LA-UR-05-0177

Approved for public release;
distribution is unlimited.

Title: Terrestrial-Based Radiation Upsets: A Cautionary Tale

Author(s): Heather M. Quinn, ISR-3
Paul S. Graham, ISR-3

Submitted to: IEEE Symposium on Field-Programmable Custom Computing Machines, April 17 - April 20, 2005, Napa, California

Los Alamos
NATIONAL LABORATORY

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the University of California for the U.S. Department of Energy under contract W-7405-ENG-36. By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

Form 836 (8/00)

Terrestrial-based Radiation Upsets: A Cautionary Tale

Heather Quinn, Paul Graham
Los Alamos National Laboratory
Los Alamos, NM 87544
hquinn, grahamp@lanl.gov

Abstract

Problems with terrestrial-based neutron radiation from cosmic rays have become more commonplace. While the incident rate from neutron radiation is lower than space-based radiation, physics, system design and system locations have combined to make systems increasingly vulnerable to terrestrial radiation. FPGA systems are particularly sensitive to neutron radiation, as the FPGAs, microprocessors and memory are all sensitive to upsets. We are interested in reconfigurable supercomputers,which need to be highly reliable and highly available despite being very sensitive to radiation. In this paper, we estimate the error rate for FPGAs, memory, and microprocessors so that predictions for the sensitivity of the Cray XD1 reconfigurable supercomputer can be made. We also present possible mitigation methods that are appropriate for neutron radiation upset rates.

1 Introduction

Mitigating neutron radiation effects is often overlooked when building large scale, terrestrial-based computer systems. In recent years, several cases of cosmic ray upsets (*soft errors*) in microprocessor systems have been reported [16, 15]. The Q cluster and System X, which were the second and third fastest supercomputers on the November 2003 Top 500 Supercomputer Sites list [1], both experience fatal soft errors presumably caused by cosmic ray induced neutrons. System X was tested during a Coronal Mass Ejection, which led one architect to joke that they "felt like [they] had not only built the world's third fastest supercomputer, but also one of the world's best cosmic ray detectors [14]." Over a 10 week period from late April 2004 through early July 2004, the Q cluster experienced an average 26.1 CPU failures per week [16]. Fatal soft errors deriving from cosmic ray induced neutrons are one example of CPU failures, so this number provides an upper bound for the average weekly number of fatal soft errors experienced by Q over the same period.

Unfortunately, SRAM-based FPGA systems are not exempt from neutron radiation problems. All aspects of FPGA systems are vulnerable to terrestrial-based radiation. Besides the microprocessors, the FPGAs and memory are affected. Memory upsets are the most common side effect of radiation interference. SRAM-based FPGA systems are susceptible to both soft error induced state and configuration changes. Both Xilinx and Altera FPGAs exhibit the ability to upset in terrestrial-based systems [11, 6]. Even ECC-protected memory has issues with neutron radiation [3]. All of these components need to be analyzed to determine the *total cross-section* (the area sensitive to neutron upsets). We are interested in highly available and reliable reconfigurable supercomputers, which might have thousands of processors, thousands of FPGAs, and many gigabytes of RAM. These systems have very large cross-sections and have noticeable *soft error rates* (SERs). This paper focuses on estimating the cross-sections for memory, FPGAs and microprocessors, so that the cross-section for large scale systems can be analyzed.

Many designers assumed that with shrinking transistor size and better manufacturing processes that soft errors should become less prevalent, but the opposite has been true. The increase can be explained by a combination of three factors: physics, system design, and system location. First, there are physical issues when the transistors shrinks. Smaller transistors are more sensitive to smaller changes in charge, so shrinking transistor size makes them more sensitive. Beyond physics, larger systems and more complex components contribute to keep cross-sections increasing. At the microscopic level of system design, such as memory chips, components become more complex each generation. Density, size and geometry combine to make component cross-sections grow [7]. At the macroscopic scale,

large multiprocessor and multi-FPGA systems are becoming common. Increasing the number of devices increases the system cross-section. The third issue is the location of the system. Airborne multiprocessor systems are not uncommon, such as the multiprocessor SAR implementation that was designed for a Boeing 707 [9], and airborne FPGA-based radar systems are being researched [10]. High altitude military and commercial aircraft systems encounter more than 300 times the amount of neutron radiation than a sea-level system. With all of these factors, we believe problems with atmospheric radiation will become more problematic as designers and manufacturers push the limits of system complexity and use.

The expense of soft errors in terrestrial systems is not straightforward. On one hand, a company's goodwill can be devalued by unmitigated or unshielded soft errors in their manufactured systems. Sun Microsystems received negative press from the way they handled soft errors in their high-end servers [15]. On the other hand, since the upset rates are lower than space-based systems, the expense of shielding or mitigating might be very expensive as well.

System designers who specialize in highly available, highly reliable systems, such as reconfigurable supercomputers, need to mitigate or shield neutron radiation. Shielding systems from neutron radiation is a challenge, as neutrons can bore through five feet of concrete. Shielding for neutron radiation is usually done with meters of concrete, rock, dirt, and water. IBM tested their DRAM chips in an "underground vault, shielded by 50 feet of rock" [5] to be able to test their components without cosmic ray interference. Since shielding is difficult, mitigation methods are needed. Current mitigation methods, such as Triple Modular Redundancy (TMR), are designed for larger incidence rates than the SERs from neutron radiation, which makes the increase in power, speed, and area too expensive. Low-impact mitigation methods that guarantee reliability without sacrificing speed, power or area are needed. Especially attractive are methods that can be tuned to match the expected SER for a given location and system. Ideally, these methods will be researched now while the problem only affects the very high-end large-scale computing systems and not every high-end commodity server. Section 3 of this paper presents several options for low-impact mitigation.

In this paper we explore estimating cross-sections and mitigating soft errors. First, the cross-sections for FPGAs, memory chips, and microprocessors are estimated in Section 2. From these estimates we gauge the SER of the Cray XD1 reconfigurable supercomputer. In Section 3 low-impact mitigation methods are presented. The paper concludes with future work.

2 Estimates

In this section estimates for the SER for a variety of hardware components for nine locations are presented. The hardware components that we analyze are FPGAs, memory chips and microprocessors. As neutron flux is location-dependent, we estimate the performance of these hardware components for a variety of locations, as listed in Table 1. We used a variety of methods to determine the neutron flux, including the JEDEC Standard [12], a Xilinx talk on neutron testing [11], and the cosmic ray intensities from IBM [28, 8]. Note that there are several high altitude aircraft options, such as the U-2 and the ER-2, which operate in the 55,000-70,000 feet range [22, 17]. These aircraft operate at an altitude where neutron flux peaks and represent the worst case scenario for these estimates.

Location	Altitude (feet)	>10 MeV Flux ($\frac{n}{cm2 \cdot hr}$)
San Jose (SJ)	0	14.40
Albuquerque (AB)	5,200	53.28
Cheyenne (CH)	6,100	71.40
Los Alamos (LA)	7,200	90.00
Leadville (LE)	10,200	180.04
White Mountain (WM)	12,000	338.40
Mauna Kea (MK)	13,500	229.57
Commercial Aircraft (CA)	40,000	2041.24
Military Aircraft (MA)	60,000	4680.00

Table 1. Test Locations

2.1 General Trends in Estimating SER

Before continuing, we note some general trends in these estimates. SER is proportional to both device size and flux. Almost all hardware components have problems with soft errors at high altitudes or in large systems. Therefore, we expect the upset rate to be significantly worse when we increase the altitude or system size. The point of this paper is not to beat up on vendors but to look at these components realistically so that system designers can build reliable, available systems.

We estimate the SER of systems from the SER of a reference system. The equation for determining SER in the reference system is:

$$SER = flux \times \sigma_{dev} \quad (1)$$

where σ_{dev} is the *device cross-section*. In this paper, there are three types of cross-sections: system (σ_{sys}), device, and bit (σ_{bit}). Equation 1 holds for all three types of cross-sections. We assume the difference in SER between two locations is the ratio of the fluxes.

We also assume that the difference in SER between two systems is the ratio in the system sizes, as the cross section tends to not change much between device families [11]. If *sys1* and *loc1* are the system size and location of the reference system, then the SER of another system with size *sys2* and location *loc2* is estimated as:

$$SER_{loc2} = \frac{flux_{loc2}}{flux_{loc1}} \times \frac{\sigma_{sys2}}{\sigma_{sys1}} \times SER_{loc1} \qquad (2)$$

In Equation 2 the ratio between the two fluxes scales the SER from one location to the next. The difference in two systems is determined by:

$$\frac{\sigma_{sys2}}{\sigma_{sys1}} = \frac{dev_size_2 \times num_devs_2 \times \sigma_{bit2}}{dev_size_1 \times num_devs_1 \times \sigma_{bit1}} \qquad (3)$$

where *dev_size* is the number of memory bits in each device, and *num_devs* is the number of devices in the system. These equations provide order of magnitude estimates for the soft error rate from a limited amount of experimental and theoretical soft error data and the model of estimating flux.

Besides scaling for flux and system size, there are several factors that might necessitate lowering (*derating*) or raising (*uprating*) the SER. The SER for a system is dependent on what the system is calculating. If only a fraction of the system is being used, then the SER for the system should be derated. Uprating might be needed in cases where the estimates from older technology are used to predict the SER for newer technology. Finally, when the estimates are based on accelerator data, the SER from atmospheric conditions is often lower. These factors are discussed below.

The SER is design- and data-dependent. The bit cross-section values for FPGAs and memory chips we present assume that the entire component is used and that each bit is important. In reality, most designs do not use the entire device and some upsets are masked by the design. Therefore, the estimated SERs might be overstated for any given design. The general "rule of thumb" is to derate by a factor of 5-20 for system utilization.

The transistor size also plays an important role in the SER. Most of the estimates in this paper were made with one transistor size. Smaller transistor sizes are more likely to upset which might make the SERs larger than we estimated. In our ongoing proton radiation research, which mimics neutron radiation at high energies, we found that the Virtex II is 30 times more likely to have multi-bit upsets (MBUs) than the Virtex I [13, 20]. Therefore, we expect single-bit and multi-bit upsets to increase as the transistor size shrinks, but the uprating factor is still under research.

Finally, the data in this paper is from both accelerator and atmospheric tests. Atmospheric testing is ideal but time consuming. Accelerator results tend to report larger SERs than atmospheric testing, so the results need to be derated. Xilinx has researched the derating factor between accelerator and atmospheric tests. Their results is the "Rosetta Factor" of 1.5 [11].

There are several factors that can affect the SER for an actual system. These estimates are guidelines to the worst possible scenario. The actual SER for a system should be done by testing the design with the expected data.

2.2 FPGAs

In this section, we present an analysis of Xilinx, Actel, and Altera FPGA chips for soft errors. These estimates are based on three sets of previously published test results from Xilinx, Altera, and iRoC Technologies [11, 6, 21]. The Rosetta NSEU Test is an ongoing Xilinx test that does atmospheric testing of Xilinx chips. The Virtex-II Rosetta system is 100 XC2V6000 chips arranged in a 10x10 matrix for a total of 1.96E9 configuration bits. There are four of these Rosetta systems operating at different altitudes, as shown in Table 2. The next set of results presented are from the iRoC Technologies test commissioned by Actel for full spectrum neutron beam analysis. In this test, several FPGA chips (AX1000, APA1000, XC2V3000, XCS1000, EP1C20) were tested at the Los Alamos Neutron Science Center (LANSCE) neutron accelerator. With the iRoC Technologies test we correlate the Rosetta test results. The Altera test presents data for the SER of Altera FPGA devices.

2.2.1 Xilinx Rosetta Test

In the Rosetta data there is a strong correlation between flux and the number of upsets. When the upsets for the four locations are normalized to upsets/year, the relationship between flux and upsets/year for the Rosetta system are:

$$upsets/year = 3.0899 + 0.3033 \times flux \qquad (4)$$

Equation 4 determines the SER for a year and can be divided by 8760 hours/year to determine the SER per hour. Assuming the average bit cross-section for the Rosetta system is stable, we can scale flux and cross-sections using Equation 2 to determine how different locations and system configurations are affected by soft errors.

First, we increase flux. Since altitude has a first order effect on flux, we expect SER increases with altitude. Table 2 shows us that, as altitude increases, the mean time to upset (MTTU) for the Rosetta system

Loc	Flux ($\frac{n}{cm2 \cdot hr}$)	Hours	Upsets	Bit Cross-section	$\frac{Upset}{bit \cdot hr}$	MTTU (device-hours)
SJ	14.4	10632	6	2.00E-14	2.88E-13	1772.00
AB	53.28	13536	34	2.41E-14	1.28E-12	398.12
MK	229.57	2160	18	1.85E-14	4.25E-12	120.00
WM	338.40	5496	66	1.81E-14	6.13E-12	83.30

Table 2. Virtex-II Rosetta Test Results

Loc	Flux ($\frac{n}{cm2 \cdot hr}$)	Bit Cross-Section	MTTU (hours)
CH	71.40	2.02E-14	343.33
LA	90.00	1.97E-14	279.59
LE	180.04	1.87E-14	147.25
CA	2041.24	1.78E-14	14.08
MA	4680.00	1.77E-14	5.97

Table 3. Estimated Rosetta Test Results

decreases. For the locations in Table 1 not in Table 2, we estimate the MTTU for a Rosetta system in Table 3 using equation 4 and an average bit cross-section of 2.02E-14. At high terrestrial altitudes, the Rosetta system upsets 21.3 times more than at sea level. High altitude aircraft levels upset 327.8 times more than at sea level. The estimated results from the ratios for the four tested locations are consistent with the actual test results for these locations. Therefore, we should be able to use the ratio of the fluxes to estimate the SER for any given altitude using the sea-level data.

While flux is bound by location and is at its maximum at 60,000 feet, system cross-sections are not bound and can increase in two dimensions. Either the individual device size or the number of devices can increase to make the cross-section larger. The results of increasing both dimensions are presented below.

	XC2V6000		XC4VFX140		
Loc	500	1000	100	500	1000
SJ	350.80	175.40	845.12	169.02	84.51
AB	94.81	47.41	228.41	45.68	22.84
CH	70.80	35.40	170.57	34.11	17.06
LA	56.13	28.06	135.22	27.04	13.52
LE	28.06	14.03	67.59	13.52	6.76
MK	22.00	11.00	53.01	10.60	5.30
WM	14.93	7.46	35.96	7.19	3.60
CA	2.47	1.24	5.96	1.19	0.60
MA	1.08	0.54	2.60	0.52	0.26

Table 4. MTTU (hours) for Different Systems

First, we increase the system cross-section by increasing the number of devices in the Rosetta system. Using the average bit cross-section of 2.02E-14, we calculated the MTTU for 500 and 1000 device Rosetta systems for all of the locations in Table 1. These results, shown in Table 4, scale by the product of the flux and the bit cross-section ratios. As expected, the MTTU for the 500 device San Jose system decreased by a factor of 5 times from the 100 device system. Similarly the Mauna Kea 500 device Rosetta system has a MTTU that is 74 times smaller than the San Jose 100 device Rosetta system, which approximates the value of the flux ratio multiplied by the device number ratio.

Next we increased the individual device sizes in the Rosetta system. Table 4 shows the estimates for using the 4VFX140 in 100, 500 and 1000 device Rosetta systems. The estimated number of configuration bits for the 4VFX140 device is 40,678,656 bits, which was calculated by subtracting the number of bits devoted to block RAM from the bitstream length [24, 25]. Because the 4VFX140 device is twice the size of the XC2V6000, the MTTU decreases by half. Therefore, the 1000 XC2V6000 device Rosetta system is expected to upset every 32 minutes at 60,000 feet, and the 1000 4VFX140 device Rosetta system is expected to upset every 15.6 minutes at the same altitude.

Chip	Soft Errors (FITs)
AX1000	< 0.082
APA1000	< 0.038
XC2V3000	8680
XC3S1000	1240
EP1C20	N/A

Table 5. iRoC Technologies LANSCE Test

2.2.2 iRoC Technology Test

The iRoC Technology test most similar to the Rosetta test is their full spectrum test done at LANSCE. The full spectrum neutron beam at LANSCE simulates the range of energies seen in terrestrial radiation. The LANSCE test includes five chips: AX1000, APA1000, XC2V3000, XC3S1000, and EP1C20. We are in particular interested in their SEU testing shown in Table 5. The two Actel chips (AX1000 and APA1000) are not expected to experience SEUs, since the configuration is not stored in volatile memory. The remaining chips are SRAM-based and are expected to experience soft errors. The Altera chip (EP1C20) did not have readback ability, so the number of soft errors is unknown. For our analysis, we looked at the SER for individual chips, and 100 device Rosetta-like systems. Finally, we extrapolate back to the 100 device XC2V6000 Rosetta system so that the two tests can be compared.

The LANSCE data in Table 5 is in failures per 1E9 hours (FITs). First we convert from FITs to MTTU (hours). Next, we extrapolate the sea-level results to the other eight locations using Equation 2. The MTTU data for all of the locations and the Actel and Xilinx chips are shown in Table 6. As expected, as flux

Loc	AX1000	APA1000	XC2V3000	XC3S100
SJ	1.22E10	2.63E10	1.15E5	8.06E5
AB	3.30E9	7.11E9	3.11E4	2.18E5
CH	2.46E9	5.31E9	2.32E4	1.63E5
LA	1.95E9	4.21E9	1.84E4	1.29E5
LE	9.75E8	2.10E9	9.21E3	6.45E4
MK	7.65E8	1.65E9	7.23E3	5.06E4
WM	5.19E8	1.12E9	4.90E3	3.43E4
CA	8.60E7	1.86E8	8.13E2	5.69E3
MA	3.75E7	8.10E7	3.54E2	2.48E3

Table 6. MTTU (hours) for Single Chips (iRoC)

increases, the MTTU decreases. Since the two Actel chips use anti-fuse and flash memory to store the configuration, soft errors are rare, which is reflected in the MTTU data. For a single Xilinx chip, the likelihood of upsetting at low elevations is also rare, but at high elevations the SER is on the order of a few weeks to a few months.

Loc	AX1000	APA1000	XC2V3000	XC3S1000	XC2V6000
SJ	1.83E8	3.95E8	1.73E3	1.21E4	7.86E2
AB	4.94E7	1.07E8	4.67E2	3.27E3	2.12E2
CH	3.69E7	7.96E7	3.49E2	2.44E3	1.58E2
LA	2.93E7	6.32E7	2.76E2	1.94E3	1.26E2
LE	1.46E7	3.16E7	1.38E2	9.68E2	6.28E1
MK	1.15E7	2.48E7	1.08E2	7.59E2	4.92E1
WM	7.78E6	1.68E7	7.35E1	5.15E2	3.34E1
CA	1.29E6	2.78E6	1.22E1	8.53E1	5.54E0
MA	5.63E5	1.21E6	5.32E0	3.72E1	2.42E0

Table 7. MTTU (hours) for Rosetta-like Systems (iRoC)

The results of Rosetta-like systems are in Table 7. These results are the Table 6 results derated by both system size (100) and the Rosetta factor (1.5). To be able to compare the iRoC test to the Rosetta test, we estimate the SER for a 100 device XC2V6000 Rosetta system from the 100 device XC2V3000 Rosetta-like system. The XC2V3000 device is smaller than the XC2V6000 device by a factor of 2.2, so the MTTU was derated by 2.2. These results indicate an MTTU that is 2.37 times smaller than the Rosetta data, but show a strong correlation between the two tests.

2.2.3 Altera Test

Altera Corporation presented soft error test results for four of their SRAM FPGAs (EP1C6, EP1C20, EP1C25, EP1S80) for San Jose in 2004 [6]. These results are shown in Table 8 in FITs. For our analysis, we converted these results to MTTU in hours, as shown in Table 9, so that the results could be compared with the other two tests. Next we extrapolated the sea-level data to the other eight locations using Equation 2. Finally, we extrapolated the data from a one chip system to a 100 device Rosetta-like system.

With the Rosetta system data we can compare these

Chip	Soft Errors (FITs)
EP1C6	250
EP1C20	730
EP1S25	1950
EP1S80	6000

Table 8. Altera Test Results

Loc	EP1C6	EP1C20	EP1S25	EP1S80
SJ	4000000.00	1369863.01	512820.51	166666.67
AB	1081081.08	370233.25	138600.14	45045.05
CH	806722.69	276274.89	103425.99	33613.45
LA	640000.00	219178.08	82051.28	26666.67
LE	319928.90	109564.69	41016.53	13330.37
MK	250903.86	85925.98	32167.16	10454.33
WM	170212.77	58292.04	21822.15	7092.2
CA	28218.14	9663.75	3617.71	1175.76
MA	12307.69	4214.96	1577.91	512.82

Table 9. MTTU (hours) for Single Chips (Altera)

results to the Xilinx data from the Rosetta and iRoC Technologies test. On the surface, the MTTU in hours is similar to that of the Rosetta data. In particular, the 100 device EP1S80 Rosetta system upsets on the same scale as the original 100 device XC2V6000 Rosetta system. The bit cross-section is calculated using Equation 1, and these results can be fully compared to the Xilinx data. Table 10 shows the bit cross-sections for the four Altera chips. The Altera bit cross-sections are slightly smaller than the Xilinx bit cross-sections, but of the same magnitude. Therefore, Altera SRAM FPGAs might be more resistant to soft errors than Xilinx SRAM FPGAs but only by a very small margin.

Chip	Bit Cross-Section
EP1C6	1.49E-14
EP1C20	1.42E-14
EP1S25	1.72E-14
EP1S80	1.75E-14

Table 10. Bit Cross-Section for Altera Chips

2.2.4 Conclusion

From these results, we are able to estimate the order of magnitude of the soft error rate of FPGAs using reference systems. Using Equation 2 we can estimate the soft error rate of a given SRAM FPGA from a reference SRAM FPGA as long as the flux and cross-sections are known.

2.3 Memory Cells

For estimating memory's MTTU, we use data from two different tests: IBM's sea-level tests of three different types of memory and iRoC Technology's multiple

Altitude (feet)	$\frac{Upsets}{Bit \cdot Hour}$	$\frac{Upsets}{Hour}$ for 1MB	$\frac{Upsets}{Hour}$ for 1 GB
0	1.68E-012	1.4E-8	0.01
30,000	2.63E-009	2.2E-5	22.59
65,000	1.45E-008	1.2E-4	124.55

Table 11. iRoC Technology Memory tests

altitude atmospheric tests [5, 17]. The iRoC Technology data, shown in Table 11, exhibits how the upset rate per bit increases several orders of magnitude from sea level to 65,000 feet for SRAM and DRAM, which appears to be without ECC protection. While these upset rates seem very small, memory systems often have 10^6 to 10^9 bits. Table 11 shows that 1 GB of memory experiences 125 upsets every hour at 65,000 feet. Therefore, soft errors are more likely to manifest in the memory system than FPGAs.

Loc	Parity	SEC ECC	Chipkill
SJ	1.15E+01	2.92E+05	4.38E+07
AB	3.10E+00	7.89E+04	1.18E+07
CH	2.31E+00	5.89E+04	8.83E+06
LA	1.83E+00	4.67E+04	7.01E+06
LE	9.16E-01	2.34E+04	3.50E+06
MK	7.19E-01	1.83E+04	2.75E+06
WM	4.88E-01	1.24E+04	1.86E+06
CA	8.08E-02	2.06E+03	3.09E+05
MA	3.53E-02	8.98E+02	1.35E+05

Table 12. Actual and Predicted IBM Results for MTTU (hours) for 1 GB of DRAM

IBM's research on DRAMs at sea level test the mitigation efforts of parity, SEC ECC, and Chipkill technologies. Their research assumes that 4% of all soft errors involve multiple bits. Multi-bit upsets break ECC protection, which translates to lower MTTU values than expected. Their results are shown in the first row in Table 12. Using flux scaling from Equation 2, we extrapolated the sea-level results to the other eight locations. When Tables 11 and 12 are correlated, the parity DRAM estimates are within an order of magnitude of the atmospheric tests.

These estimates show a drastic increase in the number of upsets above sea level for parity DRAM. One gigabyte of RAM is not unreasonable; many commodity machines have 1 GB of RAM. Servers regularly have an order of magnitude more than 1 GB DRAM. Therefore, ECC- or Chipkill- protected RAM should be used in a server- or supercomputer-grade computing system. ECC-protected RAM is significantly more expensive, slower, and needs more power than non-parity RAM. Consequently, many system designers still use non-parity RAM.

Loc	1 MB of Parity Memory	1 MB of ECC-Protected Memory
SJ	1.20E+07	3.06E+11
AB	3.25E+06	8.28E+10
CH	2.42E+06	6.18E+10
LA	1.92E+06	4.90E+10
LE	9.61E+05	2.45E+10
MK	7.54E+05	1.92E+10
WM	5.11E+05	1.30E+10
CA	8.48E+04	2.16E+09
MA	3.70E+04	9.42E+08

Table 13. MTTU for a 1 Mb L2 Cache (IBM)

2.4 Microprocessors

Soft errors in microprocessors have been important for manufacturers for several years now. Since microprocessors are both extensively manufactured and often need to be highly reliable, unmitigated soft errors can be very expensive. Most server-grade microprocessors have ECC-protected L2 and L3 caches and parity-enabled L1 caches, as well as cache scrubbing. Cache scrubbing in either hardware or software should drastically reduce the number of failures due to soft errors.

Cosmic ray interference is still present in microprocessor systems, though. Many current and most old microprocessors do not have protected caches. Due to the financial advantage of using commodity parts many large scale microprocessor systems use older, cheaper parts, which usually have unprotected L2 caches. Despite their small size, an L2 cache is still susceptible to soft errors. An L2 cache is usually between 512 KB to 2 MB of SRAM. Table 11 shows that the MTTU for a 1 MB unprotected L2 is nearly 1370 years at sea level. At 60,000 feet 1 MB of unprotected L2 cache upsets every four years. As with FPGAs, the more microprocessors in a system, the larger the cross-section and the smaller the MTTU. Table 14 shows the MTTU for Q Cluster sized systems. These numbers show that a cluster this size would expect to upset ever 61 days at sea level and 4.5 hours at 60,000 feet. ECC-protected memory upsets are rarer: one upset per 35 million years at sea level and 107,000 years at 60,000 feet for 1 MB. For Q Cluster sized systems, the MTTU is 4270 years at sea level and 13 years at 60,000 feet. Therefore, for large clusters, only microprocessors with ECC protected caches should be used.

The second problem is that the caches are not the only memory in a microprocessor. Approximately 10% of the memory cells in the SPARC v8 microprocessor are in register files [18]. Register files are often not parity-enabled or ECC-protected, which means that an upset to a register file, while rare, can be catastrophic. Without parity, register upsets silently and undetectably corrupt register data which might contain

Loc	1 MB of Parity Memory	1 MB of ECC-Protected Memory
SJ	1.47E+03	3.74E+07
AB	3.96E+02	1.01E+07
CH	2.96E+02	7.54E+06
LA	2.35E+02	5.98E+06
LE	1.17E+02	2.99E+06
MK	9.20E+01	2.34E+06
WM	6.24E+01	1.59E+06
CA	1.03E+01	2.64E+05
MA	4.51E+00	1.15E+05

Table 14. MTTU for 8192 Microprocessors with 1 Mb L2 Caches (IBM)

the state of a program, or an intermediate calculation. Without ECC protection a register upset cannot be corrected. Fortunately, register files represent a very small amount of data.

Part Number	Number in XD1	Cfg Bits Per Chip
XC2S50E	6	597280
XC2S300E	2	1810112
XC2VP40	12	12329312
XC2VP50	6	14745184

Table 15. FPGA Parts in the XD1

2.5 Large Scale Systems: The Cray XD1 Reconfigurable Supercomputer

The Cray XD1 system is a reconfigurable supercomputer with both standard microprocessors and FPGAs connected through a high speed interconnect. One chassis has 12 Opteron microprocessors, 26 Xilinx FPGA chips (outlined in Table 15) and 24 GB of ECC-protected RAM. Assuming a worst case scenario for the Opterons, where the MBUs don't get scrubbed out of the 1 MB L2 cache, we estimated the MTTU for the XD1 system, as shown in Table 16. To determine the FPGA upset rate, the number of configuration bits is needed, which were determined by subtracting the number of block RAM bits from the configuration bitstream sizes for consistency with the previous estimates [23, 26]. The total number of configuration bits for all of the FPGA parts is 243,626,752 bits per chassis, which is an order of magnitude smaller than the Rosetta system. The MTTU for the entire system was predicted by adding together the data for the FPGA, memory and microprocessor components. Finally, we scaled the MTTU to multi-chassis systems in Table 17. We scaled these systems by the number of racks, where 12 chassis make one rack. The last column of data is a 57 rack XD1 Cray system that has roughly the same number of microprocessors as the Q Cluster.

These results show that the FPGAs and memory have nearly the same effect on the MTTU, even though

Loc	FPGA	Microprocessor	Memory	Total
SJ	14264.35	3.06E+11	12166.67	6566.13
AB	3855.23	8.28E+10	3288.29	1774.63
CH	2876.84	6.18E+10	2453.78	1324.26
LA	2282.30	4.90E+10	1946.67	1050.58
LE	1140.89	2.45E+10	973.12	525.17
MK	894.75	1.92E+10	763.17	411.87
WM	606.99	1.30E+10	517.73	279.41
CA	100.63	2.16E+09	85.83	46.32
MA	43.89	9.42E+08	37.44	20.20

Table 16. MTTU (Hours) for the XD1 (Xilinx, IBM)

the ECC-protected RAM has roughly 100 times more bits that the FPGAs. Therefore, both the memory and the FPGAs need to mitigate soft errors. The larger scale system estimates in Table 17 indicate that a Q-sized XD1 cluster will upset every 92 minutes in Los Alamos, where as the Q Cluster upsets approximately every 8 hours. With almost 18000 FPGAs and 16500 GBs of ECC-protected memory, a Q-sized XD1 cluster pushes the system design boundaries. Quite possibly these numbers will need to be derated for incomplete system utilization. This result indicates that soft error mitigation is needed to make large reconfigurable supercomputing clusters viable.

Loc	1 Rack	4 Racks	Q-sized
SJ	547.18	136.79	9.60
AB	147.89	36.97	2.59
CH	110.36	27.59	1.94
LA	87.55	21.89	1.54
LE	43.76	10.94	0.77
MK	34.32	8.58	0.60
WM	23.28	5.82	0.41
CA	3.86	0.97	0.07
MA	1.68	0.42	0.03

Table 17. MTTU (Hours) for Large-Scale XD1 Integration (Xilinx, IBM)

2.6 Summary

While the error rates caused by terrestrial-based neutron flux might be significantly smaller than the error rates for space-based radiation, the number of possible upsets and the cost of these upsets are not insignificant. Therefore, reliability methods targeted at mitigating upsets that are expected to occur in the hourly or daily time frames are needed.

3 Low-Impact Mitigation Schemes

Due to the low SER from neutron radiation, low-power and low-area mitigation methods are needed. There are many common sense approaches to system

design that can be employed, such as using ECC-protected RAM and microprocessors. Using either the host's processor or an embedded processor for mitigation, instead of the FPGA, can reduce the amount of power and area needed. Likewise, mitigation methods that can be tuned to match the SER can lower the amount of power needed to detect/mitigate upsets.

This section presents several possible low-impact mitigation methods. These processes are broken into two broad categories: support logic methods and partial configuration methods. Support logic scenarios are implemented at the design level. Partial configuration methods focus on using the device's readback support to determine the current state of the device, and then use either the FPGA or a processor for detection/correction processing.

3.1 Support Logic Methods

Two support logic methods are presented in this paper. The first is a Xilinx design for the Virtex II Pro devices and uses one of the embedded processors. The second method is selective TMR, which automatically applies TMR to the critical areas of a design.

3.1.1 SEU Controller

Xilinx has provided a solution for soft error detection and correction for the Virtex II Pro FPGA family called the SEU controller module [4]. This module uses the ICAP interface and a Power PC 405 core. The module takes between 0.10-0.96 seconds to completely scan an entire device depending on device size. The frequency of the readback is user determined, so power consumption can be reduced by matching detection/correction cycles with SERs.

Since this module is implemented on the FPGA fabric it is susceptible to soft errors itself. For a XC2VP50 device, the module takes up 0.7% of the device area. Xilinx determined that the module has a mean time between failure (MTBF) of nearly 28,600 years [4]. The module was designed to detect any soft errors originating from the SEU controller module.

3.1.2 Selective Triple Modular Redundancy

TMR is commonly used in space-based applications to mitigate soft errors. A full TMR implementation of a design increases the power and area consumption by a factor of three. Experimentation has been done on applying TMR selectively to designs [19]. In selective TMR, the design is analyzed for critical gates. All critical gates are triplicated and the necessary voters are inserted. This method is useful when only a small percentage of gates are critical. For circuits with a high percentage of critical gates, the hardened design approximates a full TMR implementation.

3.2 Partial Configuration Methods

Readback allows the designer to access the device's current configuration state, which can be analyzed for configuration upsets. There are limitations to readback, such as not being able to write to block RAM and LUT RAM during readback without data corruption. Therefore, processing during readback is risky. Since the SER is very low, halting computation temporarily to readback is not very invasive. Below we present three different methods for doing detection and correction based on readback. The first method — single frame correction — uses CRC frame checks to detect upsets in the readback. The next method, processor-based detection of critical upsets, reduces power consumption by using the host processor. Finally, scrubbing is a method of reconfiguring the FPGA in anticipation of an upset. These methods are covered below.

3.2.1 Single Frame Correction

To implement single frame correction, readback supplies the current state of the frames and CRC frame checks are used as part of the detection/correction phase. Detection/correction computation are done in parallel with readback so that entire process takes no longer than readback. Readback execution times are dependent upon the number of frames in the device, the number of bits per frame and the interface used for readback. For a XC2VP100 with 3500 frames and 9792 bits/frame, doing readback on the SelectMap interface takes approximately 90 ms and 1 s for the JTAG interface. For the Virtex 4 family, Xilinx has included ECC Logic that calculates whether there has been an upset in the configuration data while doing a readback [24].

Single frame correction for the entire chip must be faster than the SER. For example, assuming single frame correction for a XC2VP100 processes in parallel with readback, then the SER cannot exceed 11 upsets per second. For neutron radiation, the SER is several orders of magnitude slower than single frame correction. Therefore, power can be conserved by running the process intermittently.

3.2.2 Processor-based Detection of Critical Upsets

This detection/correction scheme maintains a database of bad bits for comparing readback data. The host pro-

cessor determines if the configuration has been upset while the FPGA is computing. The database entries are marked as either critical or sub-critical, as with selective TMR. Once upsets are prioritized as being critical or sub-critical, a repair cycle can be triggered in the appropriate time frame. With sub-critical upsets, computation continues to the next checkpoint before repairing, whereas critical upsets trigger an immediate repair cycle. As with the single frame correction scheme, the entire detection/correction process is faster than the rate of upset, so power conservation can be explored.

An important aspect of processor-based detection is a method to communicate from the FPGA to the host processor that a detection process needs to be started. Exception-based processing schemes allow the FPGA to raise an exception or an interrupt when an upset is suspected or detected. In this scenario, the FPGA processes normally until an upset either occurs or has likely occurred. For instance, voters can be used to raise an exception when inputs don't match. Likewise, readback can be used to raise exceptions. This scenario could help integrate FPGAs into an XD1-style system, as exceptions and interrupts are commonly used in many types of computer systems to alert the processor of issues with the peripheral devices.

Exception-based FPGA processing in the presence of neutron radiation upsets allows the designer to be more reactive than proactive when mitigating upsets. The detection/correction process would not be executed until needed.

3.2.3 Scrubbing with Cyclic Redundancy Check

Scrubbing reloads the CLB Frames from the bitstream [2]. When combined with TMR the failure rate due to neutron radiation induced bitstream upsets are zero as long as the SER is low [27]. For this scheme, support circuitry is needed to generate the memory map addresses and control the memory and SelectMap interfaces. Scrubbing is a more lightweight process than single frame correction, as there is no detection or correction. Scrubbing can also be run while the device is processing data. There is some risk with using scrubbing, though, in the rare but possible case that the SelectMap interface is corrupted by a soft error.

The scrub rate is dependent on the SER. Xilinx recommends scrubbing at rate 10 times greater than the upset rate to guarantee that no more than one upset will be present between scrubs [2]. For a SER of 13 upsets per week (or an upset every 12.92 hours), this would mean scrubbing the device every 80 minutes. One possibility to save power and to increase the probability of catching all of the upsets would be to perform scrubbing in conjunction with either single frame correction or selective TMR.

4 Future Work and Conclusions

In conclusion, we expect mitigation of soft errors to become more important in future system design due to decreasing transistor size and increasing complexity of systems. Redundancy methods that take into account the infrequent nature of these upsets are needed so that the errors can be mitigated without severely impacting system performance. This paper has presented a few possible low-impact mitigation methods that use support logic and/or partial reconfiguration.

In the future, we will be stepping back from the mitigation methods to focus on the reliability of mitigation. Currently, modeling the reliability of large circuits is very time consuming and complex. We are focusing on building a system that determines the combinational reliability of FPGA implementations of circuits using EDIF and a library of reliability calculations for FPGA cells. We would like to easily and accurately model reliability of large circuit systems on particular FPGA fabrics so that we can compare the relative reliability of different architectures.

We are also interested in more traditional reliability research on how inserting scrubbing or detection/correction methods into FPGA designs can be modeled using Markov methods. Using scrubbing with TMR virtually eliminates soft errors in the presence of radiation, but currently there is no way to empirically determine scrub rates. Scrub rate are based on "rule of thumb" calculations based on upset rate. We would like to determine a design's scrub rate using traditional reliability modeling methods.

References

[1] Top 500. Top 500 supercomputer sites. On Web http://www.top500.org/list/2003/11/, Nov 2003.

[2] Carl Carmichael, Michael Caffrey, and Anthony Salazar. Correcting single-event upsets through Virtex partial configuration. Technical Report XAPP216, Xilinx, Inc, 2000.

[3] Cisco. Increasing network availability: Detecting and correcting network-impacting singe event upset (SEU) parity events. Technical report, Cisco, (Date Unknown).

[4] Derek Curd. Single event upset (SEU) detection and correction in Virtex-II Pro devices. Technical Report XAPP673, Xilinx, Inc., 2004.

[5] Timothy Dell. A white paper on the benefits of Chipkill-Correct ECC for PC server main memory. Technical report, IBM, 1997.

[6] Amr El-Ashmawi and Joel Seely (presenter). Maintaining data integrity in programmable logic in extreme radiation environments through error detection techniques. In *The Proceedings of Military and Aerospace Programmable Logic Devices (MAPLD)*, 2004.

[7] Eto et al. Impact of neutron flux on soft errors in MOS memories. In *Internation Electron Devices Meeting (IEDM)*, 1998.

[8] J. F. Ziegler et al. Ibm experiments in soft fails in computer electronics (1978-1994). *Journal of Research and Development*, 40(1):3–18, 1996.

[9] Jeremy Kepner et al. Deployment of SAR and GMTI signal processing on a Boeing 707 aircraft using pMatlab and a bladed Linux cluster. In *Proceedings of High Performance Embedded Computing*, pages 25–26, 2004.

[10] J.I. Roman-Nieves et al. High altitude compact solid state 95 Ghz cloud radar. In *Proceedings of IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, volume 5, pages 2672–2674, June 2002.

[11] Joe Faubla, Austin Lesea, and Saar Drimer. The NSEU sensitivity of static latch based FPGAs and flash storage devices. In *Military and Aerospace Programmable Logic Devices International Conference*, 2004.

[12] JEDEC Solid State Technology Association. *JEDEC Standard 89: Measurement and Reporting of Alpha Particles and Terrestrial Cosmic Ray-Induced Soft Errors in Semiconductor Devices*, August 2001.

[13] Eric Johnson, Keith Morgan, Nathan Rollins, Michael Wirthlin, Michael Caffrey, and Paul Graham. Detection of configuration memory upsets causing persistent errors in SRAM-based FPGAs. In *Proceedings of the MAPLD Conference*, September 2004.

[14] Jason Lockhart. personal conversation.

[15] Daniel Lyons. Sun screen. On Web http://www.forbes.com, November 2000.

[16] Sarah Michalak. Estimation of the expected weekly number of soft errors in QA or QB. Technical Report LA-UR-04-5162, Los Alamos National Labratory, 2004.

[17] Eugene Normand. Single event effects in avionics. Technical report, iRoC Technologies, (Date Unknown).

[18] M. Rebaudengo, M. Sonza Reorda, and M. Violante. An accurate analysis of the effects of soft errors in the instruction and data caches of a pipelined microprocessor. In *The Proceedings of Design, Automation, and Test in Europe (DATE)*, pages 10602–106007, 2003.

[19] P. K. Samudrala, J. Ramos, and S. Katkoori. Selective triple modular redundancy for SEU mitigation in FPGAs. In *Military and Aerospace Applications of Programmable Logic and Devices (MAPLD)*, 2003.

[20] Matthew Stettler, Jim Krone, Paul Graham, and Michael Caffrey. Radiation effects and mitigation strategies for modern fpgas. In *In Proc. of 10th Annual Workshop for LHC and Future Experiments*, 2004.

[21] iRoC Technologies. Radiation results of the SER test of Actel, Xilinx, and Altera FPGA instances. On Web: www.actel.com, October 2004.

[22] Airforce Technology. U-2 high altitude reconnaissance aircraft (Dragon Lady), USA. On Web: http://www.airforce-technology.com/projects/u2/.

[23] Xilinx. Spartan-IIE 1.8V FPGA family: Complete data sheet. Technical Report DS077, Xilinx, Inc., 2004.

[24] Xilinx. Virtex-4 configuration guide. Technical Report UG071, Xilinx, Inc., 2004.

[25] Xilinx. Virtex-4 family overview. Technical Report DS112, Xilinx, Inc., 2004.

[26] Xilinx. Virtex-II Pro and Virtex-II Pro X platform FPGAs: Complete data sheet. Technical Report DS083, Xilinx, Inc., 2004.

[27] C.C. Yui, G.M. Swift, C. Carlmichael, R. Koga, and J.S. George. SEU mitigation techniques of Xilinx Virtex II FPGAs. In *Proceedings of IEEE Nuclear and Space Radiation Effects Conference (NSREC)*, 2003.

[28] J. F. Ziegler. Terrestrial cosmic ray intensities. *Journal of Research and Development*, 42(1):117–140, 1998.