

# SRAM FPGA Reliability Analysis for Harsh Radiation Environments

Patrick S. Ostler, Michael P. Caffrey, Derrick S. Gibelyou, Paul S. Graham, Keith S. Morgan, Brian H. Pratt, Heather M. Quinn, and Michael J. Wirthlin

**Abstract**—This paper investigates the viability of deploying SRAM-based FPGAs into harsh Earth-orbit environments. A reliability model is presented for estimating the *MTTF* of SRAM FPGA designs in specific orbits and orbit conditions. The model requires orbit- and condition-specific SEU rates and design-specific estimates of the probability of failure during a single scrubbing period. Probability of failure estimates are reported for several FPGA designs from both fault-injection and accelerator experiments. The model also includes a method for estimating composite mean time to failure (*MTTF*) that incorporates all orbit conditions over a solar cycle. Despite using pessimistic assumptions, the results from this model suggest that SRAM FPGA designs protected by TMR and scrubbing operate very reliably in a LEO orbit and surprisingly well in “harsh” orbits.

**Index Terms**—FPGAs, redundancy, reliability modeling, single event effects.

## I. INTRODUCTION

HERE is growing interest in using SRAM-based Field Programmable Gate Arrays (FPGAs) within space systems due to low non-recurring engineering (NRE) costs, computational efficiency benefits over general purpose processors, and reconfigurability. A variety of projects have demonstrated the benefits of using FPGAs in spacecraft [1], [2]. Specific examples include the Mars rovers, which use FPGAs for motor control and landing pyrotechnics [3], and the Los Alamos National Laboratory satellite CFESat, which uses nine FPGAs as part of its high performance computing payload [4], [5].

SRAM-based FPGAs, however, are sensitive to the space radiation environment, particularly radiation-induced single-event upsets (SEUs). The safe use of FPGAs in space requires careful design considerations and the use of well-proven SEU mitigation techniques. The most common technique combines triple-modular redundancy (TMR) and configuration memory scrubbing [6]. While TMR and scrubbing are effective, this technique is not particularly well suited to protect against

coincident SEUs that occur within a single scrub cycle. Coincident upsets violate the assumptions of TMR and therefore can cause the mitigated circuit to fail. This paper investigates the reliability of FPGA circuits protected by TMR and scrubbing operating in harsh orbits where coincident upsets are more probable.

This paper begins by describing TMR and scrubbing, the mitigation techniques used most often for SRAM FPGAs. A reliability model is then presented for estimating the design-specific probability of failure, failure rate, and mean-time to failure (*MTTF*). Design-specific SEU sensitivity data, obtained through fault injection and radiation testing, are obtained for several designs and used in the reliability model. Reliability estimates are then reported for these designs in several orbits. These estimates show that while harsh orbits or solar energetic particle (SEP) events [7] do indeed increase the probability of FPGA design failure, SRAM FPGAs are predicted to operate reasonably well in such harsh conditions.

## II. FPGA SEU MITIGATION TECHNIQUES

Triple modular redundancy (TMR) is a well known fault mitigation technique that uses redundant hardware to tolerate faults. A circuit protected by TMR has three redundant copies of the original circuit and a majority voter. A single fault in any of the redundant hardware modules will not produce an error at the output as the majority voter will select the correct result from the two working modules.

TMR is used extensively in SRAM FPGA systems to mitigate against SEUs. Design tools have been created for automating the application of TMR on FPGAs to simplify the design process [8], [9]. These design tools automatically triplicate design resources, insert voters, and apply voting in circuit feedback paths to insure sequential structures are resynchronized [10]. Several experiments have demonstrated significant improvements in reliability of TMR through fault injection and radiation testing [11].

While TMR is effective at protecting a circuit from SEUs, it cannot protect the circuit from multiple independent SEUs. If multiple SEUs occur within the configuration memory, two or three copies of the redundant circuit may fail. With more than one failure, the majority voters may choose the incorrect value and overcome the benefits of redundant hardware.

Configuration scrubbing is used in conjunction with TMR to prevent the accumulation of multiple coincident SEUs. Like conventional memory scrubbing, configuration scrubbing involves the continuous reading and repairing of the configuration data to prevent the accumulation of SEUs. Most FPGA scrubbing techniques require some external hardware including external memory for configuration data storage. Like

Manuscript received July 16, 2009; revised September 14, 2009. Current version published December 09, 2009. This work was supported by the Department of Energy through the Sensor-Oriented Processing & Networking and Joint Architecture Standard projects and the I/UCRC Program of the National Science Foundation under Grant 0801876.

P. S. Ostler, M. P. Caffrey, P. S. Graham, K. S. Morgan, and H. Quinn are with the Space Data Systems Group, Los Alamos National Laboratory, Los Alamos, NM 87544 USA.

D. S. Gibelyou, B. H. Pratt, and M. J. Wirthlin are with the NSF Center for High-Performance Reconfigurable Computing (CHREC), Department of Electrical and Computer Engineering, Brigham Young University, Provo, UT 84604 USA (e-mail: wirthlin@ee.byu.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TNS.2009.2033381

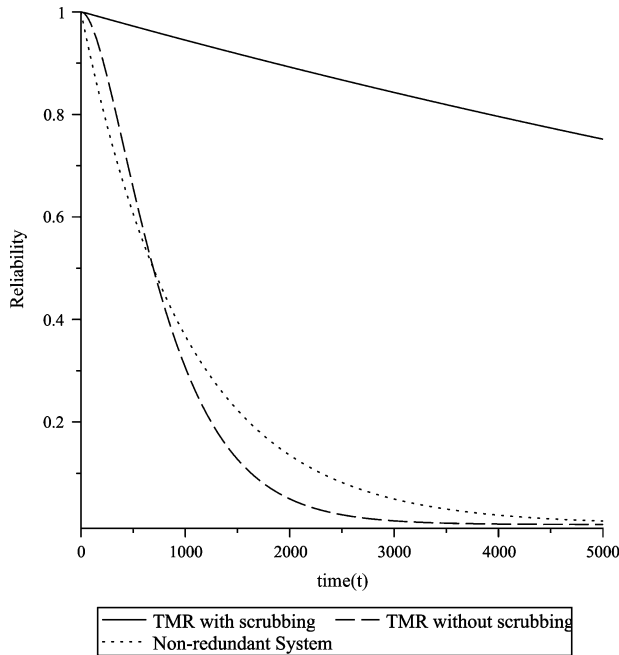


Fig. 1. Continuous Time Reliability of a System with No TMR, TMR, and TMR with Scrubbing [15].

memory scrubbing, there are a variety of ways to implement configuration scrubbing in FPGAs [12], [13]. Additionally, the time required to perform an individual scrubbing cycle on an entire device is dependent upon the size of the device and the implementation of the scrubber.

The combination of TMR and configuration scrubbing within an FPGA is much like the use of error correction codes (ECC) and scrubbing within non-volatile memories. In a previous paper, we presented a continuous time Markov model that demonstrates the improvements in reliability of FPGAs using TMR and scrubbing [14]. Fig. 1 demonstrates a continuous time reliability plot of three FPGA-based systems: a system without TMR, a system with TMR but no scrubbing, and a system with TMR and scrubbing. The system that combines TMR and scrubbing has a much higher reliability than either of the other systems.

### III. RELIABILITY MODEL

While the use of TMR and configuration scrubbing is very effective at mitigating against the effects of configuration SEUs, these techniques cannot prevent the possibility that multiple configuration upsets will occur *within* a single scrub period and overcome TMR. This paper introduces a reliability model that estimates the reliability of FPGA designs protected by TMR and configuration scrubbing. This model will be used to estimate the reliability of FPGA designs in harsh orbits when multiple coincident upsets are more probable<sup>1</sup>. We will also

<sup>1</sup>The model presented ignores multiple-bit upsets (MBUs) i.e., more than one upset from a single charged particle [16], [17]. MBUs will appear like other coincident upsets, however they are spatially correlated upsets and therefore should have less effect than independent coincident upsets. In future work we will augment the model to include MBUs

show how this model can be used with orbit-specific upset rates to estimate a composite reliability measure for intervals that span multiple orbit conditions.

Several other models for estimating the reliability of SRAM-FPGAs in the presence of SEUs have been introduced. Héron *et al.* introduced a single-parameter reliability model that combines SEU reliability and physical reliability [18]. This model analyzes the netlist of a design to estimate the essential failure modes of the design. This model, however, does not take into account the effects of hardware redundancy such as TMR or the effects of multiple coincident upsets. Edmonds introduced an analytic reliability model specifically for FPGA designs mitigated with TMR that accounts for coincident upsets [19]. The reliability model introduced in this paper takes a unique approach that integrates the results from design-specific fault injection experiments or radiation testing. Further, the model introduced in this paper estimates a composite failure rate of an orbit by combining the effects of different orbit conditions.

This model is based on  $R_s$ , a design-specific parameter for the reliability of an FPGA design during a single configuration scrubbing period. A related parameter,  $Q_s$ , is the unreliability of an FPGA design during a single configuration scrubbing period, where  $Q_s = 1 - R_s = P(F_s)$  and  $P(F_s)$  is the probability that the design will fail during a single scrub cycle<sup>2</sup>. These related parameters are design specific as each FPGA design utilizes different logic, routing, and other FPGA resources. These parameters may vary widely from design to design, as some designs are very dense and utilize most FPGA resources, while others consume few FPGA resources. An essential part of this reliability model is accurately estimating the parameter  $Q_s$ .

$Q_s$  is estimated by computing  $Q_{s,i}$  for multiple values of  $i$ .  $Q_{s,i}$  is the joint probability that the circuit fails during a single scrub cycle *and*  $i$  upsets occur.  $Q_{s,i}$  is computed with the following equation:

$$Q_{s,i} = P(F_s \cap A_i) = P(F_s|A_i)P(A_i), \quad (1)$$

where event  $F_s$  is a design failure during a *single* scrub cycle and event  $A_i$  is  $i$  SEUs during a *single* scrub cycle. For example, the probability of one SEU *and* of design failure during a single scrub cycle can be computed as follows:

$$Q_{s,i=1} = P(F_s \cap A_1) = P(F_s|A_1)P(A_1).$$

Computing  $Q_{s,i}$  with (1) requires  $P(F_s|A_i)$ , the conditional probability of failure given  $i$  upsets. This can be estimated for various values of  $i$  using fault injection or accelerator experiments. Section V will describe how these results were obtained for several FPGA designs. Equation (1) also requires  $P(A_i)$ , the probability of  $i$  SEUs within a scrubbing period. This probability is computed as a Poisson distribution of the upset rate of a specific orbit/condition (see Section IV).

<sup>2</sup>The model presented in this paper is primarily based on the probability of failure rather than reliability. The design unreliability parameter,  $Q_s$ , will be used instead of  $R_s$ .

To determine the unconditional probability of failure during a single scrub cycle,  $Q_s$ , the joint probabilities of failure for all  $i$  are summed using the equation:

$$Q_s = Q_{s,i \geq 0} = \sum_{i=0}^{\infty} Q_{s,i} = \sum_{i=0}^{\infty} P(F_s|A_i)P(A_i). \quad (2)$$

Once the model parameter  $Q_s$  is known, the failure rate  $\lambda$  (failures in time) of the circuit can be estimated as follows:

$$\lambda = \frac{Q_s}{t_s}, \quad (3)$$

where  $t_s$  is the period of a single scrub cycle. The mean time to failure,  $MTTF$ , is calculated from  $\lambda$  as follows:

$$MTTF = \frac{1}{\lambda} = \frac{t_s}{Q_s}. \quad (4)$$

Because an FPGA system will operate in a variety of orbit conditions, it is helpful to estimate a composite failure rate ( $\lambda_c$ ) and composite mean time to failure ( $MTTF_c$ ) which incorporates the failure rate during each orbit condition and the probability of operating in that orbit condition. A composite, single parameter failure rate ( $\lambda_c$ ) can be calculated for an interval that spans multiple orbit conditions by obtaining the failure rate during each orbit condition,  $\lambda_i$ , and estimating the probability of being in that orbit condition,  $\rho_i$ :

$$\lambda_c = \rho_1\lambda_1 + \rho_2\lambda_2 + \dots + \rho_n\lambda_n \quad (5)$$

where

$$\sum_{i=1}^n \rho_i = 1.$$

Since the FPGA must always operate in one of the  $n$  orbit conditions, the sum of all  $\rho$  must be one. A composite  $MTTF$  can be obtained by applying (4),

$$MTTF_c = \frac{1}{\lambda_c} = \frac{1}{\rho_1\lambda_1 + \rho_2\lambda_2 + \dots + \rho_n\lambda_n}. \quad (6)$$

Section VI-B will propose values for  $\rho_i$  for several orbit conditions during a solar cycle (e.g., solar min, solar max, worst week, worst day, and peak five minutes).

#### IV. ESTIMATING UPSETS PER SCRUB CYCLE, $P(A_i)$

The first parameter needed to determine  $Q_s$  in (2) is the probability that  $i$  upsets will occur during a single scrub cycle ( $P(A_i)$ ). This can be modeled with a Poisson distribution,

$$P(A_i) = e^{-\nu} \frac{\nu^i}{i!}, \quad (7)$$

TABLE I  
CREME96 ORBIT-AVERAGED STATIC SEU RATES

Orbit	Altitude (km) Inclination	Solar Max	Worst Week	Worst Day	Peak 5-Min.
		$\mu$ (SEUs/Device/s)			
GEO	35,786 0°	1.6E-5	1.7E-2	8.8E-2	3.3E-1
GPS	20,200 55°	1.6E-5	1.5E-2	7.6E-2	2.9E-1
Mol.	39,305/1,507 63.2°	7.9E-5	1.6E-2	8.2E-2	3.1E-1
Polar	833 98.7°	5.9E-5	3.5E-3	2.1E-2	7.8E-2
LEO	560 35.0°	2.5E-5	1.5E-6	1.1E-6	4.0E-6

where  $\nu$  is the average number of SEUs per scrub period. The parameter  $\nu$  is calculated by multiplying the orbit-averaged upset rate (SEUs per time),  $\mu$ , by the scrub period,  $t_s$ , as follows:

$$\nu = \mu \times t_s. \quad (8)$$

The parameter  $\mu$  can be estimated using modeling tools such as CREME96 [7]. CREME96 requires static cross section data for a particular device. In this work, we assume the use of a Xilinx Virtex-4 XQR4VXSX55 FPGA and use its static cross section data obtained from the Xilinx Radiation Test Consortium (XRTC) [20].

For this work, we will focus on the following four “harsh” orbits: geosynchronous (GEO), global positioning system (GPS), Molniya, and Polar. In addition, we will include a low Earth orbit (LEO) as a point of reference. We will focus primarily on SEP events since these conditions represent the harshest radiation environment in Earth’s orbit. SEU rates are estimated in each orbit for the worst week, worst day, and peak five minutes of an SEP event. SEU rates are also estimated for normal solar max conditions as a point of reference. The average SEU rate for each of these orbits and solar conditions is shown in Table I.

The orbit-averaged static SEU rate,  $\mu$ , and scrub period,  $t_s$ , are used to compute  $\nu$  using (8). For example, the XQR4VXSX55 Xilinx FPGA in a GEO orbit during normal solar max conditions should upset 1.6E-5 times per second. Assuming a scrub period of  $t_s = 15 \text{ ms}^3$ , then

$$\begin{aligned} \nu &= \mu \times t_s \\ &= 1.6E-5 \frac{\text{upsets}}{\text{s}} \times .015 \frac{\text{s}}{\text{scrub}} \\ &= 2.4E-7 \frac{\text{upsets}}{\text{scrub}}. \end{aligned}$$

Once  $\nu$  is known, (7) can be used to find the values of  $P(A_i)$ . Fig. 2 plots  $P(A_i)$  for a Xilinx Virtex-4 XQR4VXSX55 FPGA in

<sup>3</sup>The scrub rate used for our Xilinx Virtex-4 XQR4VXSX55 FPGA system is 15 ms. For the purpose of this paper we will always assume  $t_s = .015 \text{ s}$ .



Fig. 2. Probability of  $i$  upsets per scrub cycle for a Xilinx Virtex-4 XQR4V5SX55 FPGA in solar max conditions in GEO with a 15 ms scrub period.

TABLE II  
TEST DESIGN UTILIZATION

Test Design	Mitigation	Logic Slices	LUTs	FFs
BYU Shift Register	None	4011 (16%)	7950 (16%)	8004 (16%)
	TMR	12014 (48%)	23823 (48%)	24006 (48%)
SSRA	None	5381 (21%)	6294 (12%)	8832 (17%)
	TMR	18591 (75%)	19388 (39%)	26490 (53%)
Shift Reg 1B	None	8111 (33%)	16212 (32%)	16204 (32%)
	TMR	24314 (98%)	48609 (98%)	48606 (98%)

solar max conditions in GEO with a 15 ms scrub period. Note that  $P(A_i)$  decreases rapidly as  $i$  increases— $P(A_i)$  can be considered zero for  $i > 100$ .

## V. ESTIMATING PROBABILITY OF DESIGN FAILURE, $P(F_s|A_i)$

The second parameter needed to complete our model from (2) is  $P(F_s|A_i)$ , the conditional probability of design failure during one scrub cycle given  $i$  SEUs occurred during that scrub cycle. This parameter is design specific and must be estimated for each FPGA design that is to be considered. A significant part of this work was estimating this parameter using both fault injection experiments and accelerator testing for several designs. This section will summarize the test equipment, designs, and methodology used to estimate  $P(F_s|A_i)$ .

Tests were performed on three designs to estimate  $P(F_s|A_i)$ . The first design is a 32-bit wide, 250-bit deep shift register, with arbitrary combinational logic between each stage (BYU Shift Register). The second design is a digital signal processing kernel, with a polyphase filter bank to separate the data into 32 channels, followed by an FFT and a magnitude operation on each of the channels (SSRA). The third design is a 1-bit wide, 16,200-stage deep shift register (Shift Register 1B). A TMR version of each design was created to test the effectiveness of TMR mitigation. The FPGA resource utilization of both variations of all three designs is reported in Table II.

Each circuit was designed to operate on the Xilinx Virtex-4 XQR4V5SX55 FPGA within the XRTC test fixture (see Fig. 3). A configuration monitor FPGA and functional monitor FPGA are also available on this board to manage the device configuration and scrubbing, provide test patterns, and monitor circuit output. The XRTC test fixture was designed to support both fault injection and radiation testing.

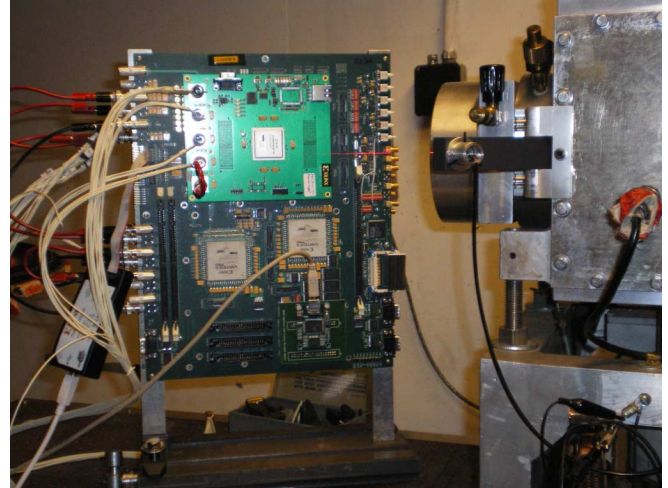


Fig. 3. The Xilinx Radiation Test Consortium Test Fixture.

```

do {
  generate 'i' random bits to upset
  inject 'i' upsets into bitstream
  check for output error
  fix upset bits
  reset device
  record data to output file
} while number of trials < 'm'

```

Fig. 4. Fault-injection algorithm.

### A. Fault Injection

Fault injection has been used extensively in the past to estimate the sensitivity of an FPGA design to configuration upsets [21], [22]. In prior work, we used fault injection to estimate  $P(F_s|A_1)$  or the probability that *one* configuration upset will cause a design to fail during a single scrub cycle.

A modified fault injection algorithm, shown in Fig. 4, was used for this work to estimate  $P(F_s|A_{i>1})$  or the probability that *more than one* configuration upset will cause a design to fail during a single scrub cycle. The algorithm begins by selecting  $i$  random bits from the configuration bitstream of the design under test (DUT). Each of these bits is toggled and injected back into the bitstream to emulate  $i$  SEUs in the bitstream. The output signals of the DUT are compared against a golden copy of the circuit to check for circuit errors. If a disparity exists between the output signals of the DUT and the output signals of the golden design, then a failure event is recorded. This process is repeated multiple times to estimate  $P(F_s|A_i)$  or the probability that  $i$  configuration upsets will cause a design to fail during a single scrub cycle.

$P(F_s|A_i)$  is estimated by dividing the number of output errors identified during the test by the number of trials performed ( $m$ ). During the course of testing,  $i$  was tested at 1–10, 20, 40, 60, 80, and 100 upsets.

Fault-injection testing was performed for non-TMR and TMR versions of all three example circuits. The values of  $P(F_s|A_i)$  obtained during fault injection for the SSRA design are shown in Fig. 5. Non-TMR and TMR data are plotted on the same graph for comparison. The x-axis corresponds to  $i$  upsets in the system for a given trial. The y-axis represents the percentage of trials

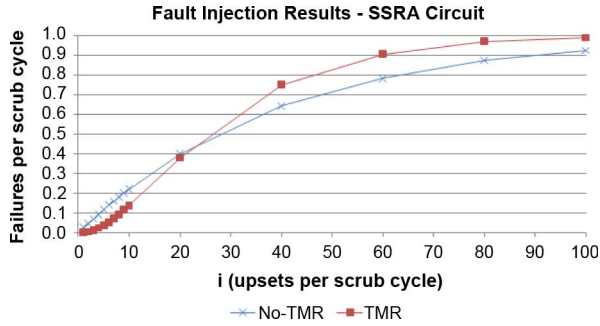


Fig. 5. Fault injection results for both the non-TMR and TMR versions of the SSRA example circuit showing the conditional probability of failure for a single scrubbing cycle given  $i$  upsets occurred during the scrubbing cycle.

```

do {
    sleep for time 't'
    get and fix upset frames
    if upsets found
        record upsets to file
        check for output error
    end if
    reset DUT
} while time < 'T'
    
```

Fig. 6. Accelerator algorithm.

(scrub cycles) that failed, or in other words,  $P(F_s|A_i)$ , the conditional probability of failure for a single scrubbing cycle given  $i$  random upsets in the system. For example, on the non-TMR plot of the SSRA design in Fig. 5, at  $i = 20$  upsets per scrub on the DUT, the failure rate is just above 40%, or in other words,  $P(F_s|A_{20}) = 0.4$ .

### B. Accelerator Testing

An alternative way of estimating  $P(F_s|A_i)$  is through radiation testing. We performed accelerator experiments at the cyclotron located at Crocker Nuclear Laboratory in Davis, CA to determine the probability of design failure with  $i$  upsets during single scrub cycle. Although the same XRTC test fixture was used, a slightly different test algorithm was developed for use at the accelerator (see Fig. 6). For each cycle of the loop, the control software pauses execution for time  $t$  to allow upsets to accumulate on the DUT. Frame readback is used to count SEUs, which are then recorded and repaired. If SEUs are found, the device is checked for output errors and the results are recorded. The algorithm continues until the trial length of time  $T$  is complete.

Due to time and cost restraints, only the non-TMR and TMR versions of SSRA were tested at the accelerator. The results for the non-TMR version are shown in Fig. 7, and the results from the TMR version are shown in Fig. 8. The accelerator data are shown with error-bars, and the fault-injection data are plotted on the same graph for comparison. The size of the error bars is calculated as

$$err = \pm \frac{\sqrt{failures}}{trials}. \quad (9)$$

There is a slight discrepancy between the results from accelerator testing and fault injection for the non-TMR design (see

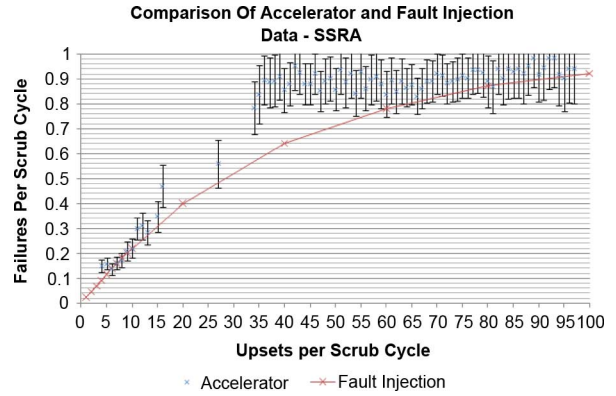


Fig. 7. The results from accelerator testing on the non-TMR version of the SSRA design. The accelerator data is shown with error bars, and fault-injection data is overlaid for comparison.

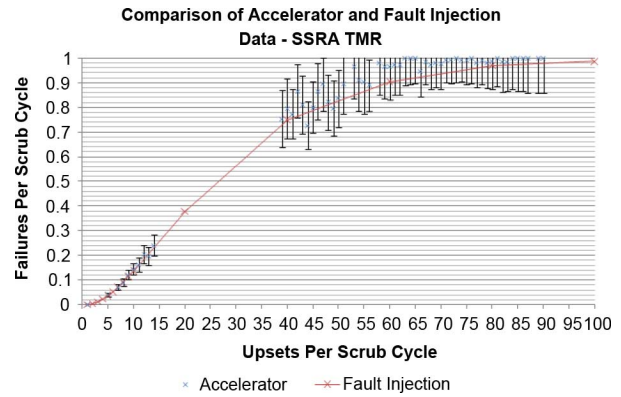


Fig. 8. The results from accelerator testing on the TMR version of the SSRA design, with overlaid fault-injection data.

Fig. 7). This discrepancy is due to the inability to prevent the beam from upsetting the device during book keeping activities. This discrepancy was not seen on the TMR design.

## VI. ESTIMATING CIRCUIT RELIABILITY

Once the joint probability of failure ( $P(F|A_i)$ ) has been obtained from fault injection or radiation testing and the probability of  $i$  upsets ( $P(A_i)$ ) has been obtained from an orbit-specific upset rate (see (7)), a circuit's reliability (or unreliability) can be estimated. To illustrate how to this is done, we will calculate  $Q_s$ ,  $MTTF$ ,  $\lambda_c$  and  $MTTF_c$  for a specific example. For this example we have selected the SSRA TMR design in a Xilinx Virtex-4 XQR4V5X55 FPGA. The period of the scrubber is 15 milliseconds. The system is in a GEO orbit. The tables in this section will also report results for various designs, orbits and orbit conditions.

### A. Estimating $Q_s$

Values of  $Q_{s,i}$  for the SSRA TMR design GEO during a peak five minute SEP event are shown in Fig. 9 for  $i = 1$  to  $i = 5$ . This plot also includes  $P(F|A_i)$  and  $P(A_i)$ . Larger values of  $i$  are not included as the conditional probability  $P(F|A_i)$  that the design will fail is high for large values of  $i$ . This is intuitive as more upsets in a scrub cycle are more likely to break the design. However, for large values of  $i$ , the probability  $P(A_i)$  of  $i$  upsets actually occurring in a single scrub cycle is very low.

TABLE III  
 $MTTF$  OF THE SSRA TMR DESIGN FOR ALL ORBITS & CONDITIONS

Orbit	Solar Max		Worst Week		Worst Day		Peak 5 Minutes	
	$Q_s$	$MTTF$ (s)	$Q_s$	$MTTF$ (s)	$Q_s$	$MTTF$ (s)	$Q_s$	$MTTF$ (s)
GEO	$1.7E-13$	$8.9E10$ (2810 yrs)	$3.7E-10$	$4.1E7$ (1.3 yrs)	$6.2E-9$	$2.4E6$ (28 days)	$7.8E-8$	$1.9E5$ (2.2 days)
GPS	$1.7E-13$	$9.1E10$ (2880 yrs)	$3.0E-10$	$4.9E7$ (1.6 yrs)	$4.8E-9$	$3.1E6$ (36 days)	$5.9E-8$	$2.6E5$ (3.0 days)
Molniya	$8.4E-13$	$1.8E10$ (566 yrs)	$3.4E-10$	$4.5E7$ (1.4 yrs)	$5.5E-9$	$2.7E6$ (32 days)	$6.8E-8$	$2.2E5$ (2.6 days)
Polar	$6.3E-13$	$2.4E10$ (753 yrs)	$4.6E-11$	$3.3E8$ (10 yrs)	$5.2E-10$	$2.9E7$ (333 days)	$5.0E-9$	$3.0E6$ (35 days)
LEO	$2.7E-13$	$5.6E10$ (1790 yrs)	$1.5E-14$	$9.7E11$ (3E4 yrs)	$1.1E-14$	$1.3E12$ (4E4 yrs)	$4.2E-14$	$3.5E11$ (1E4 yrs)

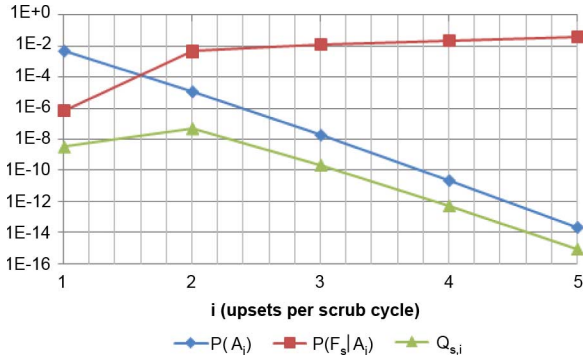


Fig. 9. Plot of  $P(A_i)$ ,  $P(F_s|A_i)$ , and  $Q_{s,i}$  for the SSRA TMR design during peak 5 minute conditions in GEO.

TABLE IV  
 PROBABILITY OF GEO ORBIT CONDITIONS WITHIN AN 11 YEAR SOLAR CYCLE AND COMPOSITE FAILURE RATE FOR THE SSRA-TMR DESIGN

Condition	Time (s)	$\rho$	$\lambda$	$\rho\lambda$
Solar Max	$2.97E8$	.856	$1.1E-11$	$9.7E-12$
Worst Week	$4.49E7$	.129	$2.8E-8$	$3.7E-9$
Worst Day	$4.97E6$	.014	$5.2E-7$	$7.5E-9$
Peak 5 min	$2.31E4$	.000067	$6.7E-6$	$4.4E-10$
Composite	$3.5E8$	1.00	$\lambda_c = 1.4E-8$	

Consequently  $Q_{s,i}$  rapidly decreases with increasing values of  $i$ .

Using (2), we estimate  $Q_s$  for the SSRA TMR design in GEO during a peak five minute event as

$$Q_s \approx \sum_{i=0}^{100} Q_{s,i} = \sum_{i=0}^{100} P(F_s|A_i)P(A_i) = 7.8E-8$$

where we have taken advantage of the fact mentioned in Section IV that  $P(A_i)$  is approximately zero for large  $i$  to only sum from  $i = 1$  to  $i = 100$ .

The largest contributor to  $Q_s$  is  $Q_{s,i=2}$ , when two upsets occur in a scrub cycle ( $i = 2$ ). For the SSRA TMR design, the probability that two upsets causes a design failure is  $P(F_s|A_2) = 4.4E-3$  and the probability that two upsets will occur in a scrub period during the peak five minutes of the GEO orbit is  $P(A_2) = 1.2E-5$ .  $Q_{s,i=2}$  is then

$$\begin{aligned} Q_{s,i=2} &= P(F_s|A_2)P(A_2) \\ &= (4.4E-3)(1.2E-5) \\ &= 5.3E-8. \end{aligned}$$

In this example scenario  $Q_{s,i=2}$  accounts for 68% of  $Q_s$ .

It is interesting to note that the second largest contributor to the probability of failure is single upsets in a scrub cycle ( $i = 1$ ). This seems counter intuitive as the use of TMR is supposed to mitigate against *all* single event upsets. However, results from both fault injection and accelerator testing suggest that there are a few configuration bits that do indeed cause the design to fail<sup>4</sup> [23]. One obvious way to improve the reliability would be to apply known design and implementation techniques that remove single point failures from the design [23].

The probability of more than two upsets ( $i > 2$ ) during a scrub cycle in GEO orbit is very low and thus individually these events contribute very little to  $Q_s$ .

### B. Estimating $MTTF$

The probability of failure during a scrub cycle,  $Q_s$ , was computed for the SSRA TMR design in five orbits under four orbit conditions.  $MTTF$  was also computed by applying  $Q_s$  and  $t_s$  to (4). Table III lists  $Q_s$  and  $MTTF$  of the SSRA TMR design for all combinations of orbits and conditions. For normal solar max conditions in GEO, the design is quite reliable with a  $MTTF$  of over 2000 years. For the peak 5 minute event<sup>5</sup> in GEO, the  $MTTF$  is five orders of magnitude lower and drops to 2.2 days. Although the  $MTTF$  for the peak 5 minute case is quite low, there is a relatively high probability that the design will operate without failure during the 5 minute event.

### C. Estimating Composite $MTTF$

A composite failure rate for an orbit can be determined by using the estimated  $MTTF$  for each orbit condition and applying these results to (5). For this work, the probabilities of operating in each orbit condition,  $\rho_i$ , were obtained by estimating the amount of time spent in each orbit condition during a solar cycle. We use the following four conditions: solar max, worst week, worst day, and peak five minutes. The amount of time estimated in each orbit condition is summarized below.

1) *Peak Five Minutes*: The CREME96 peak five minute flux model is based on the peak five-minute averaged fluxes observed on GOES in October 1989 [25]. We assume that each SEP event results in one peak five minute orbit condition (300 seconds).

<sup>4</sup>Xilinx Inc. has identified a solution to this problem. We were not able to implement this solution for the experiments in this paper.

<sup>5</sup>According to the CREME96 website, direct measurements of the high-energy heavy-ion fluxes are actually not possible on such a short time scale. As a result the peak five minute heavy-ion fluxes are scaled from the "worst-day" fluxes, using the energy-dependent peak-to-"worst-day" ratios derived from the GOES protons [24]. We make the assumption that the flux of solar energetic particles is uniformly distributed throughout the five minute event. If this assumption is not true and the flux follows a non-uniform distribution, then the reliability of the FPGA system could be lower. We have no data to suggest another distribution. Additional insight is needed into the radiation environment associated with the peak five minute event to better estimate system reliability during these extreme events.

TABLE V  
COMPOSITE  $MTTF_c$  FOR ALL DESIGNS IN ALL ORBITS

Orbit	Test Design					
	SSRA-TMR	SSRA-No TMR	BYU SR-TMR	BYU SR-No TMR	Reg1B-TMR	Reg1B-No TMR
GEO	1.1E8 (3.4 yrs)	1.2E4 (3.2 hrs)	1.5E8(4.8 yrs)	1.7E4 (4.8 hrs)	4.5E7 (1.4 yrs)	9.7E3 (2.7 hrs)
GPS	1.3E8 (4.3 yrs)	1.3E4 (3.7 hrs)	1.9E8 (6.0 yrs)	2.0E4 (5.4 hrs)	5.4E7 (1.7 yrs)	1.1E4 (3.1 hrs)
Molniya	1.2E8 (3.7 yrs)	1.2E4 (3.4 hrs)	1.7E8 (5.3 yrs)	1.8E4 (5.0 hrs)	4.9E7 (1.5 yrs)	1.0E4 (2.8 hrs)
Polar	1.1E9 (33.3 yrs)	4.9E4 (14 hrs)	1.3E9 (40 yrs)	7.3E4 (20 hrs)	2.9E8 (9.2 yrs)	4.1E4 (1.2 hrs)
LEO	6.5E10 (2100 yrs)	1.9E6 (22 days)	6.5E10 (2100 yrs)	2.7E6 (76 hrs)	1.3E10 (409 yrs)	1.6E6 (18 days)

2) *Worst Day*: The CREME96 worst-day model is based on SEP fluxes averaged over 18 hours beginning at 1300 UT on 20 October 1989. This period was the single largest flux enhancement in October 1989 [24]. We assume that each SEP results in one worst-day orbit condition for 18 hours minus the five minutes spent in a peak five minute condition (6.5E4 seconds).

3) *Worst Week*: The CREME96 worst-week model is based on SEP fluxes averaged over 180 hours (7.5 days) beginning at 1300 UT on 19 October 1989. This week was the most severe SEP environment observed in the last two solar maxima [24]. We assume that each SEP results in the worst week orbit condition for 7.5 days minus the time spent in a worst day and peak five minute condition (5.8E5 seconds).

4) *Solar Max*: We assume the remainder of the time is normal, solar max conditions. For the purpose of this model we do not distinguish between solar min and solar max conditions as their flux levels are orders of magnitude lower than the flare-enhanced conditions.

For the purposes of this paper we make the pessimistic assumption that there are seven SEP events<sup>6</sup> per year regardless of position in the solar cycle for a total of 77 SEP events during an 11-year solar cycle [25]. We also pessimistically assume that each SEP event results in a worst week, worst day, and peak five minute flux. In other words, we make the very pessimistic assumption that every SEP event is as bad as the October 1989 event.

The time spent in each orbit condition during an 11-year solar cycle is listed in Table IV. The total time is 3.5E8 seconds, the number of seconds in 11 years. The probability of operating in each of the four orbit conditions,  $\rho_i$ , is determined by dividing the amount of time spent in each orbit condition by the time in a full solar cycle. Assuming seven SEP events per year, 85.6% of the time involves normal conditions, 12.9% of the time involves worst week conditions, 1.4% of the time involves worst day conditions, and a very small amount of time is spent in the worst five minute peak conditions.

The failure rate for each orbit condition was calculated with (3) using the values of  $Q_s$  listed in Table III and a scrub rate of  $t_s = .015$ . Table IV demonstrates the computation of a composite failure rate,  $\lambda_c$ , for the SSRA TMR design in GEO. The composite failure rate is less than half the failure rate of the worst week but almost three orders of magnitude larger than the normal, solar max conditions.

The composite failure rate,  $\lambda_c$ , is used to compute a composite  $MTTF$  for a design over all orbit conditions using (6). For example, using the results in Table IV, the composite  $MTTF$  of the SSRA-TMR design in the GEO orbit is 1.1E8

<sup>6</sup>Each SEP event is assumed to produce  $10^6$  protons with energy greater than 30 MeV [25].

seconds. In other words, the mean time to failure of the SSRA circuit protected by TMR and scrubbing in the GEO orbit is 3.4 years.

This same composite reliability analysis was performed for all six designs listed in Table II. The results from this analysis are summarized in Table V. The data provide several important insights into the reliability of FPGA designs. First, the data highlights the importance of TMR and scrubbing. The  $MTTF$  of the TMR designs is over four orders of magnitude greater than the corresponding non-TMR designs. The ability to tolerate single bit upsets and the frequent scrubbing of these upsets significantly improves the reliability of the design. Second, the data highlight the difference in reliability for the same design in different orbits. As expected, the designs are much more reliable in the LEO orbit than the other “harsh” orbits. SEP events in the “harsh” orbits generate far more coincident configuration upsets that break TMR than in the LEO orbit. Third, the data highlight the difference in reliability for different designs in the same orbit conditions. Fault injection and radiation testing both confirm that each FPGA design has a unique sensitivity to SEUs. The results clearly indicate that the probability of failure is design dependent and that design-specific fault injection should be performed to estimate design reliability.

## VII. CONCLUSION

This paper presents a reliability model for estimating the  $MTTF$  of SRAM FPGA designs in specific orbits and orbit conditions. The use of this model requires orbit- and condition- specific SEU rate estimates for the FPGA family ( $\mu$ ), probability of failure estimates for each design ( $Q_{s,i}$ ), and the probability of operating in each orbit condition ( $\rho_i$ ). This model was applied to six different FPGA designs (three that use TMR and three that do not) and five different orbits. Four orbit conditions were considered for the composite  $MTTF$  including: solar max, worst week, worst day, and peak five minutes.

The results from this model suggest that with TMR and scrubbing, SRAM FPGA designs operate very reliably in a LEO orbit and surprisingly well in “harsh” orbits. While the long  $MTTF$  estimates for the LEO orbit were not surprising, we expected to see much shorter  $MTTF$  estimates for GEO and other “harsh” orbits. While these reliability estimates suggest that SRAM FPGAs are not appropriate for all situations, they may be used in many circumstances.

The model presented in this paper makes pessimistic assumptions that negatively impact the  $MTTF$  estimates. This model assumes that 77 SEP events will occur in a solar cycle and each SEP event includes the worst week flux for 7.5 days, the worst day flux for 18 hours, and worst five minute flux for five minutes. This is very pessimistic and unnecessarily biases our reli-

ability results in a negative way. Without more reliable models on average SEP events, however, it is difficult to estimate a composite reliability model. More accurate models of average SEP events and more accurate flux distributions during these events will improve our model and most likely significantly raise our *MTTF* estimates.

The fault injection results used to estimate design reliability exposed several single-point failures in our TMR designs. These single point failures negatively impacted our reliability estimates more than we expected. The reliability of these test designs can be dramatically improved by resolving these single-point failures using known FPGA design and implementation techniques. Further, several variations on TMR have been created and tested that improve the reliability of FPGA designs in the presence of coincident upsets using more frequent voting [26]. The use of these techniques will likely significantly improve design reliability in harsh environments as these techniques mitigate against a large number of coincident upsets. Finally, reliability could also be improved by shrinking the length of the scrub cycle  $t_s$  to reduce the probability of more than one upset per scrub cycle, specifically  $P(A_2)$ .

#### REFERENCES

- [1] D. Weigand and M. Harlacher, "A radiation-tolerant low-power transceiver design for reconfigurable applications," in *Proc. Earth Science Technology Conf. (ESTC)*, 2002, p. Paper A1P2 [Online]. Available: [http://esto.nasa.gov/conferences/estc-2002/Papers/A1P2\(Weigand\).pdf](http://esto.nasa.gov/conferences/estc-2002/Papers/A1P2(Weigand).pdf)
- [2] K. Morris, *FPGAs in Space*, Tech Focus Media, FPGA and Structured ASIC Journal 2004.
- [3] D. Ratter, *FPGAs on Mars*, Xilinx, xCell Journal #50 2004.
- [4] M. Caffrey, T. P. Plaks and P. M. Athanas, Eds., "A space-based reconfigurable radio," in *Proc. Int. Conf. on Engineering of Reconfigurable Systems and Algorithms (ERSA)*, Jun. 2002, pp. 49–53.
- [5] M. Caffrey, K. Morgan, D. Roussel-Dupre, S. Robinson, A. Nelson, A. Salazar, M. Wirthlin, W. Howes, and D. Richins, "On-orbit flight results from the reconfigurable cibola flight experiment satellite (CFESat)," presented at the Proc. IEEE Symp. on FPGAs for Custom Computing Machines (FCCM'03), Napa, CA, Apr. 5–7, 2009.
- [6] E. Fuller, M. Caffrey, P. Blain, C. Carmichael, N. Khalsa, and A. Salazar, "Radiation test results of the Virtex FPGA and ZBT SRAM for space based reconfigurable computing," presented at the 1999 Military and Aerospace Applications of Programmable Logic Devices (MAPLD, Laurel, MD, Sep. 1999.
- [7] A. J. Tylka, J. H. Adams, Jr, P. R. Boberg, B. Brownstein, W. F. Dietrich, E. O. Flueckiger, E. L. Petersen, M. A. Shea, D. F. Smart, and E. C. Smith, "CREME96: A revision of the cosmic ray effects on micro-electronics code," *IEEE Trans. Nucl. Sci.*, vol. 44, pp. 2150–2160, Dec. 1997.
- [8] Xilinx TMRTool, Product Brief Xilinx Corporation, 2006.
- [9] B. Pratt, M. Caffrey, J. Carroll, P. Graham, K. Morgan, and M. Wirthlin, "Fine-grain SEU mitigation for FPGAs using partial TMR," *IEEE Trans. Nucl. Sci.*, vol. 55, pp. 2274–2280, Aug. 2008.
- [10] C. Carmichael, Triple Module Redundancy Design Techniques for Virtex FPGAs Xilinx Corporation, 2001, xAPP197 (v1.0).
- [11] B. Pratt, M. Caffrey, P. Graham, E. Johnson, K. Morgan, and M. Wirthlin, "Improving FPGA design robustness with partial TMR," presented at the Proc. IRPS Conf., Mar. 2006.
- [12] J. Heiner, B. Sellers, M. Wirthlin, and J. Kalb, "FPGA partial reconfiguration via configuration scrubbing," presented at the 11th Int. Workshop, Field-Programmable Logic and Applications and Lecture Notes in Computer Sci., Aug. 2009, LNCS 2438.
- [13] M. Berg, "The NASA Goddard space flight center radiation effects and analysis group Virtex 4 scrubber," presented at the Xilinx Radiation Test Consortium (XRTC) Meeting, 2007.
- [14] K. Morgan, D. McMurtrey, B. Pratt, and M. Wirthlin, "A comparison of TMR with alternative fault-tolerant design techniques for FPGAs," *IEEE Trans. Nucl. Sci.*, vol. 54, 2007.
- [15] D. McMurtrey, K. Morgan, B. Pratt, and M. Wirthlin, Estimating TMR Reliability on FPGAs Using Markov Models Brigham Young University, 2006 [Online]. Available: <http://dspace.byu.edu>
- [16] H. Quinn, P. Graham, J. Krone, M. Caffrey, and S. Rezgui, "Radiation-induced multi-bit upsets in SRAM-based FPGAs," *IEEE Trans. Nucl. Sci.*, vol. 52, pp. 2455–2461, Dec. 2005.
- [17] H. Quinn, K. Morgan, P. Graham, J. Krone, M. Caffrey, and K. Lundgreen, "Domain crossing errors: Limitations on single device triple-modular redundancy circuits in Xilinx FPGAs," *IEEE Trans. Nucl. Sci.*, vol. 54, no. 6, pp. 2037–2043, Dec. 2007.
- [18] O. Heron, T. Arnaout, and H.-J. Wunderlich, "On the reliability evaluation of SRAM-based FPGA designs," in *Proc. Int. Conf. on Field Programmable Logic and Applications*, 2005, pp. 403–408.
- [19] L. D. Edmonds, Analysis of Single-Event Upset Rates in Triple-Modular Redundancy Devices NASA Jet Propulsion Laboratory, 2009 [Online]. Available: <http://trs-new.jpl.nasa.gov/dspace/bitstream/2014/4/11231/09-6.pdf>
- [20] G. Allen, G. Swift, and C. Carmichael, Virtex-4qv Static SEU Characterization Summary Jet Propulsion Laboratory, Pasadena, CA, 2008, JPL Publication 08-16 4/08.
- [21] E. Johnson, M. Caffrey, P. Graham, N. Rollins, and M. Wirthlin, "Accelerator validation of an FPGA SEU simulator," *IEEE Trans. Nucl. Sci.*, vol. 50, no. 6, pp. 2147–2157, 2003.
- [22] M. J. Wirthlin, D. E. Johnson, N. H. Rollins, M. P. Caffrey, and P. S. Graham, K. L. Pocek and J. M. Arnold, Eds., "The reliability of FPGA circuit designs in the presence of radiation induced configuration upsets," in *Proc. IEEE Symp. on FPGAs for Custom Computing Machines (FCCM'03)*, Apr. 2003, pp. 133–142.
- [23] M. Violante and L. Sterone, "A new reliability-oriented place and route algorithm for SRAM-based FPGAs," *IEEE Trans. Comput.*, vol. 55, pp. 732–744, Jun. 2006.
- [24] [Online]. Available: <https://creme96.nrl.navy.mil/>
- [25] R. A. Nymmik, "Relationships among solar activity, SEP occurrence frequency, and solar energetic particle event distribution function," presented at the 25th ICRC, 1999, Paper SH.1.5.16.
- [26] B. Pratt, M. Caffrey, D. Gibelyou, P. Graham, K. Morgan, and M. Wirthlin, T. P. Plaks and P. M. Athanas, Eds., "TMR with more frequent voting for improved FPGA reliability," in *Proc. Int. Conf. on Engineering of Reconfigurable Systems and Algorithms (ERSA)*, Jul. 2008, pp. 153–158.