

ON THE RELIABILITY EVALUATION OF SRAM-BASED FPGA DESIGNS[†]

Olivier Héron, Talal Arnaout, Hans-Joachim Wunderlich

Institut für Technische Informatik,
Universität Stuttgart
Pfaffenwaldring 47; D-70569 Stuttgart, Germany.
email: heron@lirmm.fr ; {talal.arnaout, wu}@informatik.uni-stuttgart.de

ABSTRACT

Benefits of Field Programmable Gate Arrays (FPGAs) have lead to a spectrum of use ranging from consumer products to astronautics. This diversity necessitates the need to evaluate the reliability of the FPGA, because of their high susceptibility to soft errors, which are due to the high density of embedded SRAM cells. Reliability evaluation is an important step in designing highly reliable systems, which results in a strong competitive advantage in today's marketplace. This paper proposes a mathematical model able to evaluate and therefore help to improve the reliability of SRAM-based FPGAs.

1. INTRODUCTION

Semiconductor devices wear out with use and suffer from two classes of failure mechanisms: physical and functional. Physical failures are permanent, essentially due to defects resulting from processing, manufacturing, packaging, metallization, bonding, die attachment failure, or particle contamination [1]. Functional failures, on the other hand, are transient and intermittent failures due to strikes of a high energy neutron / proton (present in terrestrial cosmic radiation) or an alpha particle (that originate from impurities in the packaging materials) on the sensitive parts of the device during its operation [1]. The radiation may cause a bit flip in some latch (either 0 to 1 or 1 to 0), thereby altering the functionality of the device. This phenomenon is known as Single Event Upset (SEU). Since these SEUs cause only a bit flip, without causing a permanent damage to the device, their effects are classified as “soft errors”. SRAM-based FPGAs are very sensitive to SEUs for two reasons: first, their high gate density leads to a large number of latches or SRAM cells; secondly SRAM cells are not only used for their data storage, but also to define both the circuit structure and functionality.

This paper develops a reliability model for SRAM-based FPGAs able to predict the probability that a configured FPGA will perform its task correctly over time

or will verify the user reliability expectations. In addition, this model can be used to identify the weak part of a design and aid CAD tools in making the design more reliable. The model we propose receives as inputs the failure rate and SEU rate of the device along with the design characteristics, with the assumption that the design is non-redundant. The outcome is a time-dependent probability of correct operation.

The remainder of the paper is organized as follows. In Section 2, we define reliability concepts and present briefly the previous works dealing with reliability improvement in SRAM-based FPGAs. In Section 3, we analyze the effects of SEUs in a design. From this analysis, Section 4 first develops a procedure to compute the reliability of an FPGA and next discusses the issues of the proposed model. Section 5 concludes this paper.

2. PRELIMINARIES

2.1. Reliability definition

Reliability is a time dependent probability that a device remains functional under specified conditions. The reliability depends mainly on a term known as the failure rate, which can be viewed as the number of failures observed in a population of devices during a unit time [1]. If the failure rate is represented by a function $h(t)$, then the reliability can be expressed as:

$$R(t) = e^{-\int_0^t h(x).dx} \quad (1)$$

As for physical failures, a study of many systems during their normal life expectancy has led to the conclusion that physical failure rates follow a certain basic pattern: the “bathtub” curve, as shown in Figure 1. During the infant mortality period, the devices display a high failure rate due to imperfections in the manufacturing process, which can be reduced by burn-in. Failures in the last period are typically due to aging, wear out or cumulative damage. Semiconductor devices spend most of their life in the flat portion of this curve (useful life period) that can account for almost 40 years and tends to be very low [1].

[†] This research work is funded by the Deutsche Forschungsgemeinschaft (DFG) under contract FOR 460/1-1

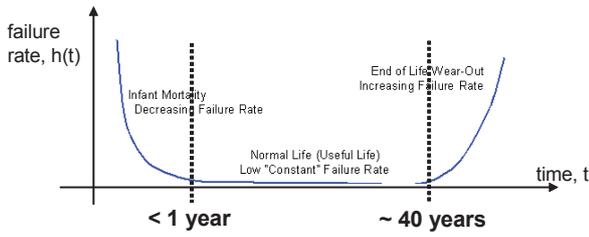


Fig. 1. "Bathtub" curve for semiconductor devices

Reliability stress tests are used to collect information for predicting the failure rate in the useful life period. One common technique is to speed up the deterioration of materials by applying accelerated life tests, which highly overstress the device.

As for functional failures, which are the main focus of this paper, the behavior is not yet clearly modeled, as the occurrence of SEUs is random and nondeterministic during the device's operation. To be able to model the behavior of the failure rate with respect to SEUs, we need to define SEU rate (SEUR) of the device as the device's functional failure rate. The SEUR could be best understood as the number of observed SEU occurrences per 10^9 hours of device functioning.

Radiation stress testing helps in determining this parameter under given test conditions. In a general way, the SEU rate is estimated through the Neutron / proton Cross Section method (NCS) [2]. The SEUR of a configuration bit using the NCS method is given by:

$$SEUR = \int_{E_{min}}^{\infty} \sigma(E) * \frac{dN(E)}{dE} * dE \quad (2)$$

where $\sigma(E)$ is the neutron SEU cross-section of the device (cm^2), defined as the ratio between the number of SEU occurrences and the neutron / proton flux in the environment (n or p / cm^2). The term $(dN(E)/dE)$ is the differential neutron flux (n or $p / \text{cm}^2 \cdot \text{MeV} \cdot \text{s}$), which mainly depends on the geographic altitude and latitude. Hence, the SEU rate increases with altitude [2].

The integration is carried out over all particle energies greater than the minimum particle energy E_{min} needed to create a glitch of sufficient strength to change the value of the configuration bit. Thus, the functional failure rate of an SRAM-based FPGA can vary with time, depending on its environment.

2.2. Related research and motivation

Many methods to improve the reliability of FPGAs rely on state-of-the-art test techniques during manufacturing. The goal of these techniques is to detect static faults, such as stuck-at faults and bridging faults [3-7], or dynamic faults, such as delay faults [8, 9, 10], that result from the occurrence of random and non-random defects in the device

(e.g. short circuits, parameter deviations, etc.). As a result, the use of these techniques allows rejecting the defective parts from the assembly line, i.e. those which do not exhibit operational specifications.

On the other hand, some defects can escape these tests and are exposed during the device's operational lifetime. Fault-tolerance techniques are proposed to bypass such occurrences in FPGAs during their functioning [11, 12, 13]. Although these techniques may increase the reliability of the device, they result in an increased system cost (due to the use of extra units) and a depreciated performance (due to design rerouting).

In addition to that, some publications even analyze the SEU sensitivity of SRAM-based FPGAs by simulating SEUs through a fault injection mechanism into the configuration bit stream [14, 15]. Reference [16] attempts to deliver an estimation of the Mean Time Between Failures (MTBF) for SRAM-based FPGA designs through an empirical method. However, none of these methods evaluates the reliability of a configured SRAM-based FPGA over time while considering the characteristics of the design running in it.

3. SEU EFFECTS

An SEU in an SRAM cell or a latch may disturb the functionality of the device depending on the SEU location and the item's behavior. Extending the definition set in [16], an *essential item* is an FPGA component which is included in the design area, i.e. configured by one or more *essential bits*, and whose configuration state affects the functionality of the design. The effects of SEUs can be looked upon at two points: essential logic items, such as look-up tables (LUT) and flip-flops (FF)/Latch, and essential routing items, such as multiplexers (MUX) and switches. Starting with essential logic items, the following effects can be noted:

- The occurrence of an SEU in an essential LUT changes the combinational function realized by it, assuming that the essential LUT operates only as a function generator. A (functional) failure appears in the essential LUT if its upset configuration bit is read, i.e. the input vector connects the output to it.
- The occurrence of an SEU in an essential FF/Latch causes a change in either its content or operating mode (FF or Latch). For both cases, a failure appears in the design if the corresponding register displays a wrong state. The fault-free behavior or the correct states of a register is determined by the nature of the sequential circuit.

Moving on to essential routing items, the following effects can be noted:

- If an SEU occurs in an essential switch, the switch flips to the opposite state. As a result, a break can occur within an already configured path (an ON

switch flips to OFF), or a short between two configured paths (an OFF switch located between them flips to ON).

- An essential MUX connects its output to one given data input. If an SEU occurs in the selection bit(s), the data input is deactivated, thus feeding into the circuit an unknown input.

An SEU in essential logic items does not always cause a functional failure; since usually it can be tolerated by the large don't care space in the device. On the other hand, an SEU in the essential routing items does in principle cause a functional failure, especially in non-redundant designs. From this analysis, we can derive the probability of an essential item failing when an SEU occurs in it, which is denoted by:

$$Pr(\text{essential item fails} / 1 \text{ SEU} \rightarrow \text{essential item}).$$

For essential logic items, determining the probability of failure requires the use of static analysis of the design. On the other hand, for essential routing items, we assume that the probability of failure is very close to 1 since SEUs, most of the times, cause them to fail, as shown indirectly in [17]. Without loss of generality, we will assume that, for a non-redundant design:

$$Pr(\text{essential routing item fails} / 1 \text{ SEU} \rightarrow \text{essential routing item}) = 1.$$

4. RELIABILITY MODEL

In this section, we propose a procedure that computes the reliability model for an FPGA. This model highly depends on the design's characteristics. To illustrate this model, we detail the derivation in what follows, after which we present a case study that applies the model. Finally, we tackle some fault-tolerance issues in SRAM-based FPGAs through using our model.

We base our overall FPGA reliability on a "series model". The assumption here is that the device is running a non-redundant minimized design that fails once the first failure mechanism occurs, and that all failure mechanisms are statistically independent. Thus, the overall failure rate would be the sum of all failure rates resulting from the considered failure mechanisms and the overall reliability is just the product of all single reliabilities. For SRAM-based FPGAs, the overall reliability $R_{FPGA}(t)$ is the product of two terms, one representing the reliability of the device with respect to physical failures $R_{structure}(t)$ and the other representing the reliability of the design with respect to functional failures $R_{design}(t)$, as shown in (3):

$$R_{FPGA}(t) = R_{structure}(t) * R_{design}(t) \quad (3)$$

where t is the expected period of functioning. The device is assumed to be working properly at $t = 0$, i.e. $R_{FPGA}(0) = 1$. The reliability of the device with respect to physical failures is represented by an exponential distribution [1], in which

the only parameter is the failure rate provided by the vendor, and exhibits a behavior similar to the one depicted with the "bathtub" curve in Figure 1. However, the reliability of the device with respect to functional failures depends on its SEU sensitivity and on the design's characteristics.

4.1. Reliability of the design

To be able to compute the reliability of the design $R_{design}(t)$, we need to start with a failure model of the design to determine the probability that a design fails when SEUs occur in it. Then, the reliability is derived from this model by considering that this figure of merit is the probability of "success" over time.

A design can be considered as a set of p partitions of essential items. A partition P_i could be composed of m_i essential LUTs or m_i essential switches or etc... (m_i being the number of essential items in the partition P_i). Let the event "functional failure", or simply "failure", be the event that the design fails. The event "failure", which is the union of the p sub events "failure" resulting in each partition P_i , is formulated as follows:

$$\text{failure} = \bigcup_{i=1}^p \{ \text{failure} \rightarrow P_i \} \quad (4)$$

where $\text{failure} \rightarrow P_i$ represents the event "failure" in the partition P_i .

Accordingly, the probability of a failure in the design would be the probability of the union of all sub events "failure" in the p partitions:

$$Pr(\text{failure}) = Pr\left(\bigcup_{i=1}^p \{ \text{failure} \rightarrow P_i \}\right) \quad (5)$$

As mentioned at the beginning of the section, we assume that the design fails when the first functional failure occurs in one of the p partitions. In other words, we assume that a partition P_j does not mask the occurrence of a functional failure in a partition $P_k \neq P_j$. A functional failure in the design is caused by a functional failure either in partition P_j , or in P_k , or in several partitions. Concerning the last case, either SEUs occur in several partitions or only one SEU occurs in a partition P_i but the partition P_i causes other partitions to display a failure. From this assumption, the probability of failure of a design is the sum of probabilities of all the possible failures in a design, which can be factorized as follows:

$$Pr(\text{failure}) = 1 - \prod_{i=1}^p [1 - Pr(\text{failure} \rightarrow P_i)] \quad (6)$$

Now, we focus on the probability that the event "failure" appears in a partition P_i . A partition is assumed to

display a “failure” when the first of its essential items fails. In a similar fashion to a design, a partition fails when one or more of its essential items fail. Hence, the probability of failure of a partition P_i is similar to equation (6), which is represented as follows:

$$Pr(\text{failure} \rightarrow P_i) = 1 - \prod_{j=1}^{m_i} [1 - Pr(\text{failure} \rightarrow I_j \in P_i)] \quad (7)$$

where m_i is the number of essential items in the partition P_i , and $\text{failure} \rightarrow I_j \in P_i$ is the event “failure” in the essential item j , denoted by I_j , in partition P_i .

Moving on with the analysis, we need now to calculate $Pr(\text{failure} \rightarrow I_j \in P_i)$. An essential item of P_i fails if and only if it is included in the partition P_i , an SEU occurs in it, and the SEU causes the essential item to fail. Therefore, the probability that an essential item I_j of P_i fails can be expressed by equation (8) as shown below, where:

- $Pr(I_j \in P_i)$ is the ratio between the number of essential items in partition P_i , denoted by m_i , and the total number of essential items in the design; denoted by $Card(\text{design})$.
- $Pr(\{1 \text{ SEU} \rightarrow I_j\} / \{I_j \in P_i\}, t)$ is the time-dependent probability of an SEU occurrence in I_j given that I_j belongs to P_i . We express this probability from the expression proposed in [18] based on the Poisson’s assumption. The distribution of SEUs is random in the device and the occurrence of an SEU at any location is independent of the occurrence of any other SEU. Hence, this probability could be expressed as:

$$Pr(\{1 \text{ SEU} \rightarrow I_j\} / \{I_j \in P_i\}, t) = 1 - e^{-SEUR(t) * b_j * t} \quad (9)$$

where b_j is the number of SRAM cells used in I_j , t is the expected period of functioning, and $SEUR(t)$ is the SEU rate of the device at time t . For a given $SEUR(t)$, the probability of SEU occurrence increases with the period of functioning and the size of a partition (number of SRAM cells), as expected.

- $Pr(\{I_j \text{ fails}\} / \{1 \text{ SEU} \rightarrow I_j\})$ is the probability that I_j of partition P_i fails when an SEU occurs in it (Section 3). Note that the probability of failure for routing items is equal to 1 when a non-redundant design is considered. For logic items, their probability of failure depends on the design behavior.

$$Pr(\text{failure} \rightarrow I_j \in P_i) = Pr(I_j \in P_i) * Pr(\{1 \text{ SEU} \rightarrow I_j\} / \{I_j \in P_i\}, t) * Pr(\{I_j \text{ fails}\} / \{1 \text{ SEU} \rightarrow I_j\}) \quad (8)$$

$$R_{\text{design}}(t) = \prod_{i=1}^p \left[\prod_{j=1}^{m_i} \left(1 - \frac{m_i}{Card(\text{design})} * \left(1 - e^{-SEUR(t) * b_j * t} \right) * Pr(\{I_j \text{ fails}\} / \{1 \text{ SEU} \rightarrow I_j\}) \right) \right] \quad (10)$$

From this failure model, we can now derive the reliability of a design, which is the probability of its success over time. Thus, the reliability is simply the complement of the probability of failure. From equations (6), (7), (8) and (9), R_{design} can be evaluated using equation (10).

We observe that this reliability model can be approximated by an exponential law. We denote the parameter of the exponential function as the “soft error rate of the design” ($SEUR_{\text{design}}$), which can be defined as the number of functional failures or “soft errors” per 10^9 hours of device functioning, yielding the equation (11):

$$R_{\text{design}}(t) \approx e^{-SEUR_{\text{design}}(t) * t} \quad (11)$$

Finally, from equations (1), (3) and (11), the reliability of SRAM-based FPGAs can be calculated using the equation (12):

$$R_{\text{FPGA}} = e^{-\lambda * t} * e^{-SEUR_{\text{design}}(t) * t} \quad (12)$$

where λ is the failure rate of the device.

4.2. Case studies

We now validate our reliability model by a case study using the Virtex XC2V3000 FPGA from Xilinx. For this device, the following information is available:

- Failure rate: $\lambda = 2.98 * 10^{-4}$ output errors / year-device [19].
- SEU rate: $SEUR = 7.24 * 10^{-9}$ SEUs / year-bit-device at sea level [20] (LANSCE results). Here, we assume that the SEUR remains constant over time.
- Overall number of SRAM cells: $\mathcal{N} = 10,494,368$ bits [21].

A cluster (CLB) of the XC2V3000 FPGA is composed of 8 four-input LUTs having 16 SRAM bits, 8 FFs/Latches, 88 MUXs having 2 selection SRAM bits on average and switches having 1 SRAM bit [21].

We have implemented the LEON3MP system on chip, LEON2 processor, crypto-core AES128 and an ISCAS’89 benchmark into the XC2V3000, using Xilinx ISE CAD tool. Using the Xilinx XDL-translator, we have converted the respective Xilinx design netlist files (NCD) into the readable XDL format. This enables us to parse the file and identify the number of LUTs, MUXs, FFs/Latches, wire and switch resources used in the design. This information, along with the above CLB data, yields the number of essential bits used for configuring these essential items.

Table 1. Characteristics and "soft error" rates of designs

Bench.	#LUTs (α_L)	#FFs (α_F)	#MUXs (α_M)	#ON-switches(α_S)	SER ($\cdot 10^{-4}$)	SER [16] ($\cdot 10^{-4}$)
LEON3MP	21875 (0.0466)	8079 (0.0172)	88435 (0.1882)	351463 (0.7480)	22.4	33.8
LEON2	5172 (0.0452)	1585 (0.00067)	762 (0.0138)	107002 (0.9343)	7.46	8.22
AES128	20242 (0.0814)	630 (0.0025)	498 (0.002)	227365 (0.9141)	16.6	17.9
s38584	3543 (0.0706)	1299 (0.0259)	281 (0.0056)	45029 (0.8979)	3.16	3.60

Collecting information about the OFF switches is difficult, since it is a complicated task to differentiate between OFF switches within the design area and OFF switches outside the design area. Hence, and without loss of generality, we consider only ON switches in this case study.

Table 1 shows the number of essential items used by each design (columns 2-5). The variable α given in parentheses is the ratio between the number of essential items in a partition (m_i) and the whole design ($Card(design)$). Column 6 gives the soft error rate of the corresponding design (SER_{design}), denoted as SER ("soft errors" / year-device). Column 7 derives the soft error rate obtained by using the empirical model from [16].

In order to generate the SER values using the method in [16], we had to first derive the following values from the experiments performed in [21]:

- Neutron flux: 13.9 n/cm²·hr
- Area cross section= 5.9*10⁻¹⁴ cm²/bit.

Next, the SER was obtained by computing the inverse of the MTBF [16]. Clearly, all the empirical SER values generated by the method in [16] are greater than those obtained by using our model, as expected. This difference is due to the authors' pessimistic assumption that a device fails whenever an SEU occurs onto the design area, irrespective of the nature of the affected item.

We assume that all the essential LUTs have the same failure probability. Similar assumption applies for essential FFs. We assign randomly a failure probability of 0.8 for partition LUT, and 0.5 for partition FF/Latch. We will show that these random values will exhibit a low impact on the reliability.

Figure 2 depicts the variation of the reliability of each design for a 4-year operational period using equation (10). This is represented by the discrete points marked on the chart. In addition, it presents the exponential approximation given in equation (11), which is represented by the continuous graphs. A well noticeable remark is that, as expected, the exponential law fits well over our model for this time period, rendering it a confident approximation.

Figure 3 shows the variation of the reliability $R_{FPGA}(t)$ of the XC2V3000, over time using equation (12). Comparing figures 2 and 3, we can see that the reliability loss in the XC2V3000 is mainly dominated by the SEU sensitivity of the design in it, as expected.

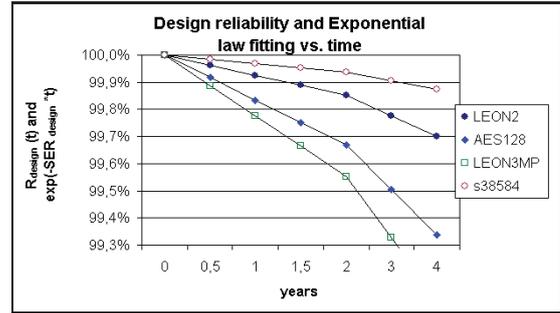


Fig. 2. Reliability of each design

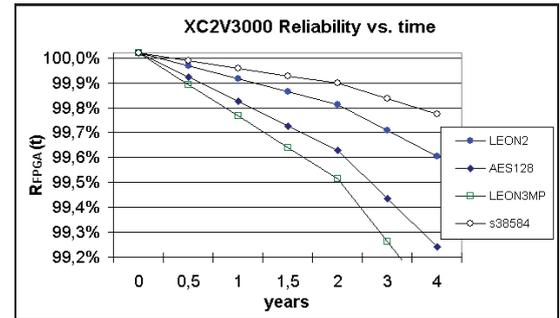


Fig. 3. Reliability of XC2V3000

4.3. Discussion

The proposed reliability model could be comprehended as follows. If we consider a population of 1000 XC2V3000 FPGAs shipped to a user who would implement the same design, say LEON2, onto them, then as shown in Figure 3, the expectation is that around 0.36% of them may fail after 4 years of functioning, i.e. 4 FPGAs. One should note that the functionality of these defective parts may be recovered by simply reloading the bit stream.

Coming into using our model for improving reliability, the following discussion can be raised. If the number of defective parts expected meets the user's outlook, then no fault-tolerance method would be necessary, thus having major savings in cost. Conversely, if this expectation does not meet the user's outlook, then fault-masking methods should be applied in order to improve the FPGA's reliability. One simple approach is to insert redundant items in the design. However, not all items should be replicated, rather only those causing a significant reliability loss.

As seen in Table 1, the switches are the most sensitive items to SEUs. Adding redundant switches will decrease their failure probability, hence improving the reliability. To illustrate this remark, Figure 4 shows the impact of the failure probability of switches, $Pr(\{switch\ fails\}) / \{1\ SEU \rightarrow switch\}$, in LEON2 on the overall reliability, assuming the same previous design. In this figure, we consider that all switches are replicated. The failure probability of switches takes the values 0.01, 0.25, 0.75 and 0.9. The lower the probability is, the higher the number of parallel switches.

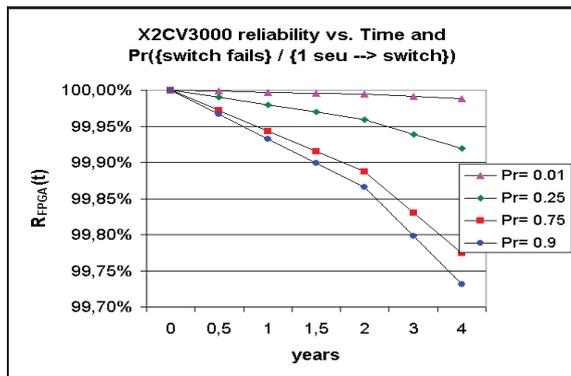


Fig. 4 Reliability improvement in LEON2 design

As expected, the reliability increases significantly when this probability decreases. On the other hand, replicating LUTs will not impact the reliability of the device as much.

In a fault-tolerance context, our model can be used as a “guide” to determine reliability critical items i.e. to determine the type and/or number of items that should be either replicated in the design through the definition of specific constraints in the CAD tools, or monitored during the device functioning through the use of online testing and diagnosis methods. This is not possible with an analysis similar to the one carried out in [16].

5. CONCLUSION

In this paper, we have first analyzed the effects of SEUs in SRAM-based FPGAs. From this analysis, we have derived a reliability model of a design in an FPGA. Finally, we have proposed an overall reliability formulation for an SRAM-based FPGA, without redundancy. This model has been applied to Xilinx XC2V3000 implementing several benchmarks as a case study. As expected, most of reliability loss is due to the SEU sensitivity of large designs in the device. Further work will apply this method to redundant and fault-tolerant designs by parallel / series reliability evaluation.

6. REFERENCES

- [1] W. Kuo, W.-T. K. Chien, and T. Kim, “Reliability, Yield and Stress Burn-in,” *Kluwer Academic Publishers*, 1998.
- [2] J.F. Ziegler and W.A. Lanford, “Effect of Cosmic Rays on Computer Memories,” *Science*, vol. 206, Nov. 1979, pp 776-788.
- [3] M. Renovell and Y. Zorian, “Different Experiments in Test Generation for XILINX FPGAs,” *Proc. of IEEE Int. Test Conf.*, 2000, pp. 854-861.
- [4] D.A. Fernandes, and I.G. Harris, “Application of Built in Self-Test for Interconnect Testing of FPGAs,” *Proc. of IEEE Int. Test Conf.*, 2003, pp. 1248-1257
- [5] W.K. Huang, F.J. Meyer, X.T. Chen and F. Lombardi, “Testing Configurable LUT-Based FPGAs,” *IEEE Trans. on VLSI Systems*, vol 6, 1998, pp 276-283.
- [6] C. Stroud, S. Konola, P. Chen and M. Abramovici, “Built-In Self Test of Logic Blocks in FPGAs,” *Proc. of IEEE VLSI Test Symp.*, 1996, pp. 387-392.
- [7] C. Stroud, E. Lee and M. Abramovici, “Built-In Self Test of FPGA Interconnect,” *Proc. of IEEE Int. Test Conf.*, 1998, pp. 404-411.
- [8] O. Héron et al., “Manufacturing-Oriented Testing of Delay Faults in the Logic Architecture of Symmetrical FPGAs,” *Proc. of 1st IEEE European Test Symp.*, 2004, pp 152-157.
- [9] E. Chmelar, “FPGA Interconnect Delay Fault Testing,” *Proc. of IEEE Int. Test Conf.*, 2003, pp 1239 - 1247.
- [10] M. B. Tahoori, “Testing for Resistive Open Defects in FPGAs,” *Proc. of IEEE Int. Conf. on Field Programmable Technology*, 2002, pp 332-335.
- [11] M. Abramovici et al., “Using Roving STARs for On-line Testing and Diagnosis of FPGAs in Fault-Tolerant Applications,” *Proc. of IEEE Int. Test Conf.*, 1999, pp 973-982.
- [12] J. M. Emmert, and D. K. Bhatia, “A Fault Tolerant Technique for FPGAs,” *Journ. of Electronic Testing: Theory and Applications*, vol. 16, 2000, pp 591-606.
- [13] J. Lach, W.H. Mangione-Smith and M. Potkonjak, “Low Overhead Fault-Tolerant FPGA Systems,” *IEEE Trans. on VLSI Systems*, vol.6, no. 2, 1998, pp 212-221.
- [14] E. Johnson et al., “Accelerator Validation of an FPGA SEU Simulator,” *IEEE Trans. on Nuclear Science*, vol. 50, no. 6, 2003, pp 2147-2157.
- [15] P. Bernardi et al., “On the Evaluation of SEU Sensitiveness in SRAM-Based FPGAs,” *Proc. IEEE of the 10th Int. On-Line Testing Symp.*, 2004, pp 115-120.
- [16] P. Sundararajan et al., “Estimation of Single Event Upset Probability Impact of FPGA Designs,” *Online Proc. of 6th MAPLD Int. Conf.*, Washington, D.C., USA, 2003.
- [17] P. Graham et al., “Consequences and Categories of SRAM FPGA configuration SEUs,” *Online Proc. of the 6th MAPLD Int. Conf.*, Washington, D.C., USA, 2003.
- [18] G. Lum and G.Vandenboom, “Single Event Effects Testing of Xilinx FPGAs,” *Online Proc. of the 1st MAPLD Int. Conf.*, Greenbelt, Maryland, USA, 1998.
- [19] Xilinx, “Device Reliability Report,” Xilinx Inc, Report UG116 (V2.3), San Jose, CA, USA, 2004, 2nd quarter.
- [20] E. Dupont et al., “Radiation results of the SER Test of Actel, Xilinx and Altera FPGA instances,” Test Report No 0.06A, iRoC Technologies, 2004.
- [21] Xilinx, “Virtex-II Series Product Specification (FPGAs),” Rep. DS031, San Jose, CA, USA, 2004.