Reconfigurable Fault Tolerant Avionics System

Mohamed Mahmoud Ibrahim, Kenichi Asami, Mengu Cho
Kyushu Institute of Technology
1-1 Sensui-cho, Tobata-ku, KitaKyushu-shi, Fukuoka, 804-8550, JAPAN
+81-080-4270-0641
mohamed@es.ise.kyutech.ac.jp

Abstract—**This paper presents the design of a reconfigurable avionics system based on modern Static Random Access Memory (SRAM)-based Field Programmable Gate Array (FPGA) to be used in future generations of nano satellites. A major concern in satellite systems and especially nano satellites is to build robust systems with low-power consumption profiles. The system is designed to be flexible by providing the capability of reconfiguring itself based on its orbital position. As Single Event Upsets (SEU) do not have the same severity and intensity in all orbital locations, having the maximum at the South Atlantic Anomaly (SAA) and the polar cusps, the system does not have to be fully protected all the time in its orbit. An acceptable level of protection against high-energy cosmic rays and charged particles roaming in space is provided within the majority of the orbit through software fault tolerance. Check pointing and roll back, besides control flow assertions, is used for that level of protection. In the minority part of the orbit where severe SEUs are expected to exist, a reconfiguration for the system FPGA is initiated where the processor systems are triplicated and protection through Triple Modular Redundancy (TMR) with feedback is provided. This technique of reconfiguring the system as per the level of the threat expected from SEU-induced faults helps in reducing the average dynamic power consumption of the system to one-third of its maximum. This technique can be viewed as a smart protection through system reconfiguration. The system is built on the commercial version of the (XC5VLX50) Xilinx Virtex5 FPGA on bulk silicon with 324 IO. Simulations of orbit SEU rates were carried out using the SPENVIS web-based software package.**

TABLE OF CONTENTS

1. INTRODUCTION 1
2. ERROR RATE ANALYSIS 1
3. PARTIAL RECONFIGURATION.....8
4. AVIONICS SYSTEM DESIGN 8
5. SIMULATION RESULTS.....10
6. CONCLUSION.....10
REFERENCES.....11
BIOGRAPHIES.....11

1. INTRODUCTION

The avionics system is a crucial core of any spacecraft. It is typically responsible for implementing the functions of commands and data handling, orbit propagation and navigation, and attitude determination and control. The development of a reliable spacecraft avionics system was traditionally aligned with the choice of high-reliability space-approved components. Nevertheless, a change in the design paradigm was introduced about one decade ago. The new paradigm becomes widely accepted as it encourages the use of Commercial-Of-The-Shelf (COTS) components to develop satellites of acceptable reliability at reasonable cost.

The development of avionics systems based on COTS made use of Field Programmable Gate Array (FPGA) technology [1]-[3]. The FPGA technology itself has radiation hardened space-approved components and commercial/industrial-grade components. The FPGA can be of two major classes: One-time programmed FPGA such as anti-fused types and re-programmable FPGA such as SRAM-based types [4, 5]. Satellite developers generally prefer to use anti-fused FPGAs to build highly reliable systems that can withstand the radiation in the space environment. Nowadays, modern reprogrammable FPGAs used in space missions have proven to function properly [3].

In this paper we introduce the use of partial reconfiguration technology of Xilinx SRAM-based FPGAs to mitigate the avionics system against SEUs while in orbit. The use of the partial reconfiguration technology facilitates the reduction of the overall power consumption as described later. This technique provides an adaptive response to the level of threat faced by the avionics system. It increases the chances to build a reliable system while keeping the power consumption profile at affordable margins.

In section 2, we present the estimation for the charged particles and the resulting SEU at a targeted orbit for a specific satellite mission. In Section 3, we present the concept of partial reconfiguration and the use of reconfigurable FPGAs. Section 4 presents adaptive reconfiguration of FPGAs based on orbital position and SEU level. Section 5 presents the results of applying the adaptive reconfiguration technique on the system average power profile. Section 6 presents the conclusion and future work.

2. ERROR RATE ANALYSIS

The SEU rate depends on the orbital parameters of the spacecraft. It can be calculated using specific models such as Creme96 [6, 7]. The sources of SEU can be due to charged particles in terms of trapped electrons and protons, high-energy galactic cosmic rays, heavy ions, solar flares, and neutrons. Practically speaking, any high-energy particle can lead to ionization of the substrate material and hence creation of electron-hole pairs along its path leading to Single Event Effects (SEE). They can be due to direct ionization by the high-energy particle or as a result of

978-1-4673-1813-6/13/$31.00 ©2013 IEEE

secondary emissions from spacecraft material and/or the component itself. The SEU rate was found to be high at specific orbit locations due to the higher flux of particles at these positions.

For the purpose of illustrating the idea, a target orbit was selected to demonstrate the estimations related to the SEU and particles flux. That orbit was selected as it matches the Horyu-2 satellite. It was developed by the Kyushu Institute of Technology and launched from Tanegashima Space Center on 18th of May 2012 [8]. Table 1 shows the main orbit parameters for the selected target orbit. It is a semi-circular orbit with high inclination for earth observation purposes.

Table 1. Orbit Parameters

Parameter	Value
Apogee	671.60 km
Perigee	651.00 km
Inclination	98.17°
Right. Ascension. of Ascending . Node	223.04°
Argument of Perigee	31.95°
True Anomaly	38.25°
Eccentricity	0.00

The orbital parameters as a function of time are shown in Figure 1. The orbit period is 1.63 hours.

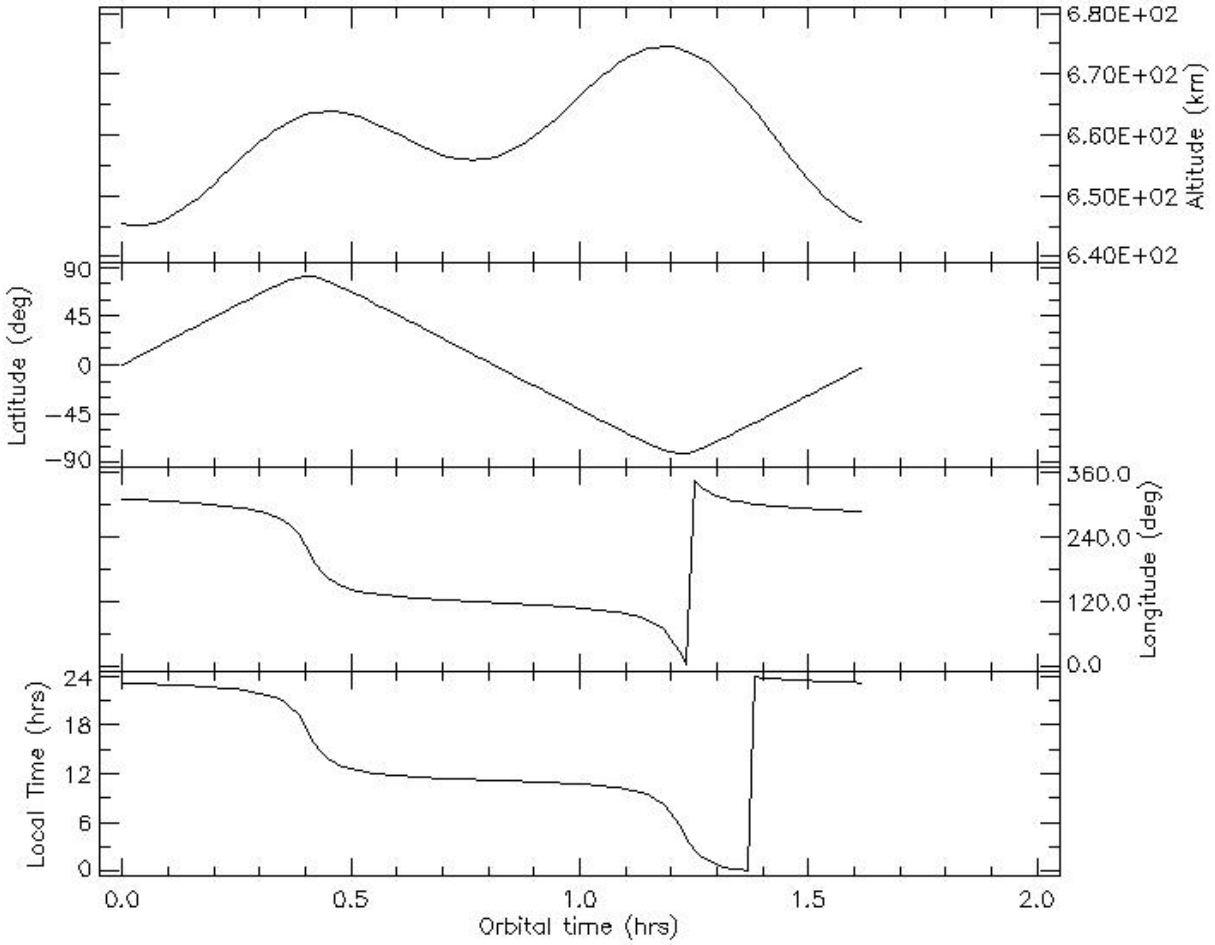


Figure 1 – Orbital Parameters as a Function of Time

The satellite in its orbit suffers from trapped proton and trapped electron fluxes which contribute to the majority of SEUs in its electronics. Figures 2 and 3 show the trapped electron and proton fluxes, respectively, as per the AE-8 and AP-8 models using the SPENVIS package [6].

The trapped electrons flux is represented for electrons at energy higher than 0.04 Mev. The trapped proton flux is represented for protons at energy higher than 0.1 Mev. The energy threshold at which upsets might start to happen depends on the cross-section function of the device under test. Cross section measures (σ) for the wide range of Linear Energy Transfer (LET) (ΔE/Δd) are conducted for the FPGA used in implementing this reconfigurable system. The neutron cross section per bit for 65nm Virtex-5 FPGAs is 6.70×10^{-15} cm^2/bit [9]. The threshold LET is on the order of 10 to 20 MeV. The proton event bit cross-section for the commercial version of the XC5VLX50 is $8.6 \times 10^{-14} \pm 4.90 \times 10^{-16}$ cm^2/bit for 200 MeV and $6.37 \times 10^{-14} \pm 1.17 \times 10^{-15}$ cm^2/bit for 65 MeV [10]. The event bit cross-section for ions were measured as well in [10]. The event bit cross-section for the highest tested LET of 68.3 Mev-cm^2/mg is $5.73 \times 10^{-8} cm^2$/bit. The Weibull fit parameters for the heavy ion event bit cross-section were: (L= 1.13e-7, L_o=0.5, S=1.5, W=30) [10]. These data were used in simulating the SEU rate for the commercial version XC5VLX50 at the orbit of Horyu-2 satellite with Aluminum shielding of 0.5 g/cm^2 and dimensions (X=1μm, Y=1μm and Z=1μm). Figure 4 shows the trapped proton shielded flux for energies higher than 30 Mev with shielding thickness of 0.5 g/cm^2.

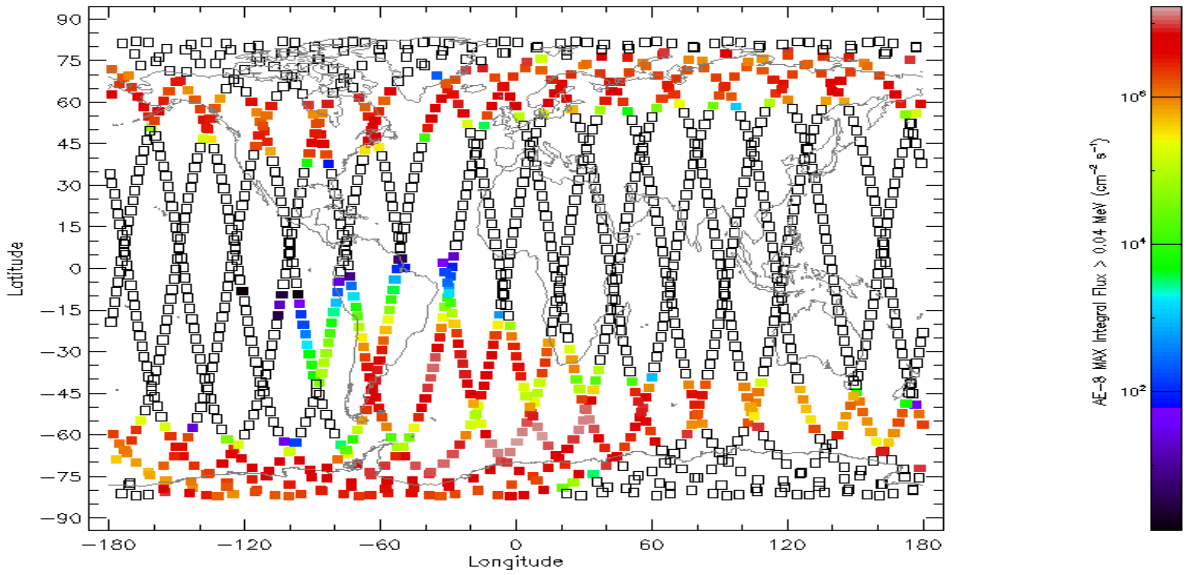


Figure 2 – Trapped Electron Flux

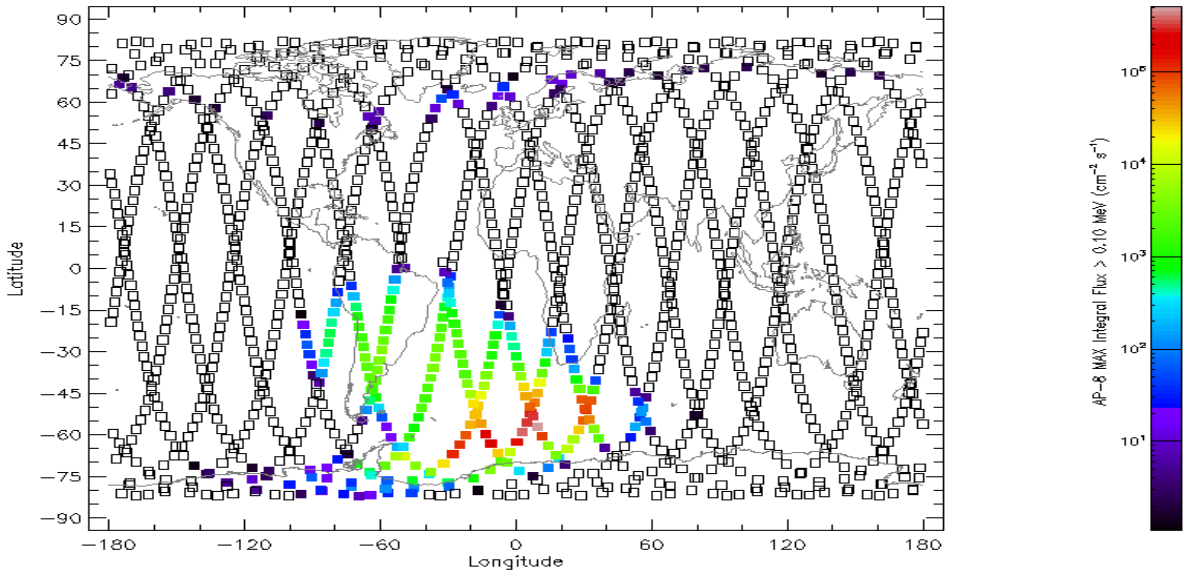


Figure 3 – Trapped Proton Flux

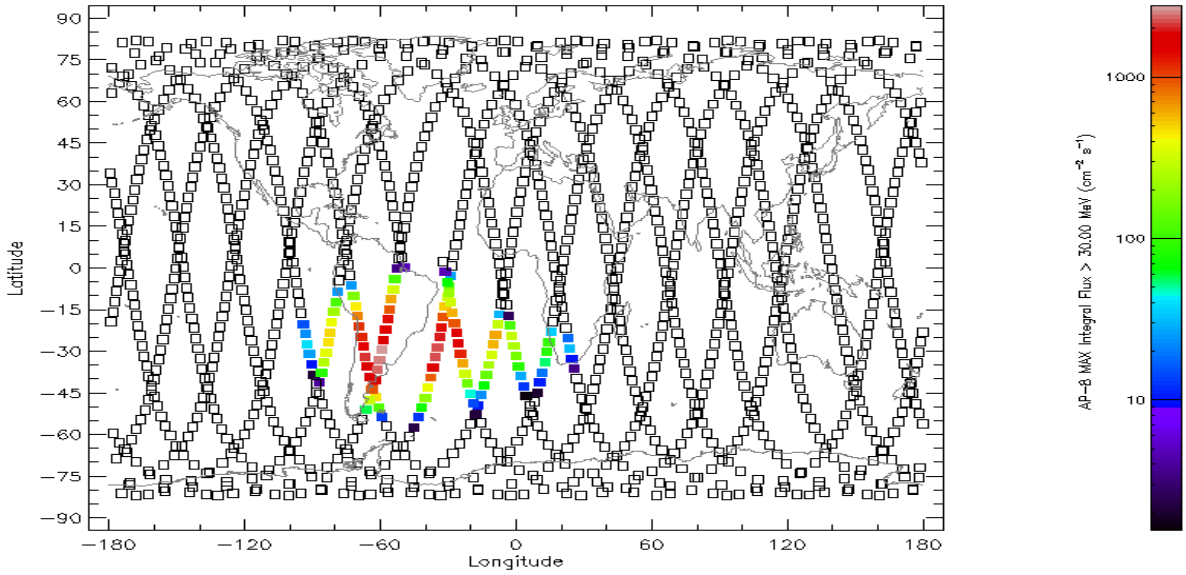


Figure 4 – Trapped Proton Flux at Energy>30 MeV

Figure 5 (a, b) show the average trapped electron and proton spectra. Aluminum shielding of 0.5 g/cm^2 was applied for in that simulation. Electron energies are less than 7 MeV in the selected orbit. Therefore trapped electron effect is almost negligible for causing SEUs in this case. However, trapped proton with energies higher than 30 MeV can penetrate the shielding with the fluxes represented in Figure 5 (b). This is a main source of SEUs along with solar-protons and high-energy galactic cosmic rays.

The average SEU rate can be estimated at different orbital positions as shown in Figure 6 (a, b). It is important to notice that SEU rate forms a peak at the SAA. The upset rate changes from the lowest value of 1.02×10^{-5} bit^{-1}day^{-1} to the highest value at 2.45×10^{-5} bit^{-1}day^{-1}. This increment gets more significant when device configuration bits gets larger. Thus for large SRAM based FPGAs that increment in the upset rate would have a crucial effect. The high upset rate duration is on average 15 minutes for the orbits passing over the SAA.

It is clear that the SAA is one of the severe radiation regions which need special handling in terms of fault tolerance design. The most commonly considered technique when strict fault tolerance is needed would be spatial redundancy through replicating the processing units. The common architecture for that is the TMR.

The Polar Regions also form locations of higher fluxes of low-energy trapped electron. This is clear in the trapped electron flux in Figure 2. However the threat from high-energy trapped protons over the SAA is much serious. The flux levels of protons and electrons passing at the same orbital location are different. Protons tend to be more stationary while electrons are more mobile. Therefore more low-energy electron flux is expected than high-energy proton flux as shown in the simulation results.

The idea that we present is based on making use of the partial reconfiguration concept in modern FPGAs to reload redundant processing modules when passing over high-energy proton regions. This would improve overall system reliability while passing over regions of higher threats. A TMR system will be formed just when passing over the SAA while continuing to use the single processing system for the rest of the orbit. This operation would significantly reduce the system power profile.

To inform the system about the areas of higher threat, one of three ways can be used:

1. using an onboard orbit propagator to estimate the orbital location and compare it with a stored look-up table for the threat coordinates.
2. Using a GPS that would feed the coordinate information directly to the avionics system which needs to compare it with the stored location for threat coordinates.
3. Through an external sensor that would allow the system to sense the shower of particles that attack the satellite and would start reconfiguring based on its judgment.

The problem with the first two methods is the need to store additional information in the form of look-up table in the avionics system memory which consumes storage area. The look-up table itself would need to be updated as the orbit parameters change due to perturbations. The use of an on-

board orbit propagator consumes higher computational cycles and needs to be updated with accurate reference to reduce error accumulation.

The third technique is better in terms of enabling the system to learn and adapt to the environment in which it operates. However This technique needs a smart system that can learn through its distributed charged-particle sensors located all over the satellite structure. It is more complex to implement it than simple clock-based switching.

The system error rate can be estimated based on the SEU rate. The FPGA stores the configuration bit-stream in its internal SRAM in the form of configuration frames. The frames consists of words each of which consists of 32 bits. The XC5VLX50 configuration array contains 392,124 words equivalent to 12,547,968 bits [11].

The configuration bits inside the FPGA SRAM need to be checked against bit upsets. This technique is called scrubbing where the configuration bits in each frame are read then a Cyclic Redundancy Check (CRC) code is calculated for them. The calculated CRC is used to generate an error syndrome that would indicate if the frame contains errors or not. If no errors exist then the next frame is read. If an error is found then it would be corrected if it was single bit error. The corrected frame would be written back to the FPGA without interrupting its operation. In case the error was in more than one bit, the error control module would issue a system reset or fetch the frame from an external flash memory where the original bit-stream is stored.

The overall system upset rate is calculated from Equation (1).

$$\text{Total Upset Rate} = \text{Bit-stream Size} \times \text{SEU Rate} \quad (1)$$

In case of non-triplicated design, two Microblaze processors with additional peripherals are used. One Microblaze is used for FPGA scrubbing and error control while the other Microblaze is used for implementing avionics system functions. In the triplicated system, four Microblaze processors are used. one for scrubbing and error control and the other three for TMR scheme.

In the non-triplicated mode, about 40% of the FPGA slices are utilized. That roughly corresponds to 5,019,187 configuration bits. In the triplicated mode, about 80% of the FPGA slices are utilized. That roughly corresponds to 10,038,374 configuration bits. Table 2 shows the total system upset rate at two SEU rates. The nominal SEU ($1.0\text{x}10^{-5}$ $\text{bit}^{-1}\text{day}^{-1}$) and the peak SEU ($2.5\text{x}10^{-5}$ $\text{bit}^{-1}\text{day}^{-1}$). The system error rate in case of the non-triplicated mode would be the same as the total upset rate in the worst estimation. The reason is that the upset event is not known whether it would have an effect or not. So for the worst case we consider that each upset would lead to an error in the non-triplicated mode.

Mitigation against induced upsets and generated system errors takes place through full FPGA scrubbing. The scrubbing rate is recommended to be 10 times the upset rate [12]. For a scrubbing clock of 100MHz, taking into consideration that one configuration word needs 1 clock cycle for retrieving, the XC5VLX50 would be configured in approximately 3.92msec. Therefore the maximum scrubbing frequency would be 255 scrub/sec or 22,032,000 scrub/day which is far much higher than all the upset rates in Table 2. When the non-triplicated system operates in the majority of the orbit, the error rate is considered to be 50.19 errors/day. This is equivalent to upset rate as described earlier. The error rate follows Poisson distribution while the interval between errors follows exponential distribution [13, 14]. For simplicity of the analysis we would consider equal time intervals between error events. Thus the time between two consecutive errors would be (86400/50.19 = 1721.458 sec). The scrubbing rate would be (10 x 50 = 500 scrub/day), which means one full scrub every 172.8 sec. The system reliability is estimated from:

$$R = e^{-\lambda T} \quad (2)$$

$$\lambda = 1 \text{ failure per } 1721.458 \text{ sec} = 5.809029\text{e-}4 \text{ failure/sec} \quad (3)$$

$$T = 172.8 \text{ sec(scrubbing cycle time)} \quad (4)$$

$$R = e^{-0.0005809029 \times 172.8} = 0.9045 \quad (5)$$

The non-triplicated system reliability at the scrubbing rate of 500 scrub/day will be 90.45%. If the scrub rate decreases the reliability would decrease as well. For example a scrub rate of 250 scrub/day means that one full scrub is executed every 345.6 sec. Thus (T = 345.6 sec) while (λ = 0.0005809029 failure/msec) therefore (R = 0.8181).

In the triplicated mode where processors vote for a consensus on their outputs and state codes, the system reliability is calculated as:

$$R_{TMR} = 3R^2 - 2R^3 \quad (6)$$

To calculate the improvement in reliability when using triplicated design in the peak SEU regions, we first calculate ($R_{\text{non-triplicated}} = e^{-0.001446759 \times 69.12} = 0.9048$). Where the failure rate (λ = 0.001446759 failure/sec) and the scrubbing cycle (T = 69.2 sec). If we apply the same failure rate for each processor on the triplicated mode and calculate the reliability then we have:

$$R_{\text{Triplicated-mode}} = 3\text{x}0.9048^2 - 2\text{x}0.9048^3 = 0.9745 \quad (7)$$

Table 2. System Upset Rates

	Total Upset Rate (upsets per day) Nominal SEU	Total Upset Rate (upsets per day) Peak SEU
Non-triplicated	50.19187	125.479675
Triplicated	100.38374	250.95935

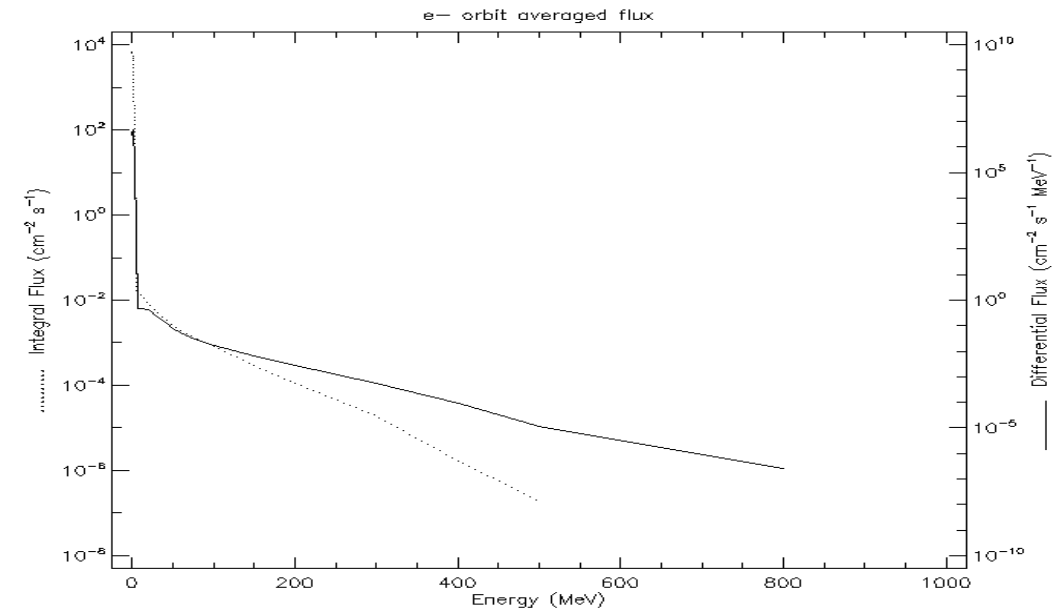


Figure 5a – Average Trapped Electron Spectra using Aluminum Shielding of 0.5g/cm^2

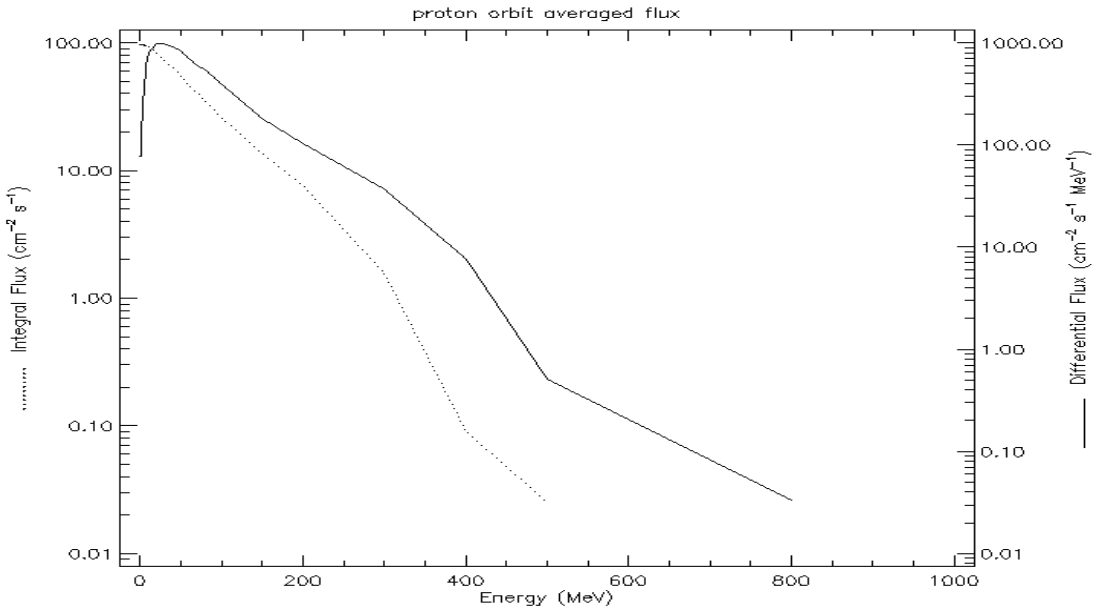


Figure 5b – Average Trapped Proton Spectra using Aluminum Shielding of 0.5g/cm^2

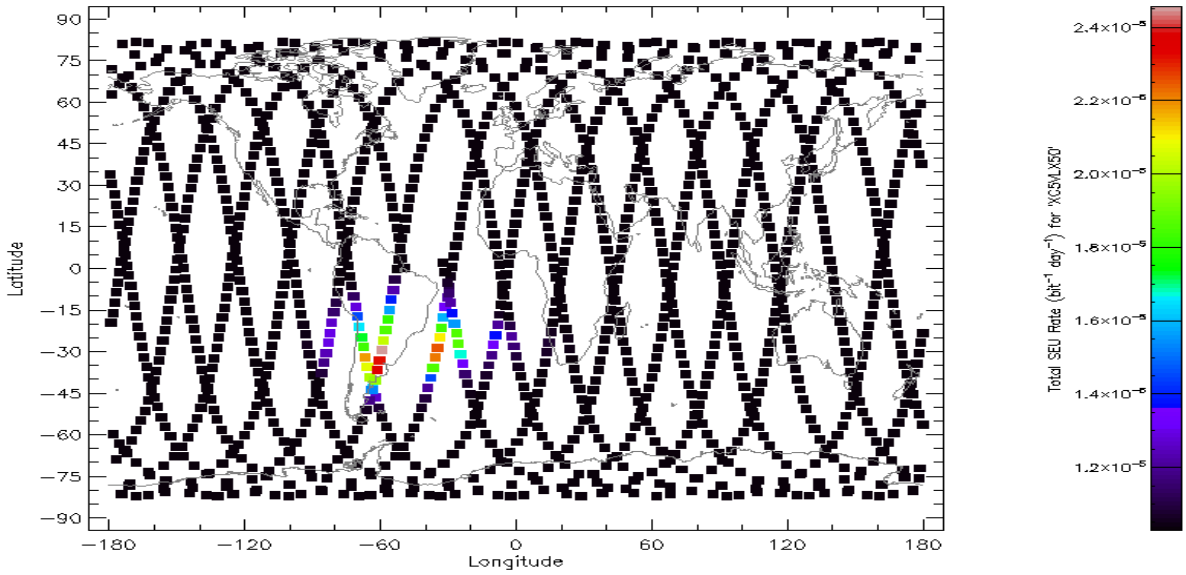


Figure 6a – Average SEU Rate during First Day of Flight

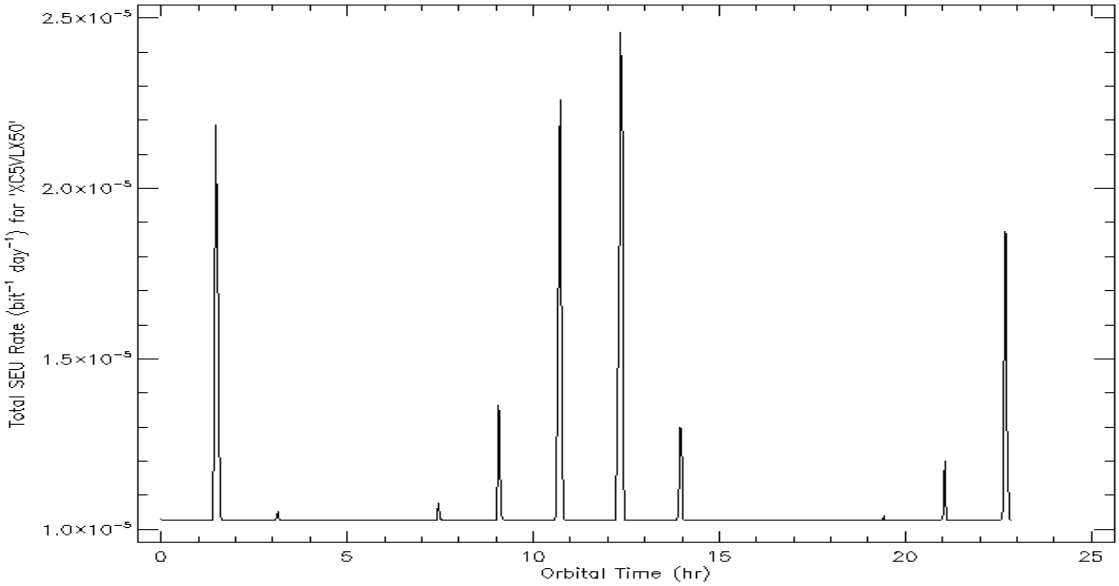


Figure 6b – SEU Peaks at SAA during First Day of Flight

3. PARTIAL RECONFIGURATION

Some manufacturers of SRAM-Based FPGAs introduced the technology of reconfiguring a running design without affecting it, such as the Altera-Stratix and Xilinx-Virtex FPGAs [5, 15]. This technology is crucial in two aspects. First, it provides the flexibility to change the design initially loaded in the FPGA giving more space for extra design functionalities even if there was originally no room to accommodate all of them. Second, it helps to maintain the system performance at a high level through flexible scalability. The system can be expanded or reduced as needed, which means better management of the system power.

The concept of reconfiguration is realized in the Xilinx Virtex family by defining two main partitions: the fixed and reconfigurable. The fixed partition contains the static logic that should not change. The reconfigurable partition is tentative and can be loaded with partial bit streams of different designs. The fixed partition would contain the logic used to load the partial bit-streams representing the replaceable functions. The loading is done by accessing the external flash memory which holds the hardware design in the form of binary bit-streams. In case the FPGA is loaded from an external controller then only the reconfigurable partition will be defined. Figure 7 illustrates the concept of reconfiguration using the Internal Configuration Access Port (ICAP). It is used with an internal controller/microprocessor such as the MicroBlaze soft IP core. Another option is to reconfigure the FPGA externally through an external controller/microprocessor via the SelectMap interface.

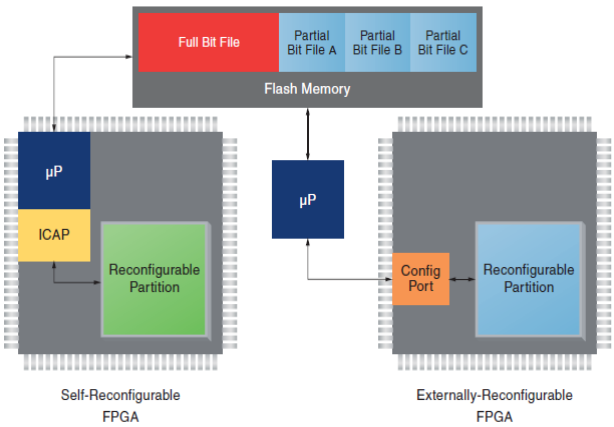


Figure 7 – Partial Reconfiguration Concept- Courtesy of Xilinx

The internal processor can be hard-IP core. The hard-IP cores are implemented in the FPGA hardware. When the FPGA is programmed the hard-IP cores will not be overwritten by the soft-IP cores in the bit-stream. The benefit of using hard-IP processors, such as the PowerPc processor, is to reduce the possibility of being affected by the upsets to the internal FPGA SRAM. If a soft-IP core is used then its routing information and configuration bit-stream is subject to upsets from the space environment radiation. To build a robust system the scrubbing rate should be multiples of the upset rate to reduce the possibility of failures.

The ICAP is connected to the processor bus as a normal peripheral. It receives read-back and writing commands from the processor based on the configuration sequence of the FPGA [11]. The ICAP can be replaced with the SelectMap interface which facilitates reading the FPGA bit-stream from an external port. An external radiation hardened controller can be used to manage the reconfiguration of the FPGA via the SelectMap port.

The configuration bit-stream is stored inside the FPGA in the form of frames in the FPGA SRAM. The frames contain CRC codes for error checking. Xilinx provided primitives for handling the frame error checking and read-back/writing of the frame bits. The read-back of frames bits is useful in scrubbing the frames and correcting the bit upsets. The controller/microprocessor which is installed either internally or externally is responsible for read-back, error checking, error correction and write-back the correct frames.

Reconfigurable FPGA can be designed in such a way that it contains multiple functions which can be swapped in and out according to the application requirements. This swapping mechanism needs all of the different functions to be available in an external flash memory that is accessible by the configuration controller as shown in Figure 7. Whenever a reconfiguration is needed, the configuration controller would load its partial bit-file, such as the partial bit-file A in Figure 7. Bit-files are loaded to specific partitions called the reconfiguration-partition which is planned on the FPGA floor using the PlanAhead tool from Xilinx. The loaded design can be overwritten by other partial bit-files whenever the FPGA needs to change its function. The fixed part of the design logic that should not be overwritten , is the configuration controller itself.

4. AVIONICS SYSTEM DESIGN

The system is designed from soft-IP cores available from Xilinx using the ISE system edition package. PlanAhead is used for the implementation of the partial reconfiguration design. EDK is used to develop the embedded system which is constructed from the basic set called the CPUset shown in Figure 8 [16]. The triplicated system is shown in Figure 9 [16]. The configuration controller and error manager is responsible for loading other design modules. The controller itself is protected from SEUs through triple execution of functions and watchdog timer for resetting in case of hang up.

In the triplicated mode, whenever a processor receives an external input, that leads to changing its current state or entitles it to produce a new output vector, it checks the validity of its calculations with other processors before proceeding. The system is designed to use the soft-IP core

Microblaze processor in implementing the Error Management Module (EMM) module and the CPUsets. The Microblaze has lower power consumption than hardwired processors such as the PowerPC in the Xilinx Virtex5-FX family. In this design we want to prove that soft-IP core processors are safe to use as system level controllers.

The EMM handles system level tasks as it monitors the operation of other CPUsets and receives status information from them. It can reset any particular CPUset and force other CPUsets to resynchronize the processor level data with it.

The system monitors two types of errors:

1. Errors in the FPGA configuration bit-stream.
2. Errors in the system functionality and stored data.

The FPGA can be viewed as an SRAM that is subjected to bit-flips due to SEUs. The contents of the SRAM is not program-code or data, it contains configuration bit-stream of the routing information, Look-Up-Tables (LUT), and BRAM contents.

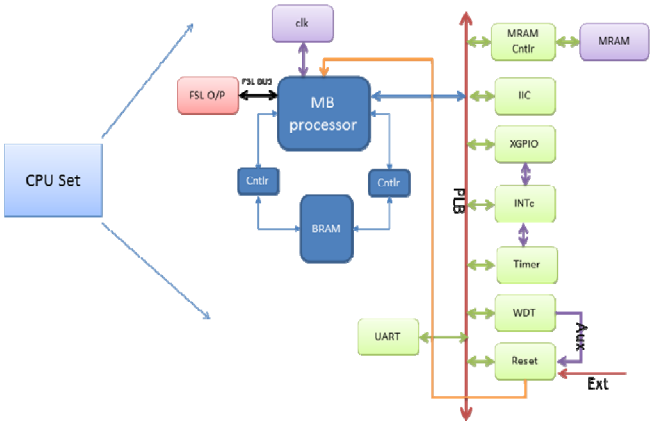


Figure 8 – Single CPUset

The system mitigates itself against accumulated errors in the FPGA SRAM via scrubbing. The scrubbing rate is multiples of the environment SEU rate. In high-energy regions the system is reconfigured through the EMM shown in Figure 10 [16]. The EMM send a heart-beat signal to an external Watch-Dog Timer (WDT) that would reset the system in case of not receiving the EMM heart-beat. Each processor in the CPUset has its own WDT that receives a heart-beat from the relevant processor and would reset the CPUset if the signal is not received. The system makes use of Magneto-resistive Random Access Memory (MRAM). The MRAM is used as the memory for the CPUsets which acts as the program-code and data memory at the same time thus releases the need for flash and SRAM combination.

The MRAM is interfaced to each CPUset through the Xilinx EMC soft-IP core (External Memory Controller). Data is checked among the processors before writing to the MRAM.

The output is applied through voted Fast Simplex Link (FSL) interface, XGPIO soft-IP core and IIC. An interrupt controller is used to handle timer-based interrupts and emergency input signals. The Timer is necessary for running the Xilinx Kernel which is a light real time operating system as well as providing system level time services.

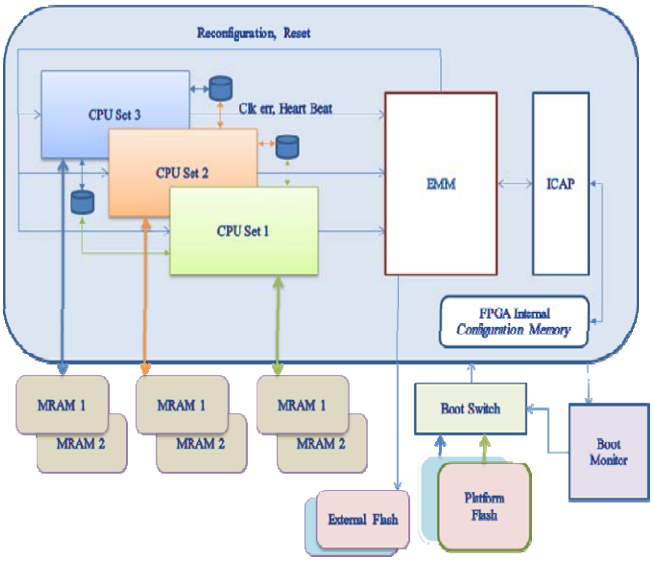


Figure 9 – Triplicated System Loaded at The SAA

The software which runs on each processor adopts a state-machine design. It runs in two modes depending on the system configuration. In case of non-triplicated mode, the software on the EMM is responsible for scrubbing and monitoring the CPUset execution. If the CPUset stops working then the EMM would reset it as well as its own WDT. In case of triplicated-mode, the EMM runs the scrubbing and monitoring routines besides handling the task of checking processors synchronization via receiving time-stamped synchronization messages. If synchronization is lost the EMM restores synchronization among processors through comparing their time-stamped synchronization packets and forcing the system to recover from a previous state. The system keeps storing a Recovery-Block (RB) for its states, flags, registers, and state-machine states and variables for restoring operation whenever it is corrupted.

The software which runs on the CPUset in the non-triplicated mode performs triple execution of critical code sections such as generating output vectors and/or changing execution state. It then checks the consensus of the generated data before actually applying it through voting. triple storage of critical variables is performed in the external MRAM. In the triplicated mode, the software running on the CPUset performs an extra check through exchanging data through the mailbox device driver with other processor modules to make sure that they generated the same values. It also handles the generation of time-stamped synchronization packets and send them to the EMM for checking synchronization among processors. The EMM code is small enough to run from its BRAM.

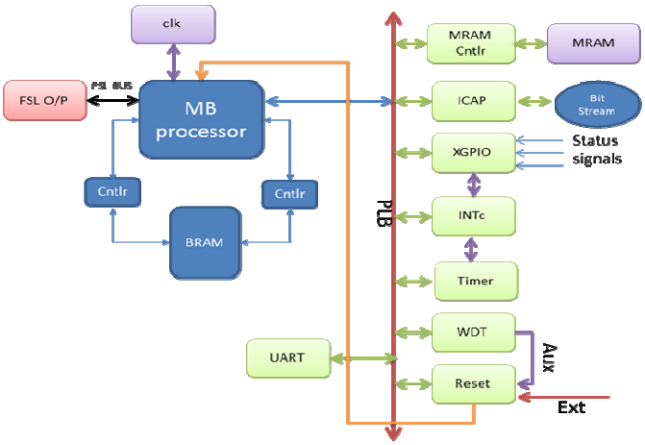


Figure 10 – EMM Block Diagram

The scrubbing algorithm executed by the EMM for mitigating the SEUs is shown in Figure 11 [16]. The EMM corrects the frames if they contain single bit errors. If more than one bit error exists in the read frame then the EMM retrieves the frame from the external MRAM. CRC calculation is done through Xilinx primitive for frame error calculation.

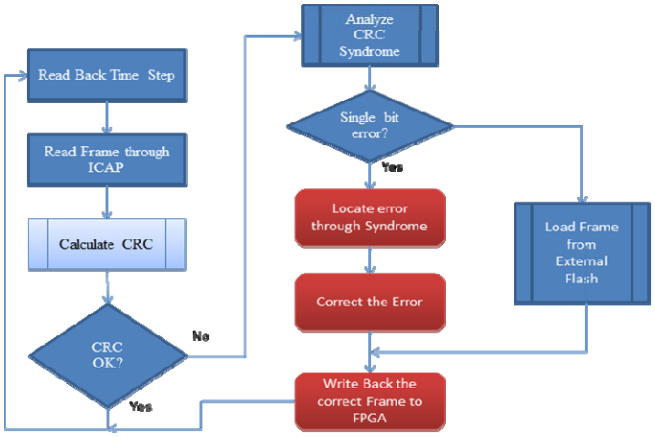


Figure 11 – EMM Scrubbing Algorithm

5. Simulation Results

The average power consumption per single CPUset is around 0.3 Watt. In Figures 12 and 13, the system average power consumption is estimated during a single orbit simulation. It is assumed that the value of the SEU would be fixed during the day for all other orbits. It should be noted that this assumption is only for simplicity. A single orbit does not represent a complete simulation profile for the system operation. This is due to the fact that the average SEU rates slightly change among orbits due to the change of the orbital position of the satellite. Therefore a more complex simulation should be run over many random orbits to achieve a more accurate power consumption average.

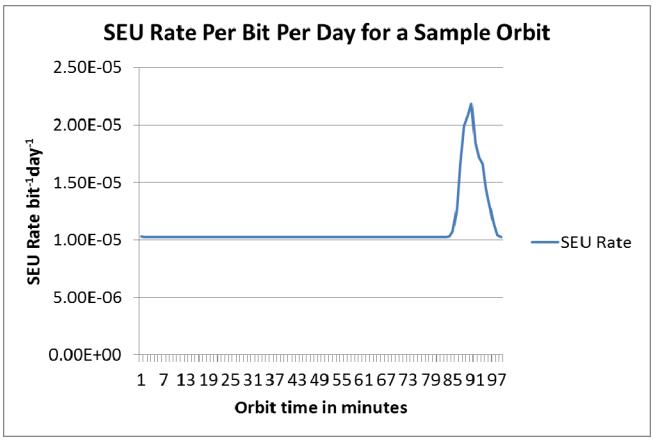


Figure 12 – SEU Per Day Per Orbit

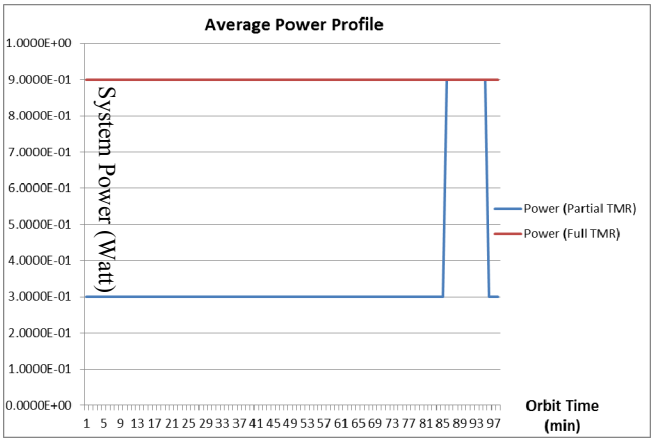


Figure 13 – Average Power Profile in One Orbit

Figure 13 shows the average estimated power consumption profile in one orbit if reconfiguration is used. If the system applies full TMR during all the orbit time, which is 1.63 hours (97.8 minutes), then the average power consumption would be 1.467 Watt-hour. If partial TMR is applied during the orbital position corresponding to higher SEU rates as shown in figure 12, which corresponds to higher fluxes, then the change in the average power consumption would be:

$$P_{Av\text{-}Partial\ TMR} = (11\ min/orbit \times 0.9\ Watt) + (86.8\ min/orbit \times 0.3\ W) = 35.94\ Watt.min/orbit \quad (8)$$

$$\Delta P_{Av} = (P_{Av\text{-}Partial\ TMR} - P_{Av\text{-}Full\ TMR}) / P_{Av\text{-}Full\ TMR} \quad (9)$$

$$\Delta P_{Av} = -59.17\% \quad (10)$$

6. Conclusion

Design of satellites with accepted reliability while maintaining a low-power budget is a target for developers. Modern FPGAs provide the chance to adapt to orbital changes by reconfiguring themselves to withstand external threats [17]-[20].

They can be used to provide functions-on-demand without the need to build the systems using large power consumption systems.

The paper presents a design concept that can be used in developing avionics systems by making use of an important feature of the currently available FPGAs. This feature is the partial reconfiguration technology introduced by Xilinx. In case a system is developed using SRAM-based FPGAs than about 60% of the total power would be saved if the system is reconfigured to use TMR only at orbital locations with high threats. The FPGA configuration data are protected through scrubbing via read-back and calculation of the frames CRC. Increasing the scrubbing rate improves the system reliability.

REFERENCES

[1] Rutter, I.; Vladimirova, T.; , "A CCSDS Software System for a Single-Chip On-Board Computer of a Small Satellite, " *AIAA Small Satellites, AIAA ,* USA, 2001.

[2] Kuwahara, T.; FPGA-based reconfigurable on-board computing systems for space applications, Dr.-Ing. Thesis, Faculty of Aerospace Engineering and Geodesy of the Universitat Stuttgart, Germany, 2010.

[3] Caffrey, M.; Morgan, K.; Roussel-Dupre, D.; Robinson, S.; Nelson, A.; Salazar, A.; Wirthlin, M.; Howes, W.; Richins, D.; , "On-Orbit Flight Results from the Reconfigurable Cibola Flight Experiment Satellite (CFESat)," *Field Programmable Custom Computing Machines, 2009. FCCM '09. 17th IEEE Symposium on* , vol., no., pp.3-10, 5-7 April 2009.

[4] Actel website: www.actel.com

[5] Xilinx website: www.xilinx.com

[6] SPENVIS website: www.spenvis.oma.ba

[7] ECSS-E-ST-10-04C, Space engineering – Space environment (November 2008).

[8] Kyutech website: http://kitsat.ele.kyutech.ac.jp/index_e_new.html

[9] Xilinx, Device Reliability Report, Second Quarter 2012.

[10] 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.

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

[12] Carl Carmichael, Michael caffrey, Anthony Salazar, "Correcting Single-Event Upsets through Virtex Partial Configuration," XILINX, XAPP216, (v1.0), 2000.

[13] Allen, G.; Edmonds, L.D.; Swift, G.; Carmichael, C.; Chen Wei Tseng; Heldt, K.; Anderson, S.A.; Coe, M.; , "Single Event Test Methodologies and System Error Rate Analysis for Triple Modular Redundant Field Programmable Gate Arrays," *Nuclear Science, IEEE Transactions on* , vol.58, no.3, pp.1040-1046, June 2011.

[14] L. Edmonds, Analysis of SEU Rates in TMR Devices, Internal Document, JPL Publication 09-6, February 2009

[15] Altera website: www.altera.com

[16] Ibrahim, M., Asami, K., and Cho, M., The Use of Reconfigurable FPGAs in Developing Reliable Satellite On-Board Computers, Online Proc. of AIAA SPACE 2012 Conference & Exposition, Pasadena, California, USA, September 11-13, 2012.

[17] Smith, G.L.; de la Torre, L.; , "Techniques to enable FPGA based reconfigurable fault tolerant space computing," *Aerospace Conference, 2006 IEEE* , vol., no., pp.11 pp., 0-0 0

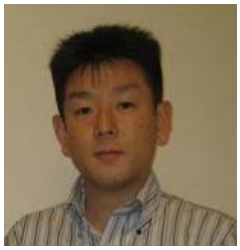
[18] Zhuo Ruan; Yuzhang Han; Hongbo Cai; Shengzhen Jin; Jianguo Han; , "A Dynamically Partial-reconfigurable FPGA-based Architecture for Data Processing on Space Solar Telescope," *Industrial Embedded Systems, 2007. SIES '07. International Symposium on* , vol., no., pp.194-199, 4-6 July 2007.

[19] Yousuf, S.; Jacobs, A.; Gordon-Ross, A.; , "Partially reconfigurable system-on-chips for adaptive fault tolerance," *Field-Programmable Technology (FPT), 2011 International Conference on* , vol., no., pp.1-8, 12-14 Dec. 2011.

[20] Quinn, H.; Graham, P.; Morgan, K.; Baker, Z.; Caffrey, M.; Smith, D.; Bell, R.; , "On-Orbit Results for the Xilinx Virtex-4 FPGA," *Radiation Effects Data Workshop (REDW), 2012 IEEE* , vol., no., pp.1-8, 16-20 July 2012.

BIOGRAPHIES

Mohamed Ibrahim *received a B.Sc. in Engineering from Cairo University, Egypt in 2000. He worked in the development of small satellites for earth observation for 10 years with the National Authority for Remote Sensing and Space Sciences/Egyptian Space Program, Egypt. He Finished his M.Sc. in computer Engineering from Al-Azhar University, Egypt in 2011. Also, he holds an MBA degree from Arab Academy for Science Technology and Maritime Transport, in 2012. Currently he is pursuing his Ph.D. at Kyushu Institute of Technology, KitaKyushu, Japan. He is the project leader of the Horyu-V Space Environment Exploration Satellite.*



Kenichi Asami *received his B.Sc. from the Faculty of Computer Science and Systems Engineering, Artificial Intelligence Dept., Kyushu Institute of Technology (Kyutech), KitaKyushu, Japan in 1992. He obtained his Doctor of Information Engineering degree in intelligent informatics from Kyutech in 1997. Currently, he is associate professor at the faculty of engineering, Applied Science for Integrated System Engineering Dept., Kyutech, Japan. His research interests include embedded systems, evolutionary robotics, and reconfigurable computing.*

Mengu Cho *Received his B.Sc. and M.Sc. from the department of Aeronautics, University of Tokyo, Tokyo, Japan in 1985 and 1987, respectively. He received his Ph.D. in plasma physics from the department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, USA. Currently, he works as a professor in the department of Applied Science for Integrated System Engineering at Kyushu Institute of Technology, KitaKyushu, Japan. He is the director of the Laboratory of Spacecraft Environment Interaction Engineering at Kyutech. His research interests are in space systems engineering, spacecraft environmental interaction, spacecraft charging, computer simulation and developing nano-satellites such as the Horyu-series.*