Effectiveness of Internal vs. External SEU Scrubbing Mitigation Strategies in a Xilinx FPGA: Design, Test, and Analysis

Melanie Berg,Member IEEE, C. Poivey, Member IEEE, D. Petrick, D. Espinosa, Austin Lesea, K. LaBel, Member IEEE, M. Friendlich, H. Kim, Anthony Phan

***Abstract*—We compare two scrubbing mitigation schemes for Xilinx FPGA devices. The design of the scrubbers is briefly discussed along with an examination of mitigation limitations. Proton and Heavy Ion data are then presented and analyzed.**

***Index Terms*—FPGA, Reconfiguration, Scrubbing, Xilinx**

I. Introduction

THE Advancement of commercial CMOS technology has afforded the Field Programmable Gate Array (FPGA) community a considerable increase in functional complexity and implementation options. Commercial FPGAs have become more efficient requiring less power with higher electrical performance than predecessors. Unfortunately, due to the reduction in core voltage, decrease in transistor geometry, and increase in switching speeds, commercial CMOS transistors have become more susceptible to incurring faults.

Because of the harsh environment of space and its effects on electronic devices, the aerospace community has traditionally followed a rigid/conservative design methodology. Selected devices targeted for flight (referred to as SEU-hardened) are generally made for optimal operation in mild to harsh radiation environments. SEU-Hardened FPGAs have configuration error cross-sections close to 0 even in worse case conditions. State of the Art, Rad-Hard FPGA logic error cross-sections are typically, between 1e^{-08}cm^2/flip-flop to 1e^{-07} cm^2/flip-flop under worst case. Unfortunately, current rad-hard FPGA parts have become very expensive and leave no room for mistakes or changes (one time programmable devices). This has lead space-system architects and researchers to investigate alternative approaches to design. Although it is well understood that commercial FPGA devices are more susceptible to upsets, one aspect of this investigation is to determine how to fully take advantage of cutting-edge commercial FPGA devices while adhering to the rigid constraints of flight projects.

One category of commercial devices that is being given a considerable amount of attention is reprogrammable SRAM based FPGAs. Xilinx (Virtex II, Virtex 4, and Virtex 5) SRAM-based FPGAs are the forerunners for aerospace research. This type of device can be reprogrammed because its configuration is stored in SRAM (vs. fixed configuration types such as anti-fuses). The implementation pros are speed and agility. However, a caveat is that the configuration memory is not radiation-hardened and is susceptible to faults. It is also important to note that SRAM-based FPGAs not only incur upsets in configuration but also incur upsets in their functional logic paths.

Proposed mitigation techniques for configuration vs. functional logic are very different and are not easy to implement. The purpose of configuration mitigation is to protect the FPGA configuration from upsets. This approach requires the system to have the ability to write into the configuration memory space with correct configuration data and can be categorized as follows:

Category 1 Reconfiguration: Apply a full reconfiguration to the FPGA (every so often or upon detection of error state). This requires that the system operation will come to a stop. Returning back to the previous state of operation can only be accomplished by having redundant circuitry (or extra-knowledge).

Category 2 Scrubbing: Write over portions of the configuration memory that do not disrupt operation. The system is fully operational. The caveat is that some portions of configuration memory and interface controls are not able to be written ("scrubbed") and are therefore still susceptible to upsets. Upon faults to inaccessible configuration space that cause Single Event Functional Interrupts (SEFIs), the system will have to come down and a category 1 reconfiguration will have to be performed.

Manuscript received September 7, 2007.

M. Berg is in support of NASA Goddard Space Flight Center, Greenbelt, MD 20771 USA

C. Poivey , M. Friendlich, H Kim, A. Phan, are with MEI Technologies are with MEI Technologies in support of NASA Goddard Space Flight Center, Greenbelt, MD 20771 USA (telephone: 301-286-1023, e-mail: cpoivey@pop500.gsfc.nasa.gov).

D. Petrick, K. A. LaBel are with NASA Goddard Space Flight Center, Greenbelt, MD 20771 USA.

This paper will focus on category 2 (scrubbing) configuration-upset mitigation schemes. A comparison of performance and error cross-sections between a self-contained category 2 scrubber developed by Xilinx [4] vs. an external category 2 scrubber developed by the NASA/GSFC Radiation Effects and Analysis Group is presented.

II. Trade-off: Complexity vs. flexibility

The major concern of implementing designs targeted for space within SRAM-based FPGAs is how to mitigate and protect against an increased probability of faults. For flight critical designs, the aerospace industry is conservative and uses hardened devices. However, for non-critical portions of a mission (such as data capture or processing information that can be recaptured or lossy), the aerospace industry is investigating the utilization of devices that provide increased flexibility. The ability to reprogram a function while in-flight is of great advantage to many missions, therefore projects are now considering the tradeoff between an increase of complexity for an increase in flexibility.

We show in this paper that for most missions it will be mandatory to apply configuration mitigation if implementing a SRAM-based FPGA design. We concentrate on three modes of category-two mitigation:

(1) No scrubbing
(2) Internal Scrubbing (Xilinx proprietary)
(3) External Scrubbing

III. Self-Contained Xilinx Scrubber – SEU Controller Block

A. General Description

The hardware for the SEU Controller block is comprised of several sub-modules as illustrated in Figure 1. Configuration memory is accessed through the Internal Configuration Access Port (ICAP). This port cannot be accessed by any other device but Xilinx internal circuitry. The Frame ECC block is responsible for error detection and correction. It uses a hamming code Single Error Correct and Double Error Detect (SECDED) scheme [4].

B. SEU Correction Operational Flow

The configuration memory space is divided into frames containing a corresponding parity syndrome. Once the SEU controller commences operation, frames are read one at a time and a SECDED ECC is performed [4]. If a single error within a frame is detected, the SEU DETECT (Figure 1) flag will go high, after the correction is calculated and the corrected frame is written into configuration memory, the SEU DETECT will go low. If there is a double bit error the ECC circuitry will raise the SCAN ERROR Flag (see Figure 1).

C. SEU Correction and Detection-Validation.

It is important to emphasize that the Xilinx SEU Controller can only correct one bit within a frame. Our objective is to validate SEU correction and investigate the accuracy of the double error detection circuitry. We are able to observe response by utilizing the error inject controls on the Xilinx SEU controller. We first inject single bit errors. All are corrected. When we inject double bit errors within one frame; all are detected and not corrected. Several variations of multiple bit injections are under investigation by varying the separation of bit addresses (thus varying error patterns within memory). We observe that all odd number injections are noted as corrected (SCAN_ERROR). However, this is an impossible event because the ECC implementation is SECDED (multiple bit errors can not be corrected). We also observe that 4 and 8 bit error injections go undetected. These results must be taken into account when analyzing MBU cross-sections while utilizing the Xilinx SEU controller.

D. Internal Scrubber Integrity

The Xilinx internal scrubber has 4 general methods of inoperability:

1. MBU causing the SECDED to either: not be able to correct and therefore potential fault accumulation can occur (within a frame) or improperly correct a frame thus creating massive interconnect errors
2. Internal scrubber circuitry gets hit and causes improper function or cease of function
3. Utilized BRAM (pico-blaze of internal scrubber uses BRAM) gets hit and causes malfunction.
4. ICAP interface becomes inoperable

We investigate the affects of these potential faults on the generally operability of a selected Design Under Test.

IV. NASA/GSFC REAG external scrubber

A. General Description

Due to the observed MBU data cross-sections [3], and the inability of SECDED ECC to handle MBU occurrences, the NASA/GSFC Radiation Effects and analysis group investigated alternative approaches to configuration memory mitigation. The external scrubber developed by NASA/GSFC does not use ECC circuitry in order to correct. Instead, a golden configuration is stored. An external device periodically overwrites the configuration memory through the Xilinx select map interface port with the golden information. In our case the external device is our tester. In a flight project, the external device would be a hardened FPGA. This is a scrubber therefore functionality is never disrupted unless a SEFI occurs. Because ECC is not used, performance is not limited by syndrome length and correction capability i.e. all errors within accessible configuration can be corrected in absence of a SEFI.

B. *External Scrubber Integrity*

Concerning the Xilinx device alone, there is one general method of External Scrubber inoperability: Interface circuitry malfunction (includes associated registers, counters, state machines, etc…)

V. Design under Test

The NASA-GSFC REAG group is responsible for testing many types of FPGA devices. In order to perform a direct comparison among FPGA error cross-sections, the windowed output shift register architecture was implemented as the Design Under Test (DUT) [7]. There were 3 types of shift registers each with 300 flip flops within the string (2 copies of each were implemented in the DUT):

1. Only Flip-flops within the string
2. Flip-flops with 8 inverters between each stage
3. Flip-flops with 20 inverters between each stage

Two categories of design architectures were implemented:

1. Shift registers (plus associated combinatorial string logic – 0, 8, and 20 inverters),
2. Shift registers (plus associated combinatorial string logic – 0, 8, and 20 inverters), including Internal Scrubber circuitry.

All shift registers were run at 100 MHz and all 6 strings are contained in one FPGA device.

VI. Test Results

A. *The Tester*

We constructed a daughter board for the LX25 Xilinx devices. The LX25 board connects to the NASA/GSFC Low Cost Digital Tester (LCDT) [5]. The LCDT provides clock, data, and reset inputs to the DUT. The LCDT is also responsible for capturing the outputs from the DUT and reporting and errors to the user-Host computer (refer to figure 2).

We obtained data at both the Texas A&M Cyclotron Institute (TAMU). Figure 3 illustrates Test Results from TAMU with 24.8 MeV/U heavy ion beam (Argon) @ 0 degrees of incidence and LET=5.7MeV*cm^2/mg . The Mal-functional cross-section was calculated as functional inoperability over fluence. The readback error cross-section was determined after reading the configuration memory (number of bits in error) post irradiation runs over fluence.

B. *Malfunction Categorization*

Figure 3 illustrates two types of categories of errors:

(1) Burst: Errors occurring for a long period of time

(2) Single Point: error occurring for one clock cycle of the DUT

A category 1 malfunction occurs when a configuration bit gets hit. It can only be corrected upon reconfiguration or scrubbing. In this case, the external scrubber corrected the bit(s) and normal operation resumed after 20ms. A single point failure occurs when the internal logic portion of the FPGA reverses its state due to a radiation strike and the effect is stored within a flip-flop. The implemented function within the DUT allows all single point failures to be overwritten by the next clock cycle (no enables are utilized). This analysis affects design strategy such as mitigation, logic utilization, and time specifications and must be taken into account for flight missions.

When the output of the DUT is stuck in a burst state, other potential errors will be masked. Due to the long duration of bursts, a true cross-section can not be accurately calculated by the traditional method of events divided by fluence. In this case we calculate time in burst per test and subtract an approximate number of particles that would affect the DUT during this period:

$$\sigma = \frac{NE}{TFL - (TB * Flux)}$$

NE: Number of Events

TFL: Total Effective Fluence

TB: Time in Burst

Flux: Approximate reported particle flux (particles/second)

It is important to note, that when we reduce the time in burst (configuration memory errors), the DUT approaches the behavior of an antifuse device (configuration is hardened). However, we do not eliminate the logic errors. They still exist and will be evident as illustrated by the single bit errors in the graph as in Figure 3.

We are performing dynamic testing and are investigating malfunction @ 100MHz with obvservability of every DFF within the device. The probability of incurring a fault is dependent on both the configuration memory and on the device logic. While calculating an error cross section per bit, care must be taken because the probabilities of configuration memory and of DFFs (logic bits) are not the same. In addition, as the technology scales down, multiple bit errors become significant. Simply normalizing the error by total bit count does not take this phenomenon into account. Because of the potential discrepancy, we chose to calculate error cross-section per design malfunction. In this case, errors can only be masked by bursts and are not masked due to complex functionality. The cross section can then be easily adjusted and normalized as demonstrated in section B.

C. Cross Section Analysis

Our error cross sections are considerably larger than most reported cross-sections for Xilinx Devices (figure 4). The following are 2 general explanations:

The difference in performance (error cross-section) between the external vs. internal scrubbers was not as great as expected. We assume this is a result of running tests until uncorrectable states occur (SEFI). This cross-section does not reflect time to SEFI, it reflects malfunction during operational time. Accordingly, although the cross-sections did not have a large difference in value, the external scrubbing was always recoverable without the need for a reset or power cycle (for our DUT design), whereas the internal scrubbing was never recoverable – i.e. faults occurred that were uncorrectable. Time to SEFI cross sections will be calculated. We have seen a notable difference in performance between the external scrubber and the internal scrubber within the FX60 Power PC tests performed by our group[8]. More data is currently being analyzed.

D. Resource Analysis

The NASA/GSFC Scrubber had the best performance as expected (because it is not dependent on MBU). The external scrubber incurred zero resource errors at the end of each external run as illustrated in Figure 3. The external scrubber wrote through BRAM for each test (the design did not use BRAM so this is a valid setting).

It is interesting to note that the readback data from the internal scrubber consistently had a higher count of resource errors. We believe this is due to the fact that we performed readback post test. The tests were terminated if the device entered an uncorrectable state. At this point, the internal scrubber may have written bad data into the frame after miscalculations from the SECDED algorithm.

Data is currently being analyzed from tests that contained intermediate readbacks during irradiation. This will enable the analysis to have a finer granularity of resource observation.

Conclusions

We have presented two Xilinx Device Configuration mitigation schemes (scrubbers) – one developed by Xilinx and one developed by NASA/GSFC. The Xilinx scrubber uses SECDED ECC to detect and correct errors. This scheme is inherently limited. We have observed the limitations during fault injection and monitoring false correction detection and absent error detection. The NASA/GSFC scrubber uses a golden configuration map with no ECC circuitry. In accordance to the fact that the NASA/GSFC scrubber has the ability to correct any number of errors it has been shown that the NASA/GSFC has improved performance over the Xilinx Scrubber.

Due to the high error cross-section, and the device consistently reaching uncorrectable states, we conclude that it is beneficial for a flight mission to consider both category 1and category 2 configuration mitigation strategies to enhance performance.

References

[1] [1] J.Benedetto .P. Eaton, and R. Davies, “An examination of Single Event Functional Interrupts (SEFIs) in COTS SDRAMs,” presented at 2006 SEE Symposium, Long-Beach, CA, Apr. 2006.

[2] [2] J. George, R. Koga, G. Swift, G. Allen, C. Carmichael, and C. W. Tseng, “Single Event Upsets in Xilinx Virtex-4 FPGA Devices,” IEEE NSREC 2006 Data Workshop record, pp. 109-114.

[3] [3] Paul Graham, Heather Quinn, Jim Krone, Michael Caffrey, Sana Rezgui and Carl Carmichael, “Radiation-Induced Multi-bit Upsets in Xilinx SRAM-Based FPGAs,” MAPLD, Washington DC, Sept. 2005

[4] [4] Les Jones XAPP714 (V1.5): “Single Event Upset (SEU) Detection and Correction using Virtex 4 Devices.”. Jan. 2007. http://www.xilinx.com/bvdocs/appnotes/xapp714.pdf

[5] [5] J.W. Howard, H. Kim, M. Berg, K.A. LaBel, et al. “ Development of a Low Cost and High Speed Single Event Effects Tester based on Reconfigurable Field Programmable Gate Arrays (FPGA),” Presented at SEE Symposium 2006, Long Beach, Apr. 2006.

[6] [6] J. George, R. Koga, G. Swift, G. Allen, C. Carmicheal, and C.W. Tseng. “ Single Event Upsets in Xilinx Virtex-4 FPGA Devices.” IEEE NSREC 2006 Data Workshop record. pp 109-114

[7] [7] M. Berg et.al. “An Analysis of Single Event Upset Dependencies on High Frequency and Architectural Implementations within Actel RTAX-S Family Field Programmable Gate Arrays,” *IEEE Trans. Nucl. Sci.*, vol. 53, n° 6, Dec. 2006.

[8] Christian Poivey, Melanie Berg,Kenneth A. LaBel, *et al.* “Single Event Effects (SEE) Response of Embedded Power PCs in a Xilinx Virtex-4 FPGA for a Space Application.”, to be presented at RADECS 2007, Deauville, Sept. 2007

[9] A. Lesea and K. Castellani-Coulie, “Experimental Approach of Soft Error Effects in 90 nm Xilinx FPGA and Beyond,” to be presented at RADECS 2007, Deauville, Sept. 2007.

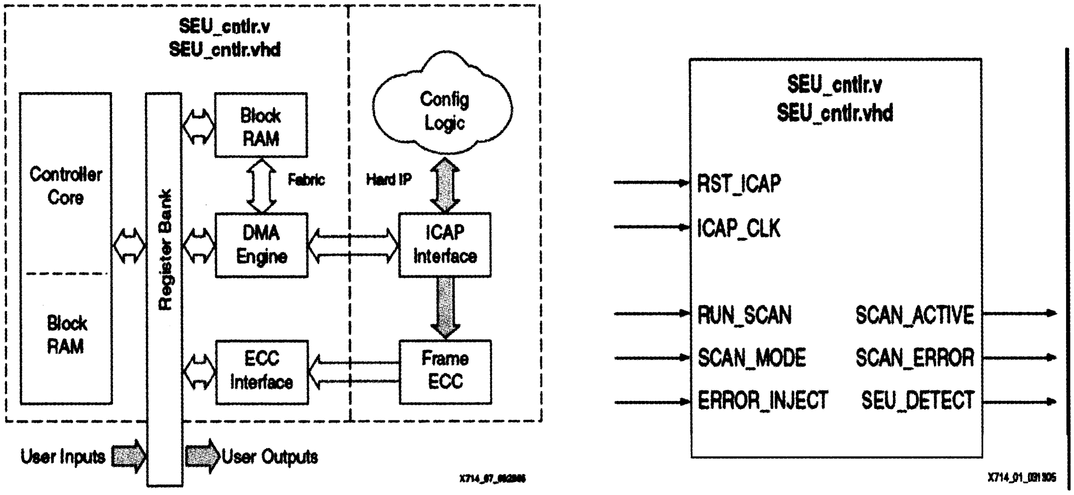

Figure 1: Xilnx SEU Controller Block Diagrams [4]: The Right Most Block Diagram Illustrates I/O for the Core

Table 1:Xilinx LX25 Utilization Charts for 2 Implemented DUT Designs

[illegible]			
Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	1829	21504	8%
Number of 4 input LUTs	16,625	21504	77%
Logic Distribution			
Number of occupied Slices	9841	10752	91%
Total Number 4 input LUTs	16625	21504	77%
Total equivalent gate count for design	114382		

LX25 shift register Device Utilization Table Summary			
Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	1829	21504	8%
Number of 4 input LUTs	16,625	21504	77%
Logic Distribution			
Number of occupied Slices	9841	10752	91%
Total Number 4 input LUTs	16625	21504	77%
Total equivalent gate count for design	114382		

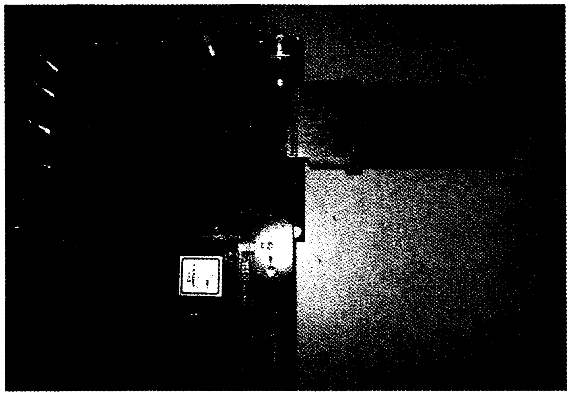

Figure 2: NASA/GSFC REAG LCDT with LX25 DUT

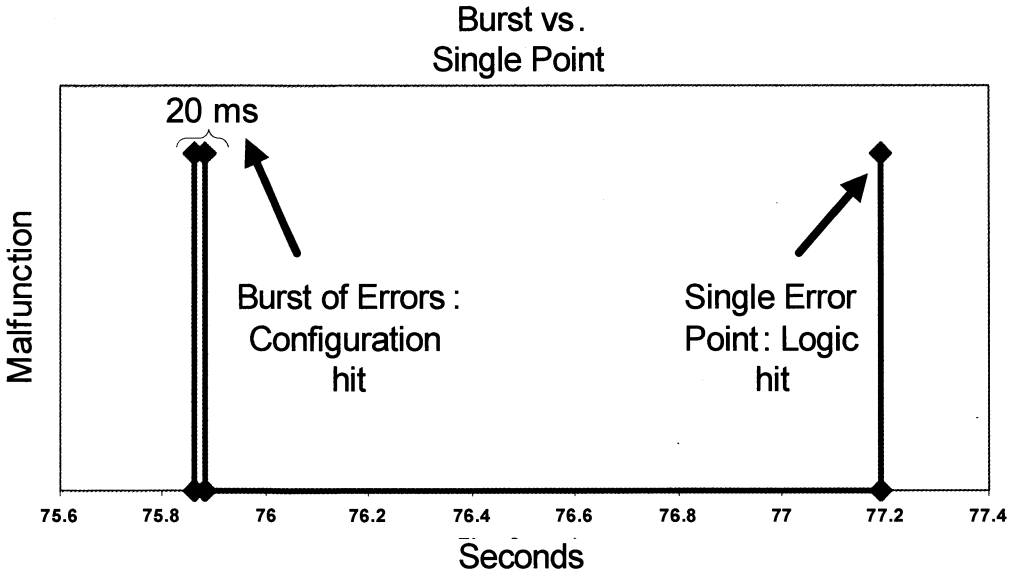

Figure 3: LDCT Captured Data: Burst and Single Point Failures during External Scrubbing Mode

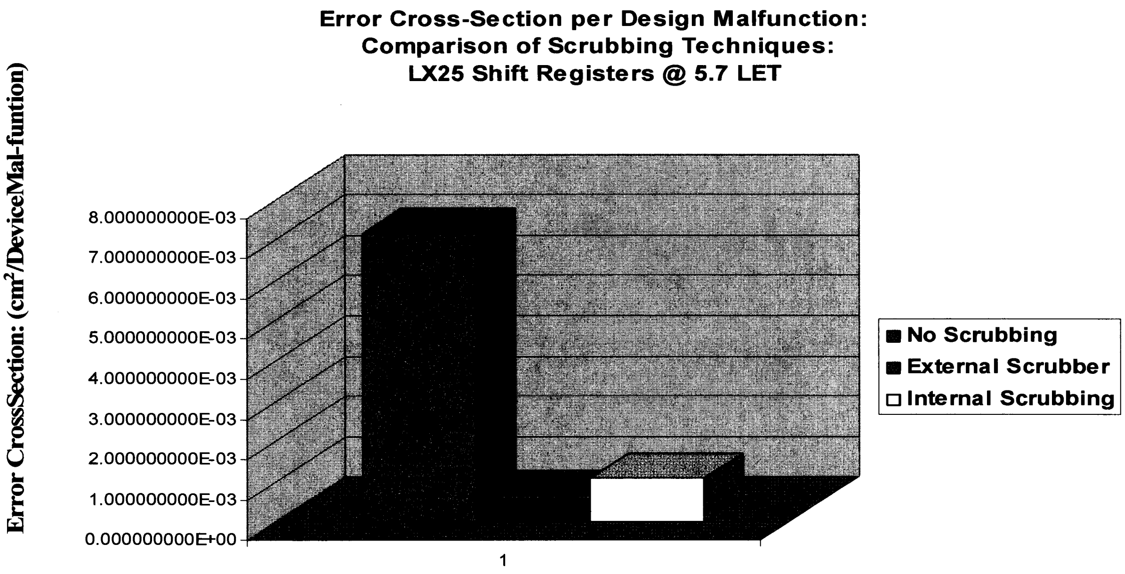

Figure 4: External Scrubbing vs. Internal Scrubbing @ 0 Degrees Incidence

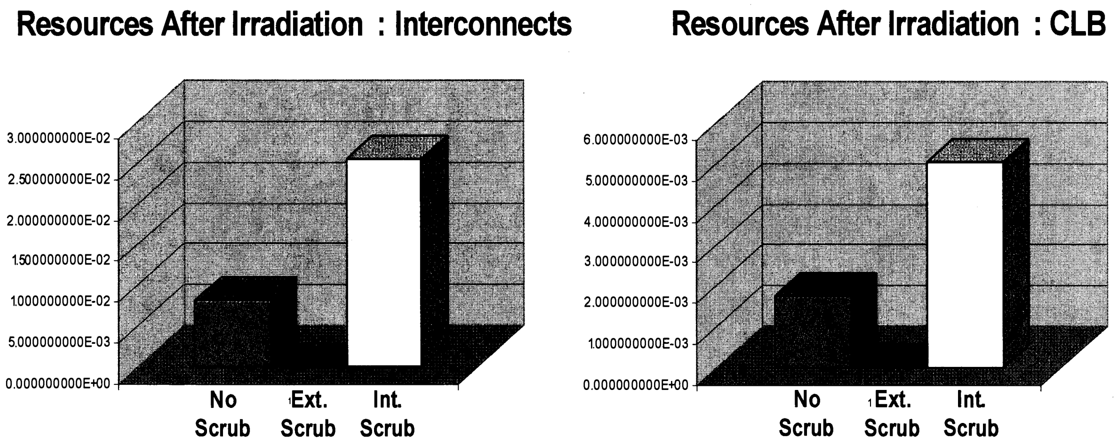

Figure 5: Comparison of Resources post-irradiation for No Scrubbing vs. Internal Scrubbing