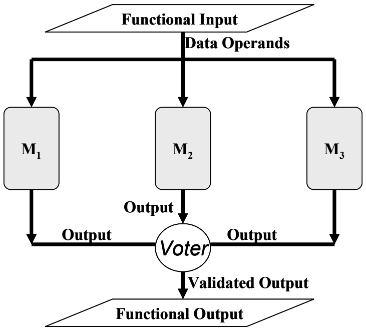

Figure1. TMR System

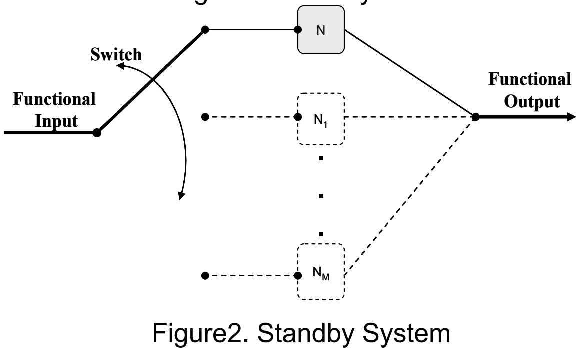

Figure2. Standby System

1.1. Embedded Device Properties Influencing Redundancy Strategies

SRAM-based reprogrammable device known as a *Field Programmable Gate Arrays* (FPGAs) are large multi-million gate-equivalent devices that employ these technologies extensively. The FPGA configuration is stored in bitstream format in the PROM and loaded into or read back from the FPGA chip through *Configuration Logic Interface*. The different connections on the FPGA chip integrate the Configuration Logic Blocks (CLBs) or Look Up Tables (LUTs) to implement computation logic tasks [4].

Environmental challenges to reliability in space applications can be modeled as having a uniform failure rate exposure despite status and locations of device activity in the system. Therefore, the impact of device wear-out (active components vs. cold spares) is small relative to radiation exposures, which makes ambiguous the active vs. standby role in terms of reliability in the various standby models.

Radiation-induced Single Event Upsets (SEUs) can produce soft failures in both the configuration memory itself and in the mapped circuit on the throughput data-path. The result of a SEU that makes the device totally or partially lose functionality is generally defined as Single Event Functional Interrupt (SEFI) [1].

FPGAs are a highly suitable platform for integrating reliability models like *NMR* and *SB*. Their unlimited reprogrammability makes standby component switching feasible with low delay and overhead. Thus, reprogrammability recovery mechanisms can be realized using a resource pool to extend mission lifetime compared to a non-repair system.

The emerging field of autonomous repair has focused on systems for deeper space exploration mission demanding sustainability, availability, and serviceability [12] [5] [13]. Additionally many techniques have been developed to generate the pre-complied alternative fault tolerance configurations which are stored in memory in order to reconfigure when a fault occurs [9].

The proposed approach is a hybrid system architecture to handle a wide range of transient faults through automatic FPGA reconfiguration and also permanent failures though automatic selection from a diverse set of standby components. The standby configurations implement identical functionality but may use a different arrangement of physical resources, and a dynamic update of these alternative configurations.

This paper is organized as follows. Section 2 summarizes redundancy approaches and recent concerns about fault tolerance techniques for SRAM-based FPGAs. Section 3 describes the theoretical analysis about the TMRSB model. The simulation approach and result are presented and reported in Section 4. Conclusions and future work are presented in Section 5.

2. Related Work

2.1. Standby and NMR Redundancy Approaches

The TMR approach, first proposed by Von Neumann [2], is shown in Figure 1. It was widely used in software fault tolerance [11] and reliable hardware [3] applications. The primary drawback of the TMR approach is resource overhead. The TMR design triples the area and power consumption of physical resources over a simplex design. Duplex systems with a hot standby component based on a process pair [6] paradigm for fault tolerance are widely implemented in Network Access Devices (NAD) [7] and other uninterruptible operational systems.

However, much of the superiority of TMR and Standby type systems hinges upon some critical components. The reliability (or lack or reliability) of the majority voter in TMR systems and the Standby system switch mechanism may be detrimental to the overall system reliability. There are other issues to consider including: the reliability of memory which stores the standby configurations, the capability of sensing improper operation to trigger a switch, or how the majority voter and the switch operation must maintain data consistency between the primary and backup components.

2.2. Redundancy Applications involving FPGAs

Several previous works on TMR systems for FPGAs are introduced in [10] [3] and [14]. In [10], the TMR system with voting technique is combined with bitstream scrubbing implemented in a Virtex FPGA device in order to mitigate *Single Event Upset* (SEU) effects. The voting mechanism identifies the faulty configuration based on single failure assumption and reconfigures (scrubs) the device with an alternative bitstream. However, the reconfiguration has to take place off-line and can only deal with a transient faults which can be restrictive for use during deep space missions.

A TMR application for the Virtex series of Xilinx FPGA is described in [3]. The Majority voter is implemented with 3-State buffers based on the Virtex bus structures. Different types of data structures such as Throughput Logic, State-Machine Logic and I/O Logic are illustrated in terms of a TMR technique. Some special features provided by the Virtex architecture are also mentioned. In the next section the reconfiguration properties of FPGA devices are employed to realize a combination of the TMR and Standby models.

3. SYSTEM RELIABILITY ANALYSIS

3.1. Standby System

Consider the SB system configuration depicted in Figure 2. It contains *m+1* identical components of which exactly one is active at any time and the remaining *m* components act as switchable spares. Up to *m* of these spares may provide feasible alternative standby configurations in order to extend the mission lifetime.

A Standby system with only one active component C_i where *i*=0, 1, 2…*m* designates the feasible set of one active and *m* standby components, will be considered first. The components are modeled with an exponential failure rate λ. Assuming that the de-energized components do not operate until a fault is detected in the active component, or otherwise dictated by the reloading schedule, the lifetime which is time to failure, *Z*, of such system can be characterized in term of the lifetime, X_i, of each individual configuration C_i is $Z = \sum_{i=0}^{m} X_i$.

Initially, assume the switch mechanism is completely reliable and all of the standby configurations are fault-free. Let X_i and $X_{j \neq i}$ be random variables denoting the independent failure of each component. Assuming an exponential distribution given by the parameter *λ*, then the pdf is given by $f_{Xi}(t) = \lambda e^{-\lambda t},\ t > 0$. Since *Z* = X_i + X_j, the density of the sum of two non-negative independent random variables is given by the convolution of the individual densities [15], we have:

$$\begin{aligned} f_Z(z) &= \int_0^Z \lambda e^{-\lambda t} \lambda e^{-\lambda(z-t)} dt \\ &= \lambda^2 e^{-\lambda t} \int_0^z dt \qquad (1) \\ &= \lambda^2 z e^{-\lambda t} \quad , \qquad z > 0 \end{aligned}$$

Thus Z has a two-stage Erlang distribution [15] for the *m*=1 case and *m*-stage Erlang distribution [15], in general. Thus, for the *m*=1 case, the failure distribution function of Z is given by:

$$\begin{aligned} F(t) &= 1 - \sum_{k=0}^{m} \frac{(\lambda t)^k}{k!} e^{-\lambda t} \\ &= 1 - \left(1 - \lambda t e^{-\lambda t}\right) \qquad (2) \\ &= \lambda t e^{-\lambda t} \ (t \geq 0, \lambda > 0, m = 1) \end{aligned}$$

Then the *m*>1 reliability function is obtained by

$$\begin{aligned} R_{Standby}(t) &= 1 - F(t) \\ &= \sum_{k=0}^{m} \frac{(\lambda t)^k}{k!} e^{-\lambda t} \qquad (3) \\ &= e^{-\lambda t} + \sum_{k=1}^{m} \frac{(\lambda t)^k}{k!} e^{-\lambda t}, \ t \geq 0, \lambda > 0, m = 1,2,\ldots \end{aligned}$$

In Equation (3), $e^{-\lambda t}$ term represents the reliability of the initially-selected active component. The subsequent summation term in Equation (3) represents the probability that each standby component will provide a viable alternative. For example, suppose the initial active component fails and one of the standby components becomes energized to maintain the system availability. In this case, the summation of the reliabilities of all such replacements plus the initial component reliability determines the system reliability.

However, the specific characteristics of the space application mentioned in Section 2 eliminate the variety of the different standby approaches. Because the standby configurations are stored in non-volatile memory and the circuits they describe are mapped into SRAM-based FPGA architecture, we need to assume the standby individual failure status is unknown until they are selected for operation.

A few simplifying assumptions are made since failure of the switch mechanism will cause the whole standby system to permanently cease recovery operations:

1. Faults in an active configuration will simultaneously disable that configuration and initiate a single configuration switch operation.
2. There is always at least one fault-free standby configuration available in the standby pool.

Wherever the standby configurations are stored and whatever state they are in, radiation may induce faults in their storage representation by mechanisms similar to faults that affect the active elements. Even with a perfect switching mechanism, a faulty standby configuration will generate an unexpected output. Faulty standby configurations will be detected when they are online and the switch mechanism is modeled as continually loading alternative backup configurations out of the standby pool until a fault-free one is running. When the fault-free configuration is loaded as active the one, the selection will be suspended until next fault occurs and impacts the current active selection.

Because of the unknown status of standby configurations the probability that the system fails after switching to a standby configuration with a faulty configuration follows the number of failures before the first success. It can be modeled as a geometric random variable with probability mass function of $p \ (1-p)^v$ in which *v* is the number of the failure selection trails (*v*<<*m*) and *p* is the probability of fault-impact for each configuration.

Assuming the survival rate *p* follows an exponential distribution and the selection process is a binomial distribution, the reliability for R_{TMRSB} is given by:

$$R_{TMRSB}(t) = e^{-\lambda t} + (1-p)^v \sum_{k=1}^{m} \frac{(\lambda t)^k}{k!} e^{-\lambda t}, \qquad (4)$$
$$t \geq 0, \lambda > 0, v < m, m = 1,2,\ldots$$

The number of the successful fault-free standby configurations determines the system feasibility and, according to the assumption 2 above, *v* will always far less than *m,* which will leave *m-v* number of standby configurations without any impact on the Equation (3). Figure 3 shows R_{TMRSB}(t) accordingly.

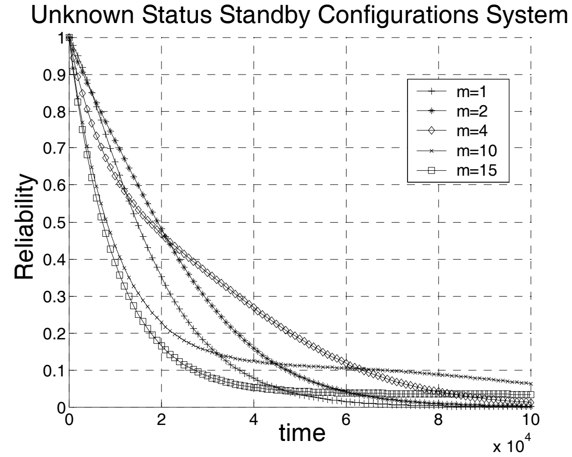

Figure3 . The Standby System with Unknown configuration Status configuration

According to Figure 3, the reliability of unknown configurations is not a linearly increased with the number of the standby configurations. That is because more configurations may bring more fault impact when more faults occur in the system. In Figure 3, we can see when *m=2* and *m=4* have more superior performance than higher *m* values. Comparing the case *m=2* to the case *m=4*, better performance occurs with *m=2* initially and as time goes on, the *m=4* case exhibits better performance because more standby resources can be exploited over a longer time mission. However, a larger value of m also causes overhead during the switching operations and status check which cause *m=10* and *m=15* performance to be worse than *m=2* and *m=4*. So in the later analysis, we select *p*=0.9, *v*=0.1 and *m*=4 as an optimized parameter set, and we also set λ=0.001.

3.2. *NMR* System

General treatments of *NMR* systems were developed starting in the 1950s [2]. Most of them assume a perfect voter in the system, and the reliability expression is based on binomial distribution given by:

$$R_{NMR} = \sum_{i=k}^{n} \binom{n}{i} p^{i} (1-p)^{n-i} \qquad (5)$$

If each component follows an exponential distribution $p = e^{-\lambda t}$, then

$$R_{NMR} = \sum_{i=k}^{n} \binom{n}{i} e^{-\lambda t i} \left(1-e^{-\lambda t}\right)^{n-i} \qquad (6)$$

Given R_{NMR} is equal to the system reliability R then the Reliability of TMR system is $R_{TMR} = 3e^{-2\lambda t} - 2e^{-3\lambda t}$.

3.3. Hybrid System

The TMRSD system in Figure 4 embeds the Standby system into the TMR framework in order to achieve a higher reliability and maintainability design. The system can be viewed as three functionally identical parallel subsystems with a majority voter, and each subsystem has *m-1* standby components. Equation 7 shows the mathematical model of the TMRSB system, the parameters are same in the equation (4) and (6). Components in this case are defined as functionally identical subsystems that utilize varied physical resources. To simplify the computation, in this paper, we only consider an identical number of standby components for TMR subsystems.

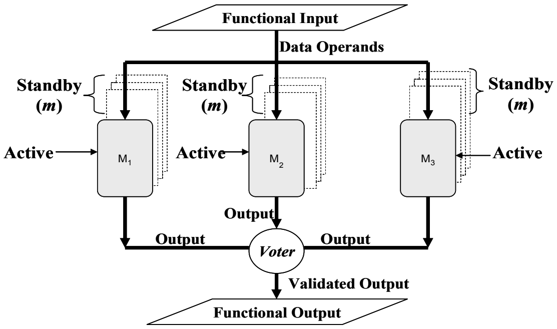

Figure4 . Comparison of Variant System Reliabilities

$$\begin{cases} R_{TMRSB} = \sum_{i=k}^{n} \binom{n}{i} R_{S\tan dby}{}^{i} \left(1-R_{S\tan dby}\right)^{n-i} \\ R_{S\tan dby}(t) = e^{-\lambda t} + (1-p)^{v} \sum_{k=1}^{m-1} \frac{(\lambda t)^{k}}{k!} e^{-\lambda t} \end{cases} \qquad (7)$$

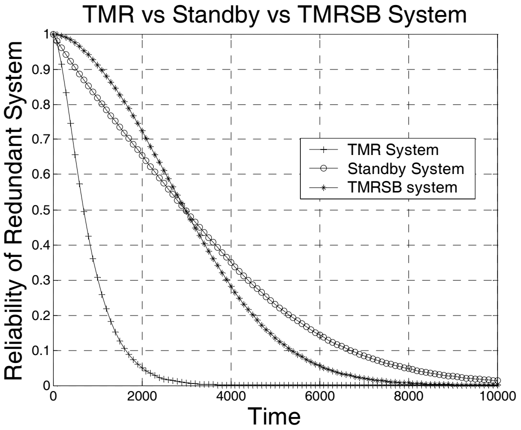

Figure5 . TMR vs. Standby vs. TMRSB system reliability comparison

The reliabilities of TMR, Standby, and TMRSB systems are presented in Figure 5. The TMRSB system improves the reliability only for the limited period time which can be utilized in short time mission. In this figure, we use previous optimal set parameters *p*=0.9, *v*=0.1, *m*=4 and λ=0.001.

However, because *x=λt*, when the λ is very small and the time *t* can be varied. This means the component reliability is essential factor of the system performance. Furthermore, this analysis shows that the system level reliability is based on the basically reliable components. In other words, a redundancy-based technique may not improve or worsen the system reliability based on unreliable components.

4. SIMULATION RESULTS

4.1. Case Study and Model Parameters

To substantiate the analytical results, BlockSim 6 developed by ReliaSoft was used to simulate TMRSB configurations. BlockSim 6 allows analysis of any process or product to estimate system reliability (including system reliabilities, mean times, failure rates, etc.), to calculate the optimum scenario to meet system reliability goals and to obtain maintainability, availability and throughput results through discrete event simulation. BlockSim's components can be defined with the reliability characteristics of each portion of the process or product. One can then configure these blocks into a Reliability Block Diagram (RBD) that represents the reliability-wise configuration of the system and analyze the diagram in order to determine the reliability in terms of the cumulative density function (cdf) of the entire system. During the simulation, the exponential distribution is used on the experiment. According to the above discussion, we assign the same distribution on the both active and standby configurations.

4.2. Standby System

Figure 6 shows simulation results for the standby system. Simulation results correspond well with the analysis given in Section 3. The TMRSB approach is also simulated in the BlockSim and the result listed below in Figure 7. The result shows that TMRSB improves the reliability compared to the single standby system and the higher component reliability, additional improvement can be achieved. Here we used *m=4* in the TMRSB system, other parameters as same as previous section.

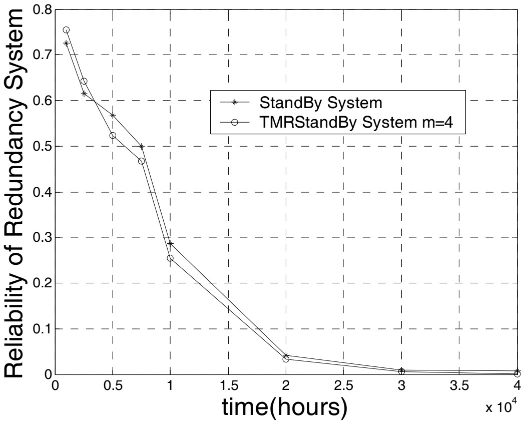

Figure6 . Simulation Result of Standby System and TMRSB system

4.3. TMRSB System

The simulation result also demonstrated the reliability of TMRSB is not improved linearly with the number of the standby components which actually agree with results that a different number of components may show variations in performance between different time periods. We plotted the result of *m=2* and *m=4* for TMRSB in Figure 7 which agrees with analysis as well.

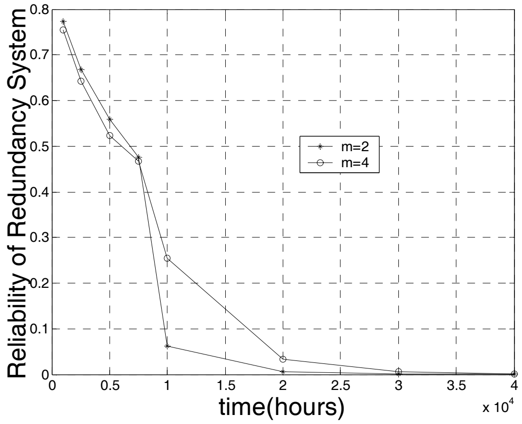

Figure7 . M=2 and M=4 in TMRSB System

5. CONCLUSION

A new approach to realizing a fault tolerance architecture is feasible using dynamic reconfiguration under a hybrid of TMR and Standby models. The target application platform are FPGAs or other reconfigurable logic devices. Based on the analysis and simulation, we can reach the conclusion that the TMRSB system can improve the system reliability with lower storage overhead over certain ranges of system longevity.

The reliability of standby-based systems may not linearly increased with the number of standby configurations. The reliability of the configurations which are both active and standby will be an essential factor determining overall reliability. TMRSB presents an interesting resource cost vs. reliability benefit tradeoff with a new interpretation because the standby configurations do not require distinct functional resources: they only reconfigure the same original resource pool. We are currently implementing the TMRSB approach for benchmark applications on a Xilinx FPGA hardware platform which will allow tradeoff of switch delay and diverse standby configurations.

REFERENCES

[1] Carmichael, C., et al, “SEU Mitigation Techniqies for Virtex FPGAs in Space Applications,” MAPLD99, Laurel, MD USA, pp.24-32, September 28-30 1999.

[2] J. Von Neumann, “Probabilistic logics and synthesis of reliable organisms from unreliable components,” in Automata Studies, C. E. Shannon and J. McCarthy, Eds. Princeton, NJ: Princeton Univ. Press, pp.43–98, 1956.

[3] Xilinx Inc , “Triple Module Redundancy Design Techniques for Virtex FPGAs”, November 2001, available at http://www.xilinx.com/bvdocs/appnotes/xapp197.pdf

[4] Xilinx Inc , “Virtex-II Pro and Virtex-II Pro X Platform FPGAs: Complete Data Sheet”, October 2005, available at http://www.xilinx.com/bvdocs/publications/ds083.pdf

[5]K. Zhang, R. F. DeMara, C. A. Sharma, “Consensus-based Evaluation for Fault Isolation and On-line Evolutionary Regeneration,” in Proceedings of the International Conference in Evolvable Systems (ICES'05), pp. 12 - 24, Barcelona, Spain, September 12 - 14, 2005.

[6] Joel F. Bartlett, “A nonstop kernel,” in the Proceedings of the Eighth ACM symposium on Operating systems Principles, Pacific Grove, California, United States, pp 22-29, 1981.

[7] Srikant Sharma, Jiawu Chen, Wei Li, et al. “Duplex: A Reusable Fault Tolerance Extension Framework for Network Access Devices,” In Proceedings of 2003 International Conference on Dependable Systems and Networks (DSN 2003), June 2003.

[8] F. Lombardi, N. Park, M. Al-Hashimi, and H. H. Pu, “Modeling the dependability of N-modular redundancy on demand under malicious agreement,” In Proceedings of the Pacific Rim International Symposium on Dependable Computing, pp. 68 – 75, Dec. 2001.

[9] J. Lach, W.H. Mangione-Smith, and M. Potkonjak, “Low Overhead Fault-Tolerant FPGA Systems,” IEEE Transactions on VLSI Systems, Vol. 6, No. 2, June 1998, pp 212-321.

[10] F. Lima et al. “A fault injection analysis of Virtex FPGA TMR design methodology,” on Radiation and Its Effects on Components and Systems 6th European Conference, pp.275–282, 10–14 Sept. 2001.

[11] Swapna S. Gokhale, Michael Rung-Tsong Lyu, “A Simulation Approach to Structure-Based Software Reliability Analysis,” IEEE Transactions on Software Engineering, Vol. 31, No. 8 pp. 643-656, August 2005.

[12] S. Vigander, Evolutionary Fault Repair of Electronics in Space Applications, Dissertation, Norwegian University Sci. Tech., Trondheim, Norway, February 28, 2001.

[13]M. Garvie and A. Thompson, “Scrubbing away transients and jiggling around the permanent: Long survival of fpga systems through evolutionary self-repair,” in Proc. 10th IEEE Intl. On-Line Testing Symposium (C. Metra, R. Leveugle, M. Nicolaidis, and J. Teixeira, eds.), IEEE Computer Society, 2004, pp. 155–160.

[14]L.Sterpone and M. Violante, “Analysis of the Robustness of the TMR Architecture in SRAM-Based FPGAs,” IEEE Transaction on Nuclear Science, Vol. 52, No. 5, October, 2005

[15] Kishar Shridharbhai Trivedi , “Probability and Statistics with Reliability, Queuing and Computer Science Applications”, pp 375-378., Prentice Hall, 1982,