Dynamic partial FPGA reconfiguration in space applications

Rafal Graczyk*[a,b], Marcin Stolarski[a,b], Marie-Catherine Palau[b], Piotr Orleanski[a]
[a]Space Research Centre Polish Academy of Sciences, Bartycka 18A, 00-716 Warsaw, Poland
[b]Astri Polska sp.z o.o., Bartycka 18A, 00-716 Warsaw, Poland

ABSTRACT

Design and implementation of hardware mock-up of high performance system for general avionics testing in reconfigurable FPGAs. Strong emphasis is put on exploiting dynamic partial reconfiguration capability as a method for functionality multiplexing and fault mitigation. Additionally, dynamic reconfiguration can be used for fault injection which makes Single Event Upset in configuration memory simulation possible. LEON3 processors are used to create an avionic systems test-bed, for testing the mock-ups of real system flight software and testing dynamic full and partial reconfiguration. Experiments with different means of reconfiguration are performed to measure reconfiguration times and stability of software. Several solutions for whole system reconfiguration controller have been implemented and tested.

Keywords: reliability, fault, mitigation, FPGA, LEON3, space, avionics

1. INTRODUCTION

Progress in field of space technologies and resultant more efficient expansion in space is, to great extent, related to successful adaptation of civil, earth-grade solutions to more demanding requirements of aerospace industry, especially in terms of environmental hardness. This conclusion is derived from following rationales. First, aerospace industry is slowly recognizing widening gap in system performance when compared to state-of-the-art commercial solutions, mainly due to strong emphasis on device flight heritage and it proven and continuously tested reliability. Modern devices, having more logic or processing capacity are also more susceptible to radiation and often dissipate more heat in smaller volume, therefore are subject of imminent failures. The data acquisition and processing capabilities of proven systems (as for 2012), considered by aerospace industry as safe and with sufficient heritage are similar to those of commercial grade from mid-90s. As modern, civilian solutions performance grow in exponential manner – the capability gap between military/aerospace and commercial systems become more and more visible.

Second, there is a tendency to build more sophisticated systems, having more processing power capability or having similar processing power capability but with lower size, weight, electric power requirements. In parallel to increase of space systems complexity three other development vectors become significant: limiting flight software volume in order to keep it testable and manageable in reasonable budgets constraints, early prototyping and fast solution verification approach in order to check it's technical feasibility without risking project success, reduction of project cycle time in order to keep up with competition and customers.

Those tendencies results in more frequent application of advanced, high-performance programmable logic devices – even in systems with very high environmental requirements, also in term of its vulnerability to ionizing radiation. On one hand it is push for moving the functionality from software running on processor into the hardware finite state machines where it is easier to prove it to be safe. On the other hand, aerospace equipment developers tend to test what they design as soon as possible so ease of prototyping, modifying and verification of the design is crucial. Additionally it is beneficial to reuse already tested parts and design new components in a way they can be implemented again in future projects.

*rgraczyk@cbk.waw.pl; www.cbk.waw.pl

Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2012,
edited by Ryszard S. Romaniuk, Proc. of SPIE Vol. 8454, 84540G · © 2012 SPIE
CCC code: 0277-786/12/$18 · doi: 10.1117/12.2000190

As an outcome, high performance, SRAM memory based (partially) reconfigurable FPGA make their way into aerospace industry and newly designed advanced data processing and control systems. Very good examples of such devices are a Xilinx Virtex4 and 5 families. Those FPGA offer vast logic resources, lots of input-output pins, embedded hard IP cores like PowerPC processors or Ethernet controllers and support for high-speed backplane buses (PCIe, Infiniband). One of more interesting feature of Xilinx Virtex devices is their capability of being dynamically (in run time), partially (just a part of design) reconfigurable. Dynamic reconfiguration can be managed from inside of FPGA fabric (using internal port) or from outside of FPGA (using standard configuration memory access method or JTAG).

As a drawback, SRAM based FPGAs, even in military or aerospace versions are susceptible to faults induced by radiation. As Xilinx devices, from all available SRAM FPGAs, have best radiation immunity versus logic resources or performance, therefore this vendor is considered as provider of described integrated circuits. Total Irradiation Dose (TID) is a factor that needs to be taken into account during design phase but Single Event Effects (SEE) are much more destructive and much harder to mitigate as happen in system operation phase. Circuits inside FPGA that are susceptible to Single Event Upsets (SEU) are mainly configuration memory (largest part of FPGA sensitivity), flip-flops in data path logic, block RAMs, Look-Up Tables and latches. Additionally, bearing in mind statistics that shows that around 80% of all FPGA transistors are used for routing signals and even in most highly utilized designs only up to 20% of all transistors are used, it becomes obvious that most likely to happen in FPGAs are Single Event Functional Interrupts (SEFIs) induced by SEUs in configuration memory [1].

In case of Xilinx devices no destructive event have been reported. All SEUs in configuration memory can be repaired by overwriting the altered memory contents. In case SEU occurs in one of service circuits (ICAP, JTAG or so) a FPGA reset and system reconfiguration is necessary [2].

Therefore, in order to safely operate high performance SRAM based FPGA in space environment an additional protection methods must be implemented that ensure proper operation of at least, mission critical parts of design. Due to vast logic resources of Xilinx FPGA, triple-modular redundancy (TMR) or even Quadruple Modular Redundancy is one of vast solutions. If coupled with additional measures like configuration memory scrubbing and dynamic reconfiguration of faulty units system is getting robust.

2. TEST SYSTEM FOR AVIONICS RESEARCH

To evaluate effectiveness of each fault mitigation mechanism a avionics system mock-up has to be built. Chosen FPGA technology makes it possible to implement several system architecture variants with different degree of system reliability and complexity, and logic resources usage.

In aerospace domain, in recent years, a LEON processor become very sound solution to typical control needs. Its third version, LEON3 is presently considered as a standard for a general purpose processor. LEON 3 comes in two variants, a nominal one (issued under General Public License, GPL) and a radiation hardened variant, with Triple Module Redundancy (TMR) on register or memory cell level. The TMRed variant is considered commercial and available only as a net-list for dedicated FPGA, not as a VHDL source code, unlike GPL version. The drawback of a TMRed variant is that designer is no longer in charge of what blocks processor consists of nor how they are placed in FPGA configurable fabric.

In order to raise the system reliability and to prevent that Single Event Upsets (SEUs) affect the system operation Triple Module Redundancy (TMR) is proposed on processing unit level as a baseline (option 1). Second, and third options are optimization removing memory or communication units from reconfigurable regions of processing unit. The memory or communication units, moved over the voting logic have to be considered as “fault-safe” for the purpose of the experiments till the moment of implementation of system option 4, where every existing functional unit inside and outside FPGA is TMRed.

System architecture options:

1. 3 complete processing units, voting on memory and communication interfaces
2. Memory controller moved over voter logic, AMBA connected to voter
3. Memory and communication controllers moved over voter logic, AMBA connected to voter
4. Global TMR –3 processing units, 3 ext. memories, 3 communication units

TMR scheme will be realized in a way that three independent processor units execute the same software and communicate via the same interface. All the input and output operations in IO and memory interface are voted, and decision of majority is assumed to be valid. Therefore, such system is single point of failure free from the perspective of processing unit (processor and accompanying units). Architecture of system utilizing TMR in described way is shown on Fig. 1.

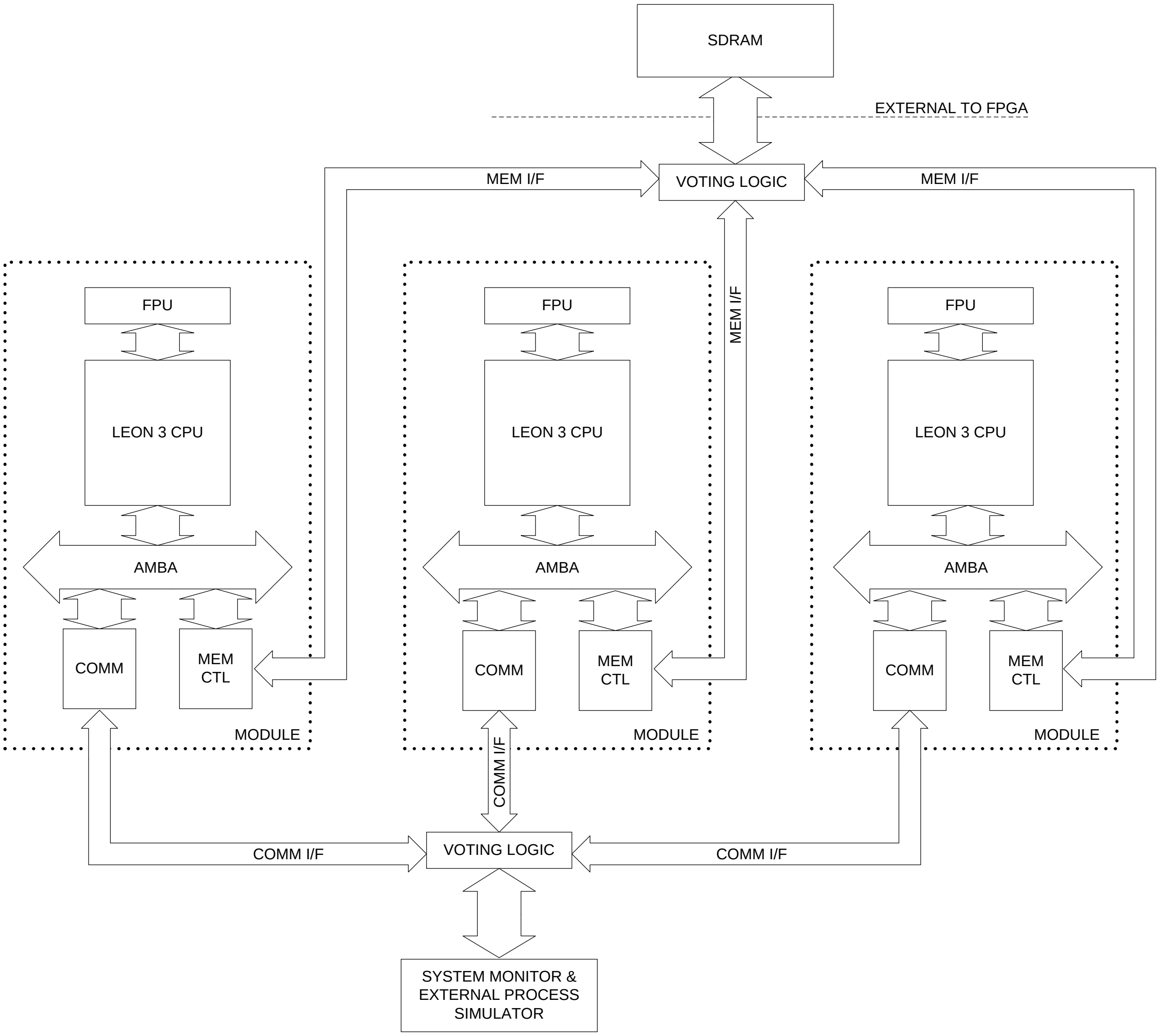

Fig. 1 System architecture - option 1

It is worth noting that each processing unit is considered a reconfigurable module placed in reconfigurable region. It means that the partial bit stream, dealing with FPGA configuration memory related to only this region is available. Therefore it is possible to rewrite configuration memory of this region in case it is suspected to be altered by SEU.

Alternative system architectures are going to be implemented to check their suitability.

In second option (Fig. 2) memory controller is moved out of processing unit. There are three reasons for this approach. First, memory controller can be TMRed itself so can be considered as safe. Second it simplifies the processing unit and lowers usage of resources. Therefore only one (TMRed) unit is necessary, and processing units communicate with memory controller through AMBA bus and voters. Third, SDRAM memory controllers may need to issue some refresh cycles and other housekeeping commands, it may be much more convenient to have only one on board.

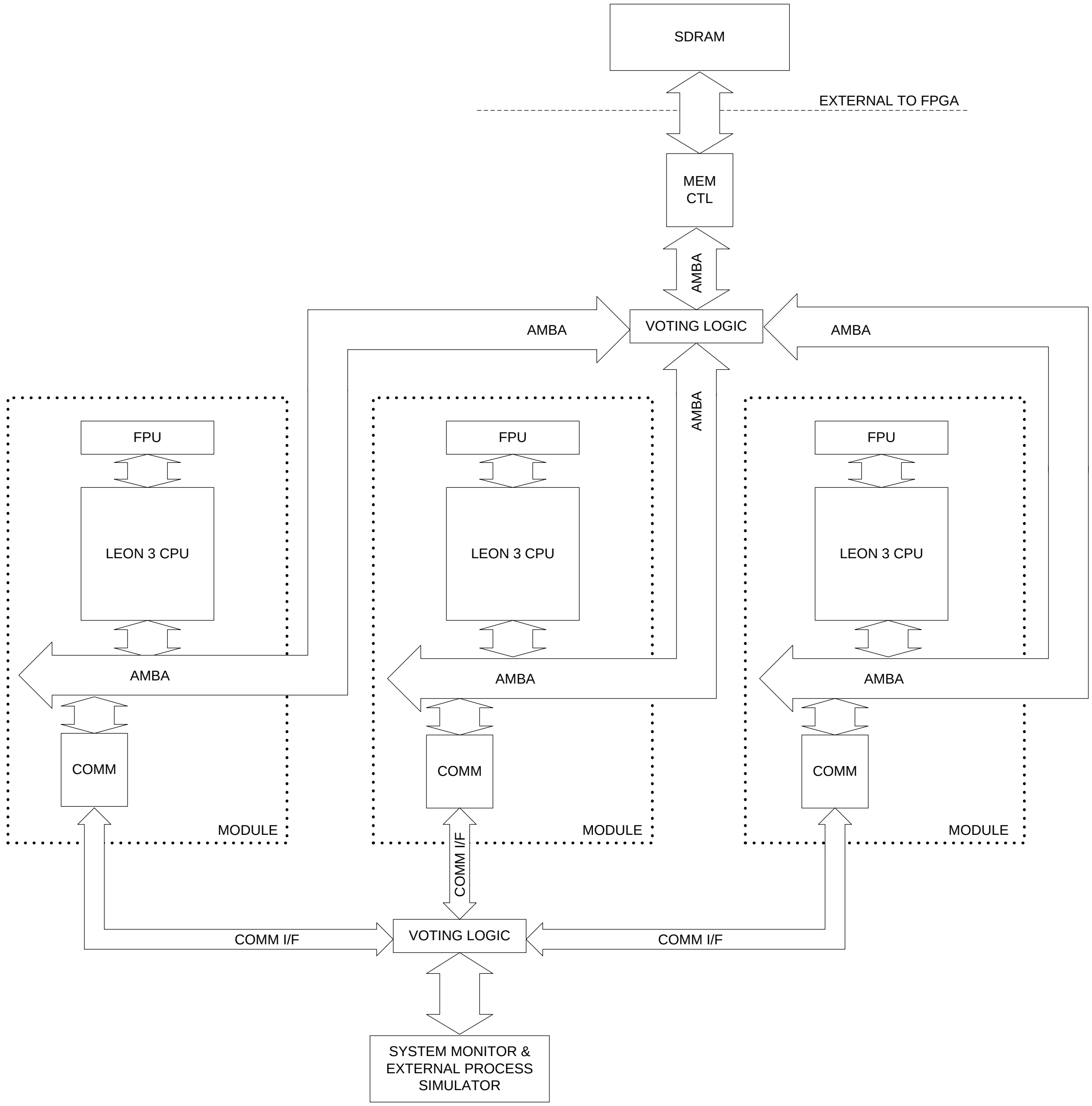

Fig. 2 System architecture - option 2

Another alternative, third option (Fig. 3) is extension of idea presented in option 2 (Fig. 2).

Here processing unit, contains only LEON3 processor with caches and FPU and AMBA bus controller. AMBA bus is the interface going outside reconfigurable module, where it joins AMBAs from other processing elements in voters. Both Memory Controller and IO controller have to be triplicated to make it hardened to SEU occurrence.

Implementation detail: voters and memory and IO controllers have to reside in reconfigurable regions so partial bit streams will be available for them. Reconfiguration controller have to perform scrubbing (at frequency which depends on SEU rate in configuration memory) to make sure that circuits remain as designed.

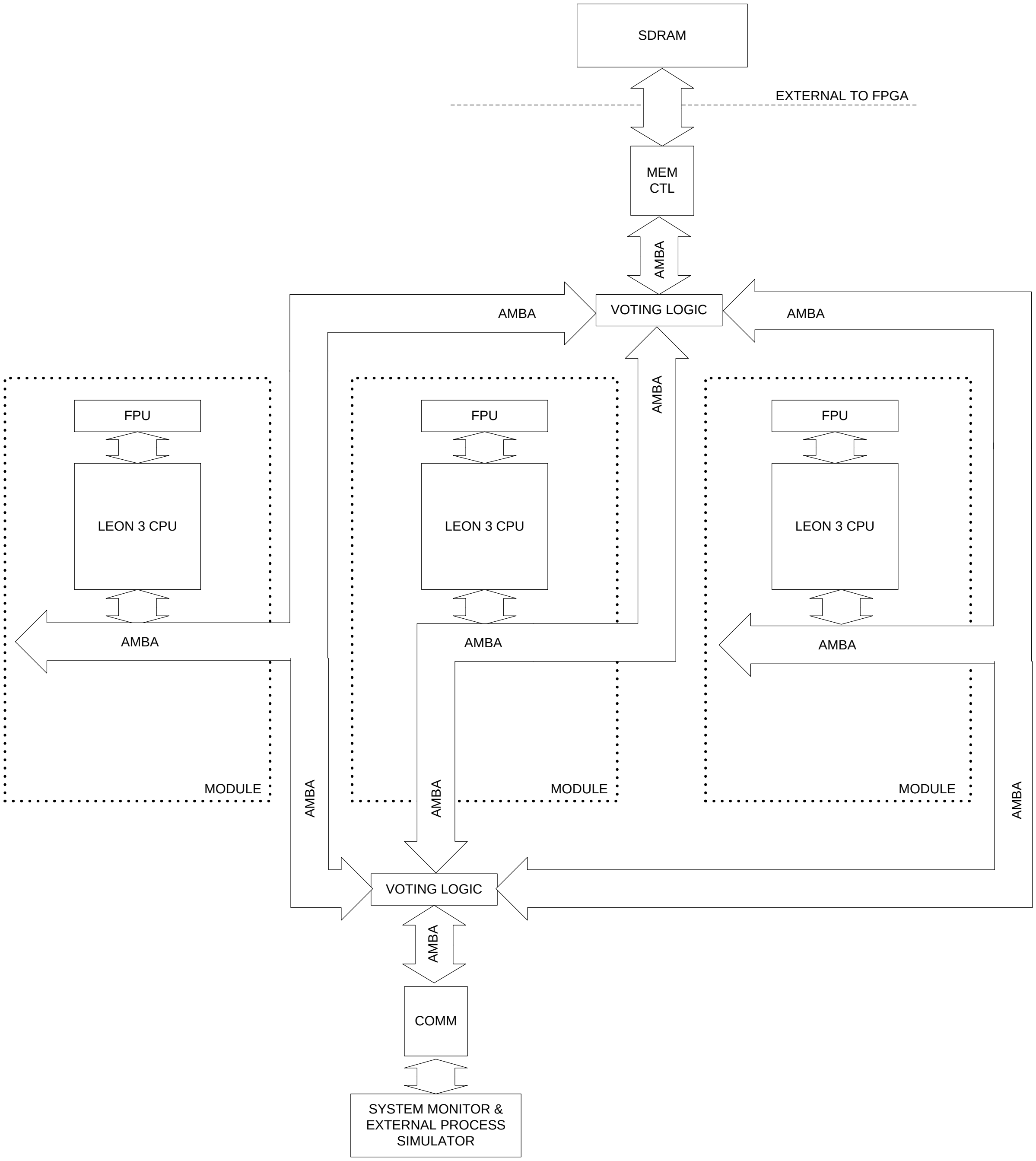

Fig. 3 System architecture - option 3

From the system architecture perspective, the most reliable case is when whole design is globally triplicated (clocks, memories and so on) so ideally would be to construct a system with triplicated external memories. Such case involves system board redesigning therefore, simplified architectures will be tested initially, and the globally triplicated architecture will be tested when the new system board is ready. This is considered to be fourth option.

3. PARTIAL RECONFIGURATION AS A FAULT MITIGATION METHOD

Dynamic reconfiguration controller is the unit responsible for managing the contents of configuration memory. It writes the configuration memory in given reconfigurable region on periodic basis (scrubbing) or when trigged by detected error.

As it has been described in system architecture options, signals on buses managing the information exchange with external components are voted in order to mitigate functional or data errors in each of processing unit. To detect errors, each voter input is fed into the signature generation circuit in order to generate a unique signature for past occurrences of data on the bus. Signature is shorter than lengths of data buses therefore easier to compare. If any of signatures differs from others (so the data is different), even if error is corrected on the voter output, reconfiguration controller is triggered to take action, as error could be caused by faulty configuration. Error flag contains information which processing element module is generating errors, and which, in turn has to be immediately checked against errors, and restarted – to make sure its register contents are the same as in other 2 processing units.

As a first approach Reconfiguration Controller will be located internally in FPGA. For further developments it is necessary to implement such controller in high-reliability device like antifuse FPGA or qualified microprocessor with reliable memory. When implemented in SRAM FPGA such controller has to be TMRed, checked constantly against errors in configuration memory and properly reset on periodic basis. Errors in configuration memory can be overwritten without error notification by scrubbing process, or errors can be detected directly by frame readback. Triplication is also a good design practice when Reconfiguration controller is implemented in antifuse.

To keep system as simple as possible and ease debugging, Reconfiguration Controller will be implemented as Finite State Machine (FSM) rather than microprocessor (rationale: easier TMR, lower resources consumption, similar functionality as implemented on processor). Reconfiguration controller will be connected to first, Top ICAP port. ICAP ports (there are two in Virtex 5) are the gateways to FPGA's configuration memory. Second, Bottom ICAP ports are going to be used to inject bit-flips into configuration memory to simulate SEUs there. As both ICAP port can't be used simultaneously, error injection functionality will be coupled with operation of Reconfiguration Controller.

4. SUMMARY

A system containing three LEON3 processor operating in hot redundancy, with voting circuit as a base for avionics test-bed, implemented in SRAM FPGA has been shown. Majority voting circuits are necessary to combine TMR system outputs and mitigate radiation induced faults. Dynamic reconfiguration controller manages all configuration memory activities including blind scrubbing and partial reconfiguration on demand (in case module in fault is detected) – taking advantage of SRAM technology. Two controller manager implementation option are considered – one internal to FPGA fabric (that has to be TMRed) and one external to FPGA (that has to be implemented in radiation hardened technology).

As LEON3 is considered a standard for aerospace solutions some applications containing flight software mock-up for Ariane 5 and ATV vehicles has been created. Ariane 5 mock-up software is based on small data flows and very tight control loops, while ATV mock-up software is based on higher data-flows and less stringent control loop constraints.

Taking into account that whole Virtex5 FX130T reconfiguration takes around 30ms (via ICAP or SelectMAP, smaller partitions respectively faster in linear manner) globally TMRed design is absolutely fault-safe in Low Earth Orbit, even for commercial devices expecting several upsets a day. Conclusion is that commercial Virtex 5 FX130T is a viable option for creating high performance control and data acquisition and processing systems for space applications, but safety measure has to be taken into account (TMR, QMR, Error Correction Coding, scrubbing, reconfiguration on demand).

Final decisions on reconfiguration controller placement, whether internally or externally to FPGA has to taken. Further analyses are necessary. It seems feasible (experiments have shown) to build internal reconfiguration controller protected enough to ensure safe operation of functional modules. Although, there are still SEU that can happen in internal circuits of FPGA that lead to necessary hard reset – which might have to be performed externally.

REFERENCES

[1] C. Carmichael, E. Fueller, P. Blain, M. Caffrey, “SEU Mitigation Techniques for Virtex FPGAs in Space Applications”, <http://china.xilinx.com/esp/mil_aero/collateral/presentations/SEU_mitigation_technique.pdf>

[2] Xilinx “Radiation Effects and Mitigation Overview” <http://www.xilinx.com/esp/mil_aero/collateral/presentations/radiation_effects.pdf>

[3] Graczyk, R.; Stolarski, M.; Cormery, P. "Exploratory study about the use of new reconfigurable FPGAs in space”, Adaptive Hardware and Systems (AHS)," 2011 NASA/ESA Conference on Digital Object Identifier: 10.1109/AHS.2011.5963940 Publication Year: 2011 , Page(s): 220 - 226

[4] Melanie Berg, C. Poivey, D. Petrick, D. Espinosa, Austin Lesea,K. A. LaBel, M. Friendlich, H. Kim, and Anthony Phan, "Effectiveness of Internal Versus External SEU Scrubbing Mitigation Strategies in a Xilinx FPGA: Design, Test, and Analysis," IEEE TRANSACTIONS ON NUCLEAR SCIENCE, VOL. 55, NO. 4, AUGUST 2008

[5] Portela-García, M. López-Ongil, C. Valderas, M.G. Entrena, L. "Fault Injection in Modern Microprocessors Using On-Chip Debugging Infrastructures, " IEEE Transactions on Dependable and Secure Computing, Volume 8 issue 2, ISSN: 1545-5971.

[6] Pedro Yuste, Juan Carlos Ruiz, Lenin Lemus and Pedro Gil, "Non-intrusive Software-Implemented Fault Injection in Embedded Systems, " Lecture Notes in Computer Science, 2003, Volume 2847/2003, 23-38

[7] Ar lat J. , et al . , "Comparison of Physical and Software-Implemented Fault Injection Techniques, " IEEE Trans. on Computers, vol. 52, no.9, pp. 1115-1133, 2003.

[8] Benso, A., Prinetto, P.: [Fault Injection Techniques and Tools for Embedded Systems Reliability Evaluation], Kluwer Academic Publishers. (2003).

[9] Richard Stallman, Roland Pesch, Stan Shebs, et al., “Debugging with GDB”, 2010-10-16.

[10] Areoflex Gaiser AB, “TSIM2 Simulator User`s Manual – ERC32/LEON2/LEON3”, March 2011.