Evaluation of SRAM based FPGA Performance by Simulating SEU through Fault Injection

Mohamed Mahmoud Ibrahim, Kenichi Asami, Mengu Cho
Integrated Systems Engineering Dept.
Kyushu Institute of Technology
KitaKyushu, Japan
mohamed@es.ise.kyutech.ac.jp

Abstract—This paper presents the technique and results of Single Event Upsets fault injection in the configuration bit-stream of SRAM-based FPGAs through partial reconfiguration of configuration frames. The Xilinx Virtex5 LX50 is used in the experiments. The Single Event Upset controller macro is used injecting faults to random locations of the FPGA bit-stream. The effects were studied on a design consisting of 4 embedded processor systems implemented in the FPGA. MATLAB is used for developing the external fault injection control environment.

Keywords—Single Event Upset; FPGA; Fault Tolerance; Fault Injection; Embedded Systems

I. INTRODUCTION

The space environment is characterized by having harsh conditions for operating electronics. The charged particles, cosmic rays, and heavy ions radiations have severe effects on their performance and lifetime. There exists a continuous need to develop systems that can operate flawlessly in the space environment while being close to the state-of-art technology. Another challenge is the cost of space approved systems. It is too high to be afforded through limited budget missions.

Small satellites being developed by many universities all over the world need to be of lower costs and sometimes of relatively high performance. This led to miniaturizing the space systems being developed for small satellite missions while using Commercial-Of-The-Shelf (COTS) components to reduce costs and have higher performance.

Field Programmable Gate Arrays (FPGA) are an optimum choice for miniaturization. They provide the capabilities to develop complete embedded systems with multicore processors in just one 5cmx5cm package. There exist two major types of FPGAs. The anti-fused and Static-Random-Access-Memory (SRAM) [1][2]. The anti-fused FPGA is usually used in One-Time-Programmable (OTP). It can be used in implementing systems that will not change the design after being implemented. It forms the ultimate choice for space systems that require the highest reliability. The SRAM-based FPGA is more flexible. It provides the opportunity to reprogram the FPGA with new designs even after being in-operation. The reprogramming of the FPGA design can take place remotely even without being connected to the development workstation. This feature give the SRAM-based FPGA an advantage of being flexible to accommodate design changes to improve performance and overcome detected design faults. The cost of developing a system with an SRAM-based FPGA is much lower than using the anti-fused FPGA.

Today's SRAM based FPGAs provide ultra large capacities to implement complicated logic circuits on different feature scales [2]. However, the use of such FPGAs in designing satellites' subsystems still introduces a risky challenge. The hardware logic that is implemented in the FPGA is stored in an internal SRAM memory as binary bit-stream. Any change in the binary bit-stream might mean that a change will happen in the logic circuits design itself. The challenge is in being able to effectively protect the bit-stream from being altered by the space environment charged particles and high energy cosmic rays radiations.

The bit-stream contains the configuration data that is loaded to the FPGA internal SRAM at booting up. The configuration data consists of internal Block Random Access Memory (BRAM) contents, Look Up Tables (LUT) contents and internal routing information between the FPGA Configuration Logic Blocks (CLB). The change of any of the bit-stream contents might lead, in some cases, to a serious change in the hardware design. Other changes might not have any affect at all on the running logic either because the fault does not propagate in the fault-mitigated design or the change takes place in an unused part of the FPGA. In this paper we simulate the Single Event Upsets (SEU) of the space environment at a LEO orbit on the Xilinx Virtex5 LX50 FPGA. The SEU rates are calculated using the Space Environment Simulator (SPENVIS) from ESA. The experiment for upsetting the bit-stream loaded in the internal FPGA SRAM takes place through injecting faults in it. The Fault Injection mechanism depends on upsetting randomly selected bits in the internal FPGA SRAM by toggling their binary values. The access to the internal bit-stream is provided through the use of an SEU controller. The SEU controller is a macro provided by Xilinx to monitor and correct SEUs. It makes use of the Internal Configuration Access Port (ICAP), an IP core in the Xilinx family that is used for reading back and writing of the bit-stream. The fault injection campaign takes place on a mitigated design. The mitigated system design consists of a feedback-based Triple Modular Redundancy (TMR) implementation of a multi-core embedded processors system and a system error controller. The system error controller performs the main functions of monitoring the operation of other individual

processor units, reset, recovery and reconfiguration of failed units. The selection of the bit-stream locations to be altered is based on uniform random distribution. The number of SEUs per a specific period of time, the SEU upset rate, depends on the target orbit parameters. The upset rate is probabilistic and can be modeled with a Poisson distribution. The interval between upsets is modeled in this work using uniform distribution within a period of time. Exponential distribution can be used for modeling the inter-upset events timing.

The distribution approximates the SEU phenomena expected to be in the real space environment. The total system error rates are then recorded against different values of the upset rates. The results can be used in expecting the SRAM FPGA performance when exposed to accumulated SEUs. The direct mitigation of the accumulated bit-stream upsets is through memory scrubbing. The scrubbing rate is directly related to the SRAM FPGA system reliability. Selecting the adequate scrubbing frequency can be estimated through the study of SEU simulations. The system makes use of the Xilinx-ISE and Xilinx-EDK, Embedded Development Kit (EDK), packages in developing the target platform. This work takes place at Kyushu Institute of Technology, Japan, to develop a robust SRAM-Based FPGA Avionics system for LEO satellites.

II. SINGLE EVENT UPSETS

The SEU occurs in the space environment due to the creation of electron-hole pairs when a charged particle of high energy passes through the silicon device of electronic components. The creation of the electron-hole pairs can be through direct ionization from heavy ions of high energy and/or nuclear reactions mostly from the trapped protons and electrons in the Van-Allen belts. The direct effect of this ionization would be the accumulation of charges to the limit that can alter the logic-state of any of the circuit nodes. The change in a logic-state from (0) to (1) or from (1) to (0) is known as the SEU.

The SEU belongs to the general classification of the Single Event Effects (SEE) which includes the Single Event Latch-up (SEL), the Single Event Gate Rupture (SEGR), the Single Event Burnout (SEB) and the Single Event Transients (SET). Some of these SEEs are destructive if they are not caught in the proper time as they might cause burnout for the device such as the SEL, SEB and SEGR. The SET is an upset that takes place for a tentative period of time as a glitch. It might or might not have a momentarily effect on the system functionality and usually disappears by itself. However, the SEU can not disappear by itself. Mitigation techniques should be applied to deal with it. There are two general categories of mitigation techniques: the prevention technique and the fault tolerance technique. Prevention means to avoid having SEUs at all through using mitigation by manufacturing of the electronic devices. The Silicon-On-Insulator is one example of the prevention techniques. Fault tolerance depends on mitigation by design. Fault tolerance is mainly about masking the effects induced by the SEU. Masking can be done by using redundancy. Three methods of redundancy are widely used: space redundancy, time redundancy and data redundancy. Space redundancy depends on adding additional units to the system to act as backup units. A voter is used to form a consensus among the redundant units' results. Time redundancy depends on repeating the execution of system functions and judging the correctness of operation through the use of acceptance tests. Data redundancy depends on adding additional information to the bits of data being protected to help in detecting and/or correcting errors such as the Error Correcting Code (ECC).

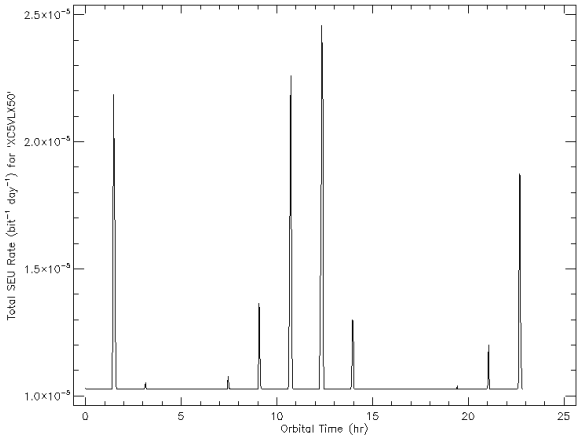
The estimation of the SEU upset rate depends on the orbital parameters. Mainly the orbit altitude and inclination. The parameters of typically target orbits for satellites being built by Kyushu Institute of Technology is shown in table 1.

TABLE I. TARGET ORBIT PARAMETERS

Orbit Parameter	Value
Altitude	671km
Inclination	98.17°
Eccentricity	0.00
Right. Ascension. of Ascending . Node	223.04°
Argument of Perigee	31.95°
True Anomaly	38.25°

The Cosmic Ray Effects on Micro-Electronics (CREME) model is used in assessing the SEU rate for space missions [3]. The SPENVIS web-based package is used in calculating the SEU rates based on the CREME model [4]. Fig. 1, shows the estimated SEU rate during 24 hours of flight for the orbital parameters mentioned in table 1.

Fig. 1. Single Event Upset Rate in 24 Hours of Flight.



The SEU estimation in Fig. 1, is based on the values of the radiation testing of the Xilinx Virtex 5 LX50 FPGA [5]. The threshold Linear Energy Transfer (LET) was in the order of 10MeV to 20MeV. The XC5VLX50 device proton bit cross-section is $8.6\text{x}10^{-14} \pm 4.90\text{x}10^{-16}$ cm^2/bit at 200MeV and $6.37\text{x}10^{-14} \pm 1.17\text{x}10^{-15}$ cm^2/bit at 65MeV [5]. The ions event bit cross-section at LET of (68.3Mev cm^2/mg) is $5.73\text{x}10^{-8}$ cm^2/bit. The Weibull fit parameters for the heavy ion event bit cross-

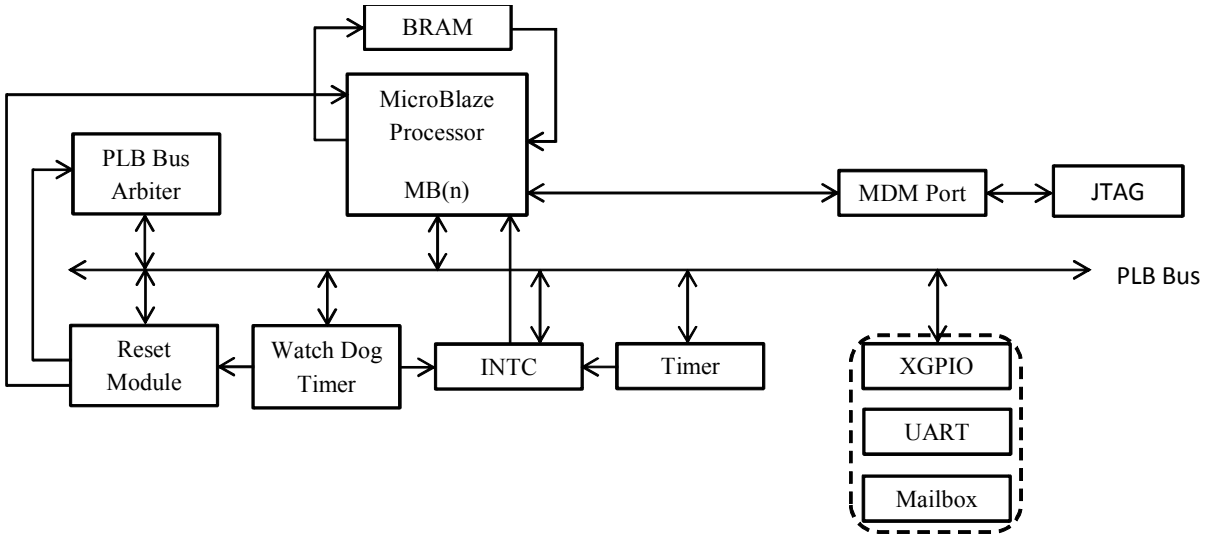
section were: (L=1.13e-7, L_0=0.5, S=1.5, W=30) [5]. These data were used in estimating the SEU rate for the commercial version XC5VLX50 at the target orbit parameters in table 1. The Aluminum shielding was set to 0.5 g/cm^2 and node dimensions in the FPGA device were set to (X=1µm, Y=1µm and Z=1µm).

III. SYSTEM ARCHITECTURE

The use of multicore systems in space applications would facilitate a lot of processing tasks. Therefore we designed a system that contained 4 complete embedded processor systems. The processor used was the Microblaze processor provided by Xilinx. The purpose of the fault injections is to evaluate the robustness of the Microblaze cores when used in a multicore design. The processor systems exchange data after processing to form a Triple Modular Redundancy with feedback system.

Fig. 2, shows the block diagram of the processor system. Each processor system runs the software application from a BRAM attached to the processor through the Local Memory Bus (LMB). The processor communicates with its peripheral IP-cores through the Processor Local Bus (PLB). An arbiter exists to coordinate the bus sharing. Daisy chain algorithm is used to grant the bus to any peripheral. However, the bus controller can be configured to operate in priority-based granting as well. The Interrupt Controller (INTC) is used to notify the Microblaze core of Watch Dog Timer (WDT) alerts. The WDT will provoke an Interrupt Service Routine (ISR) that would clear the WDT alert. If the alert is not cleared within two full rounds of the WDT then it will send a reset request to the reset module. The system reset mechanism is used to overcome software hang up. Another timer is used to provide interrupt-based clock ticks that can be used for triggering the clock tick needed by an operating system such as the xilkernel from Xilinx. The processor communicates with other peripherals that can be used for Input\Output such as the XGPIO for sending and receiving I/O signals. The UART is used for communication with external function monitoring computer. The mailbox is used to send and receive the results of processing among the processors for voting. Debugging is performed through the Microblaze Debug Module (MDM) that communicates with the debugging computer through JTAG/USB interface. The system as described and shown in Fig. 2, is instantiated 4 times inside the FPGA. Clocks are provided thorugh 1 Digital Clock Manager and 2 Phased Locked Loops.

Fig. 2. Embedded Processor Sytem Block Diagram.



The system is implemented in the XC5VLX50 from Xilinx. The utilization of the FPGA resources are shown in table 2.

TABLE II. FPGA RESOURCES UTILIZATION

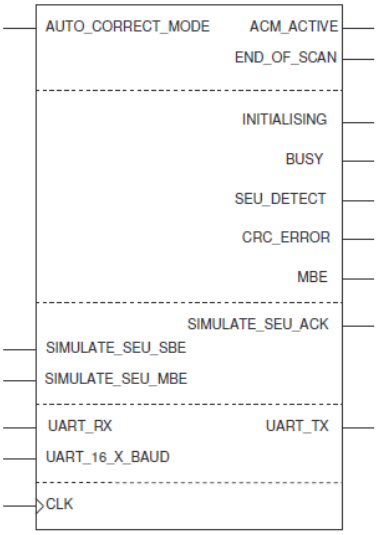
Parameter	Used	Utilization
Occupied Slices	6212	86%
Slice Registers	12716	44%
Slice LUTs	15022	52%
Slice LUTs used as logic	13908	48%
Slice LUTs used as memory	1029	13%
Total Memory Used	1278	59%

The fault injection takes place through the SEU controller Macro from Xilinx [6]. The Macro is used to facilitate the insertion of bit flips at FPGA frames in two control modes: the UART control and signals control. The UART control mode is used to send commands to the SEU controller through UART interface. The commands are sent as strings followed by the command arguments. For example the toggling command is sent as a string which starts with the character ‘t’ followed by the 6 hexadecimal digits of the frame address and then the 3 hexadecimal digits of the bit location to be flipped. It might look as ‘t000b063b2’.

The XC5VLX50 device consists of 8663 frames. The frames that are usable are from 1 till 8662. Frame 0 is not used. Each frame contains 1312 bits organized in the form of 41 words and each word contains 32 bits. In each frame 16 bits are unused from bit number 656 till bit number 671.

Fig. 3, courtesy of Xilinx, shows the external interfaces of the SEU controller macro [6]. Besides the two previously mentioned control modes there are two operation modes: the Auto Detection Mode (ACM) and the Detection Only Mode (DCM). The ACM is used to correct Single Bit Errors (SBE) as soon as it is detected in the scanning of the FPGA configuration frames. The average time to detect a fault would be one complete scan cycle time which is 6.78ms for the XC5VLX50 [6]. If Multiple Bit Errors (MBE) occurs the ACM can not correct it and the FPGA should be reset and reconfigured again. The DOM is used when testing the accumulation of errors during fault injection campaigns. The fault injection mechanism depends on reading the frame where fault should be injected then flipping the bit location selected for injecting the fault. The corrupted frame is then written back to the configuration memory and performance is monitored.

Fig. 3. SEU Controller External Interfaces, Courstey of Xilinx.

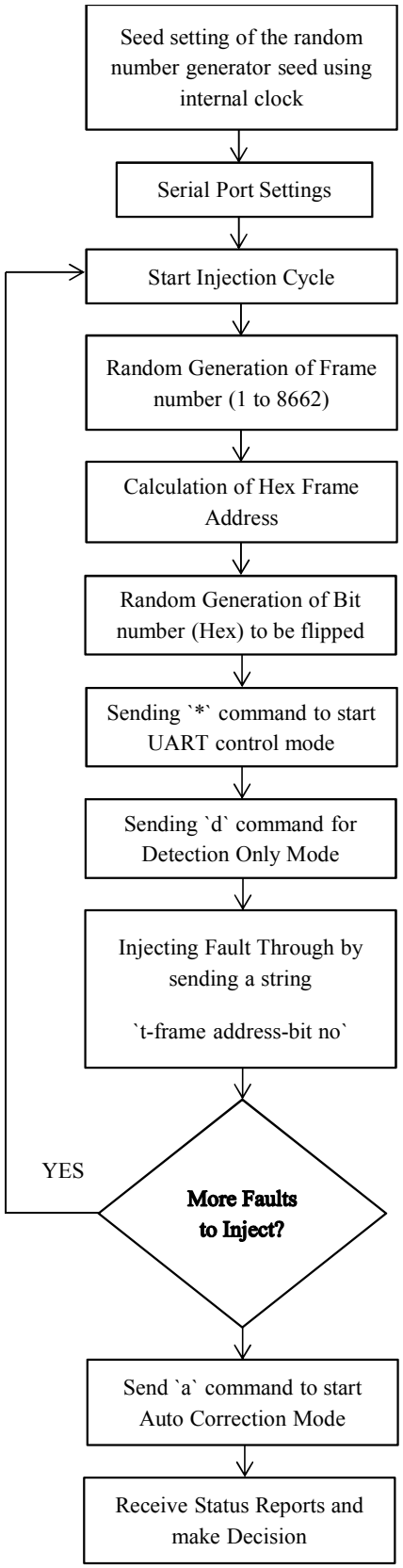


In the UART control mode only the UART=RX and UART_16_X_Baud signals are used. They are connected to the external fault injection controller which runs as a MATLAB script on an external workstation.

IV. Fault Injection Methodology

The fault injection of SBU is performed through an externally connected computer which directly communicates with the SEU controller. Fig. 4, shows the flow chart for injecting faults. The number of faults to be injected as well as the times to repeat the test can be set. Intervals between sending the 't' commands are random. Their distribution can be of uniform or exponential. The number of SEUs to be injected is estimated from the graph in Fig. 1.

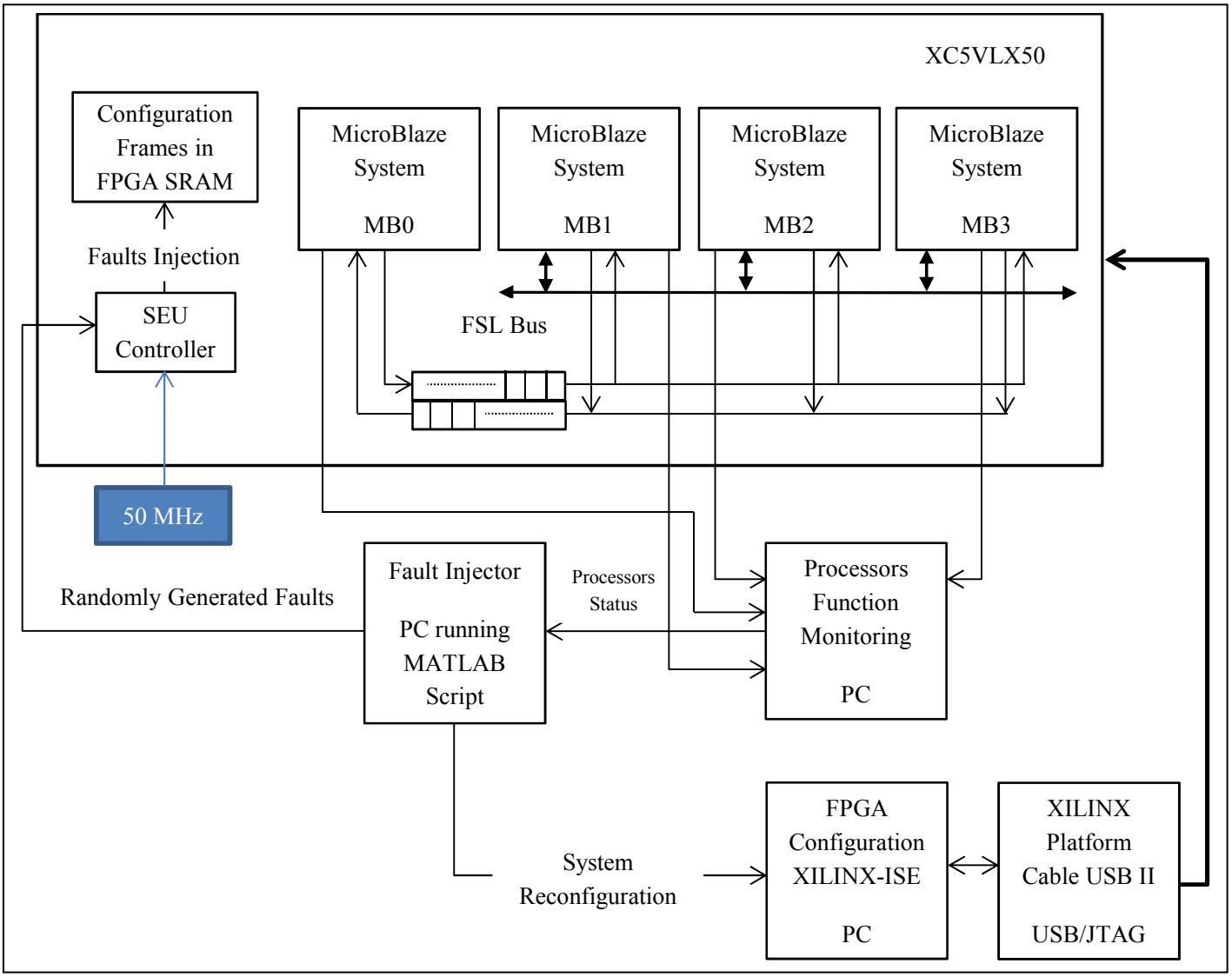
Fig. 4. Fault Injection Flow Chart



At the target orbit listed in table 1, the maximum upset rate is $2.5x10^{-5}$ (bit^{-1} day^{-1}) at the South Atlantic Anomaly regions. The minimum upset rate is $1.0x10^{-5}$ (bit^{-1} day^{-1}). The total number of configurable words in the XC5VLX50 device is 392,124 words [7]. The size of each word is 32 bits. Therefore the total number of configurable bits is 12,547,968 bits. The maximum number of upsets per device in the worst case upset rate would be (12,547,968 x $2.5x10^{-5}$ = 313.7 upsets). The minimum number of upsets per device in the lowest upset rate would be (12,547,968 x $1.0x10^{-5}$ = 125.48 upsets). These numbers of upsets are calculated for the full device bits being configured. In practice only part of the device bits will be configured and used in the design depending on the design details and size. In our case the total occupied slices as in table 2 are 86% of the total available slices. Each slice in the FPGA contains LUTs and registers The LUTs are used to implement combinational logic functions. Some LUTs can be used to implement memory elements as well. The registers can be used as Flip Flops (FF) in implementing sequential logic. They can also be used as latch-through.

When we analyze the actual number of slice registers and LUTs being used in our design, we find that they are around the range of 50% of the total number available in the FPGA. Therefore we estimate the total number of accumulated upsets per day, to be injected in our design, as (125.48/2 = approx. 63 upsets) in the normal regions of the orbit and (313.7/2 = approx. 157 upsets) in the peak regions of the orbit. Nevertheless, in fault injection any upset rate can be used as long as it simulates the real mission worst case scenarios for scanning the FPGA and\or conducting a scheduled maintenance [8]. The system was tested using two upset rates, 50 and 100 upsets per day. Fig. 5, presents the system setup while conducting the fault injection experiments.

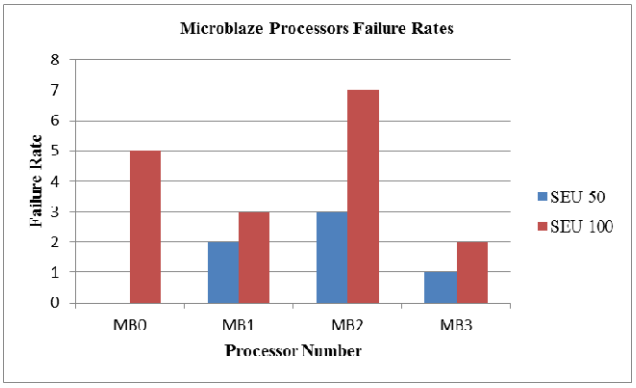
Fig. 5. Experiment Setup while Injecting Faults.



V. FAULT INJECTION RESULTS

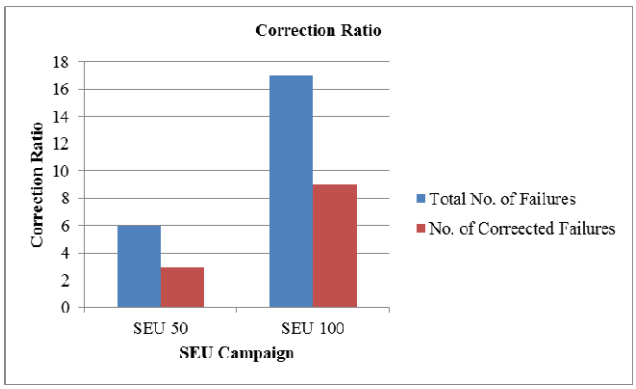
The fault injection was conducted in 10 times at two upset rates, 50 upsets per day and 100 upsets per day. In both cases about 10% of the injection developed into total system failure which required deep reset of the FPGA and reloading of the bit-stream. Other cases involved the upset of one or more processors; however, when applying the ACM mode, the operation was restored. These processors who stopped and then resumed working after applying the ACM needed software resynchronization with unstopped processors to operate in the same state. In some cases a processor stopped and was not able to retrieve its operation even after applying the ACM mode. These cases required partial reconfiguration by rewriting the processor bit-stream and not the full bit-stream. Whenever the status of the SEU controller indicated an MBE, the system was reset as the ACM can not work later until resolving that situation. Fig. 6, shows the Microblaze processors failure rates during the SEU 50 and SEU 100 tests.

Fig. 6. Microblaze Processors Failure Rates.



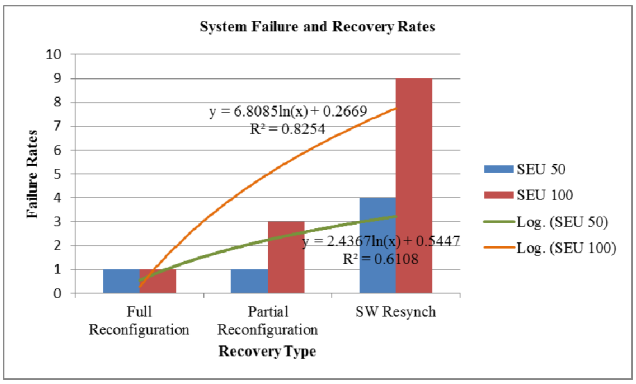
In both cases of simulations 10 runs were conducted, it needs to be increased in subsequent testing of the system, and each run contained 50 upsets or 100 upsets. The total number of injected faults in the SEU-50 campaign was 500 upsets in 10 runs, while in the SEU-100 campaign they were 1000 upsets in 10 runs. The correction ratio is depicted in Fig. 7. It is almost 50% in both cases.

Fig. 7. Correction Ratio Compared to Total Number of Failures



The overall system failure rate is shown in Fig. 8, the logarithmic trend line is illustrated with its equation and the R^2 value to show how well the fitting is. The full reconfiguration is caused due to an unrecoverable failure that could not be resolved by the ACM of the SEU controller. In that failure all of the processors totally stop and can not resume until the system is deeply reset and reconfigured once more. The partial reconfiguration takes place when a processor or more stops but can be retrieved through partial reconfiguration. A crucial situation is when the processor assigned to read the configuration bit-stream, to perform the partial reconfiguration, stops operation. At that moment the system must be reset and reloaded again. The software resynchronization takes place whenever a processor stops operation and then resumes back after ACM corrections or after it has been reloaded through partial reconfiguration. Resynchronization is needed to resume operation from the same point as the rest of the processors.

Fig. 8. System Failure and Recovery Rates



VI. CONCLUSION

The XC5VLX50 FPGA showed an appropriate performance when faults were injected up to 50 upsets per day. Higher upset rates would certainly cause an effect that can be noticed clearly and quickly such as stopping of one of the processor cores. More statistics and test cases are needed to fully judge the expected performance of the XC5VLX50 FPGA in space.

REFERENCES

[1] Actel website: http://www.actel.com

[2] Xilinx website: http://www.xilinx.com

[3] Adams, J. H., Jr., Cosmic Ray Effects on MicroElectronics, Part IV, NRL Memorandum Report 5901, 1986.

[4] SPENVIS website: http://www.spenvis.oma.ba

[5] Quinn, H.; Morgan, K.; Graham, P.; Krone, J.; Caffrey, M.; , "Static Proton and Heavy Ion Testing of the Xilinx Virtex-5 Device," Radiation Effects Data Workshop, 2007 IEEE , vol.0, no., pp.177-184, 23-27 July 2007.

[6] Ken Chapman, "New Generation Virtex-5 SEU Controller," Xilinx, Version A.2 – s4th November 2009.

[7] Xilinx, Virtex-5 FPGA Configuration User Guide, Xilinx UG191 (v3.10), 2011.

[8] Ken Chapman, "SEU Strategies for Virtex-5 Devices," Xilinx, XAPP864, April, 2010.