CSP: A Multifaceted Hybrid Architecture for Space Computing

Dylan Rudolph, Christopher Wilson, Jacob Stewart, Patrick Gauvin, Alan George, Herman Lam
NSF Center for High-Performance Reconfigurable Computing — University of Florida
327 Larsen Hall, 968 Center Drive, Gainesville, FL, 32611; 352-392-9225
george@chrec.org

Gary Crum
NASA Goddard Space Flight Center
8800 Greenbelt Rd, Greenbelt, MD, 20771; 301-286-7153
gary.a.crum@nasa.gov

Mike Wirthlin, Alex Wilson, Aaron Stoddard
NSF Center for High-Performance Reconfigurable Computing — Brigham Young University
459 Clyde Building, Brigham Young University, Provo, UT, 84602; 801-422-7601
wirthlin@ee.byu.edu

ABSTRACT

Research on the CHREC Space Processor (CSP) takes a multifaceted hybrid approach to embedded space computing. Working closely with the NASA Goddard SpaceCube team, researchers at the National Science Foundation (NSF) Center for High-Performance Reconfigurable Computing (CHREC) at the University of Florida and Brigham Young University are developing hybrid space computers that feature an innovative combination of three technologies: commercial-off-the-shelf (COTS) devices, radiation-hardened (RadHard) devices, and fault-tolerant computing. Modern COTS processors provide the utmost in performance and energy-efficiency but are susceptible to ionizing radiation in space, whereas RadHard processors are virtually immune to this radiation but are more expensive, larger, less energy-efficient, and generations behind in speed and functionality. By featuring COTS devices to perform the critical data processing, supported by simpler RadHard devices that monitor and manage the COTS devices, and augmented with novel uses of fault-tolerant hardware, software, information, and networking within and between COTS devices, the resulting system can maximize performance and reliability while minimizing energy consumption and cost. NASA Goddard has adopted the CSP concept and technology with plans underway to feature flight-ready CSP boards on two upcoming space missions.

1 INTRODUCTION

On-board computing is among the most critical needs of current- and future-generation spacecraft. The requirements for processing devices in spacecraft are being driven by two major application areas: (1) sensor processing (e.g. signal, image, and video compression and processing); and (2) autonomous processing and control (e.g., autonomous rendezvous, docking, and formation flying). Due in large part to sensor, control, and mission-scope advancements in spacecraft systems, future-generation space computers will be tasked with a much more considerable processing load than those in present-generation spacecraft, but they will continue to be constrained by the complex requirements to which all space systems must adhere, including both ionizing-radiation effects and stringent restrictions on size, weight, and power.

Current-generation space processors tend to permit considerable, and perhaps unnecessary, compromises in

certain requirements to satisfy others. A common design strategy for many space computers is to exclusively use RadHard (i.e., radiation-hardened) components for all subsystems. While this approach will almost certainly result in a reliable system, the compromises in performance, size, weight, power, and cost are far from insubstantial. Another design strategy, for missions in which requirements are not as stringent, is to entirely use COTS (i.e., commercial off-the-shelf) components. While an all-COTS solution might perform admirably in all other requirements, the reliability of the system is likely to be poor in a space environment.

Among the research goals of the CHREC Space Processor (CSP) project is the determination of a method by which we can intelligently combine RadHard and COTS components to produce a hybrid computing system which has both more computational performance (among other benefits) than an equivalent

all-RadHard system, and higher reliability than an equivalent all-COTS system. Additionally, the CSP project is concerned with researching how best to apply the principles of fault-tolerant computing to augment the inherent reliability of a mixed RadHard and COTS computing solution for varying mission needs.

The realization of this research, the first production version of the CHREC Space Processor (CSPv1), is presented here. CSPv1 is a small computer that features an innovative combination of three technologies: COTS devices; RadHard devices; and novel fault-tolerant computing techniques. By featuring COTS devices to perform the critical data processing, supported by simpler RadHard devices that monitor and manage the COTS devices, and augmented with novel fault-tolerant computing (in the form of hardware, software, information, networking, and time redundancy) within and between COTS devices, the resulting system can maximize performance and reliability while minimizing energy consumption and cost. CSPv1 is also designed to be scalable in application scope—by the use of multiple CSP cards in a system, most any computing performance requirement can be accommodated, from CubeSats to large spacecraft. We will detail a number of the strategies and design decisions involved in the creation of CSPv1 in two categories, those concerning *hardware architecture* and those concerning *software architecture*.

2 BACKGROUND

The conventional wisdom in designing a space-processing system is that, in order to ensure system reliability, it is necessary for all subsystems to be RadHard. Some representative designs of this sort are detailed in [1] and [2]. However, this approach is a costly strategy in many respects. In terms of performance per Watt, the processing devices in these systems are typically several orders of magnitude worse than modern commercial devices [3], and they also have a large size and mass footprint. Additionally, they are logistically costly, with lead times of several months, and three orders of magnitude more expensive than equivalent COTS devices.

A comparison, based on the data and methods of [3], between a modern commercial processor (the Xilinx Zynq-7020 selected for CSPv1), and several traditional RadHard processors is given in Table 1. It is apparent that the Zynq device considerably outperforms each of these RadHard devices in all respects, including performance per Watt — a very important metric for space applications. Additionally, it is worth noting that the Zynq device is much less expensive (on the order of $100) than RadHard processors. However, use of RadHard devices is pervasive, as the increased reliability resulting from a lack of ionizing-radiation effects is generally perceived to be worth the costs for certain applications.

Table 1: Normalized Performance of Zynq-7020 versus Common RadHard Processors [3]

| Performance Parameter (Ratio of Zynq to Baseline) | Baseline Devices | | |
	Aeroflex Glasier GR712RC	BAE Systems RAD750	Honeywell HXRHPPC
Computation Performance (32-bit integer)	591.8	178.0	591.8
Computation Performance (32-bit floating-point)	1549.6	291.3	484.2
Performance per Watt (32-bit integer)	235.5	236.1	1193.2
Performance per Watt (32-bit floating-point)	549.6	344.4	870.2
Input / Output Bandwidth	73.9	56.3	560.4
External Memory Bandwidth	106.6	40.1	No EMB

Broadly categorized, there are two different types of ionizing-radiation effects which are of concern for integrated circuits. The first is the consequences which are as a result of long-term exposure to a large number of highly-energetic particles. The degree of this exposure is quantified by the Total Ionizing Dose (TID). The second type of effect which concerns integrated circuits is the result of highly energetic particles hitting critical areas of the die, causing temporary changes in the state of the device (though these temporary changes have the potential to cause permanent damage if unmanaged). These are referred to as single-event effects, of which there are several subcategories.

A number of COTS devices are capable of withstanding a fairly high TID, but COTS devices typically do not have particularly strong single-event immunity [4]. Devices similar in complexity to the Zynq, which is featured on CSPv1, have been tested and are known to have TID ratings that permit them to easily operate in low-earth orbit for a long period of time (on the order of several years) [5][6][7]. So, if the single-event effects can be properly managed, these types of devices have the potential to be viable for space systems.

With regard to both fixed-logic and reconfigurable-logic devices, there is a great body of research concerning how to correct and manage single-event effects. Fixed-logic devices can track and correct

memory errors through parity information and temporal redundancy [8][9], and reconfigurable-logic devices (subject to a different set of possible errors) can be corrected and managed by means of periodic checking, and possible scrubbing, of their configuration, along with constructing hardware-redundant features in their design [10][11][12]. Additionally, COTS devices can be managed and monitored by means of external hardware and supervisory circuits (as is the case for CSPv1).

3 HARDWARE ARCHITECTURE

The hardware of CSPv1 is designed to fit inside a 1U CubeSat form factor (10 cm by 10 cm), and is mated with the surrounding system perpendicularly though a backplane connector. The primary novelty of the hardware is the possibility to selectively populate any combination of RadHard and COTS components (for the monitoring systems) on the same Printed Circuit Board (PCB), permitting a spectrum of possible hybrid boards for different mission requirements. An image of the top side of this board (for the all-COTS variant) is shown in Figure 1; the regions which are unpopulated are reserved for RadHard components. Additionally, CSPv1 makes use of a supervisor circuit to monitor the processing device in the event of an upset for which the processor is unable to internally accommodate.
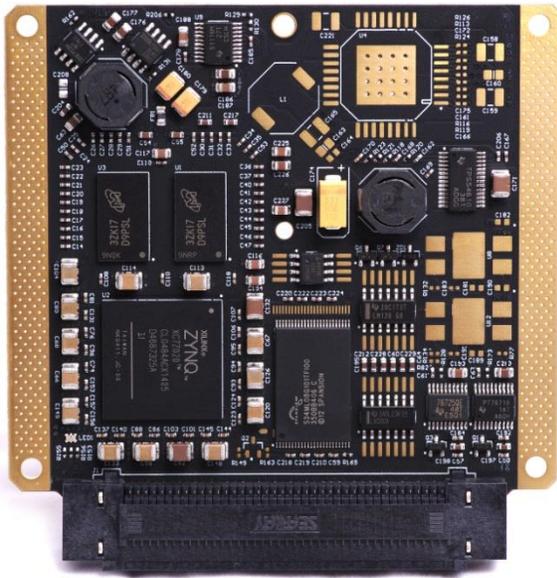

Figure 1: Top of all-COTS CSPv1 Board

3.1 Device Selection

There are some types of COTS devices that might be generally inadvisable to use in space systems in which reliability is prioritized. Power-supply circuitry generally falls under this category—even a very short single-event effect which causes the output of the supply to be out of specification can result in system failure. The way that CSP discriminates between devices which should be RadHard and those which need not be RadHard can be summarized as:

If ionizing radiation effects of either type are a concern, and the device cannot be easily monitored and managed, then it should be made RadHard.

The result of applying the above rule is that power supplies, supervisors, and sequencing circuitry are made RadHard, but the main processing device and its memory are not, as they can be monitored and corrected. Fortunately, the application of this rule results in a RadHard set of devices which are fairly simple in construction, and so result in a lessened degree of compromise.

Table 2: Unmonitored Devices

Device Type	COTS Vendor	RadHard Vendor
NAND Flash Memory	Spansion (1 GB)	3D-Plus (4 GB)
Switching Regulators	Texas Instruments	Peregrine Semi
Linear Regulators	Texas Instruments	Aeroflex
Supervisory	Intersil	Intersil
Power Sequencing	Texas Instruments	Texas Instruments
Reset Management	Texas Instruments	Texas Instruments

The devices on CSPv1 which cannot be easily monitored are able to be populated with either RadHard or COTS solutions. The vendors of these devices are shown in Table 2. There are also some devices, the *monitored* devices, which are always populated, and these are shown in Table 3.

Table 3: Monitored Devices

Device Type	Vendor	Description
Processing Device	Xilinx	Zynq 7020, 485-ball BGA
Volatile Memory	Micron	512 MB (Total) DDR3

An ancillary benefit of the population strategy of CSP is that low-cost development boards with all-COTS population can be used for software development, and the software need not change when moved to a flight (hybrid RadHard and COTS) board. These lower-cost development boards can be manufactured for about $1k, while flight boards can be made for about $10k.

3.2 Processor Architecture

The Xilinx Zynq-7020 device used by CSPv1 contains a dual-core ARM A9 processor, and a 28nm Artix-7 FPGA fabric. Attached to the ARM side of this device, the CSP has 512 MB of DDR3 memory, with PCB support for 1 GB. Some specifications of the Zynq are given in Table 4.

Table 4: Xilinx Zynq-7020 Specifications

ARM Specifications	
L1 Cache Per Core	32 KB Instruction / 32 KB Data
L2 Cache Shared	512 KB
Maximum Clock Frequency	667 MHz
FPGA Specifications	
Programmable Logic Cells	85,000
Look-Up Tables	53,200
Flip-Flops	106,400
DSP Slices	220 (18 x 25 MACCs)

From a reliability perspective, the pairing of these two architectures on a single die is of considerable utility for a space computer. They are able to communicate with and monitor each other, permitting a relatively novel manner of fault-tolerance. The specifics of the mechanisms of this fault-tolerance are given in the section on software architecture.

3.3 Hardware Reliability Assurance

A number of mechanisms for ensuring a reliable system are designed into the hardware of CSPv1. One of these mechanisms is a power-sequencing circuit (RadHard if necessary) which serves two functions: (1) ensure that the power rails are sequenced correctly; and (2) if a regulator fails transiently, keep the Zynq in a protected reset state and send a control-flow signal over the connector so that appropriate recovery action can be taken.

In the event of a software fault which is not recoverable by software alone, the supervisory and reset management circuits (all of which can be made RadHard) will assist the Zynq to return to an operational state. The principle on which these circuits operate is continuously listening for a *heartbeat* signal from the Zynq and, in the event of the Zynq failing to provide this signal, the supervisory and reset management circuits will reset the Zynq and allow it appropriate time to boot up and send out the heartbeat again.

3.4 System Integration

All externally facing connections on a CSPv1 are made through the 160-pin Samtec Searay connector. In addition to debugging and control-flow signals, there are 60 high-speed connections from the FPGA portion of the Zynq, and 26 high-speed connections from the ARM portion of the Zynq. Of these 60 FPGA pins, 48 of them can be configured as (24) differential-pairs for use with various high-speed interfaces. The ARM pins can be configured to be most any combination of I^2C, SPI, CAN, and UART interfaces, or used as GPIO.

Figure 2: CSPv1 Mated with Evaluation Board

For ground testing and evaluation, CSPv1 makes use of an evaluation board with a considerable amount of convenient circuitry to permit rapid development. This board is shown, with a CSPv1 mated, in Figure 2. From the CSP's FPGA signals, the evaluation board provides connectors for Camera Link, SpaceWire, and a number of spare single-ended and differential signals. From the CSP's ARM signals, the evaluation board has gigabit Ethernet, and USB Host, capability. A secondary purpose of the evaluation board is to serve as a reference design for the integration of various interfaces and devices.

A typical system which includes a CSPv1 will have a passive backplane, to which one or more CSPs are connected, along with a power-supply board to supply the CSPs. CSPv1 requires the input of two rails, 3.3 Volts and 5.0 Volts; the remaining voltages are generated on-board. Some typical power requirements for a single CSPv1 board, in different states, are given in Table 5. Though the table does not reflect it, we do anticipate that it is possible to produce FPGA designs, likely with a large number of external interfaces, which

could bring the power-profile of the board up to around 4 Watts.

Table 5: CSPv1 Power Consumption

Clocks (MHz)			Test Load		
ARM	**DDR**	**FPGA**	**ARM**	**FPGA**	**Power (W)**
200	200	100	Low	Low	**1.54**
200	200	100	High	Low	**1.78**
667	533	100	High	Low	**2.23**
667	533	100	High	High	**2.86**

4 SOFTWARE ARCHITECTURE

A body of software has been developed for, and applied to, CSPv1 in order to both support it as a space computer and improve system reliability. The specifics of how software is used to improve reliability is considerably different for each half (ARM and FPGA) of the Zynq.

4.1 Utility Software

In order to facilitate development of space applications, a number of custom software components were developed for CSP. CSPv1 runs a custom, lightweight, Linux-based operating system on the ARM cores of the Zynq which has extensions to support various operations, such as FPGA scrubbing (detailed later) which are necessary for reliable operation. Other extensions include those enabling easy communication with the FPGA side of the Zynq, which could, among other things, allow the FPGA to be used as an accelerator to the CPU program execution. It is also worth noting that there are a number of options for program acceleration with the Zynq. In addition to FPGA-based acceleration, the Zynq also has a NEON floating-point SIMD accelerator with each core, and the two cores in the ARM allow either OpenMP-based or MPI-based acceleration.

Concerning flight software, CSPv1 has integrated the Core Flight Executive (cFE) flight software framework [13] (designed specifically for satellites and instruments on embedded platforms) and the Core Flight System (CFS) [13] from NASA Goddard. With these software packages, a number of different applications for commanding, telemetry gathering, scheduling, health and safety, and file management are readily available.

4.2 ARM Software Reliability Assurance

Within the Zynq, the ARM processor is, to a degree, the master device over the FPGA and other parts of the system—it is connected to the non-volatile memory of the system, and is responsible for configuring and initializing the FPGA. As such, a great deal of care must be taken to ensure that the ARM device boots up properly. A number of precautions have been taken in the hardware and software of CSPv1 to make sure that this is the case. An outline of the boot sequence is shown in Figure 3. There are two types of fallback mechanism, depending on the stage in the sequence.
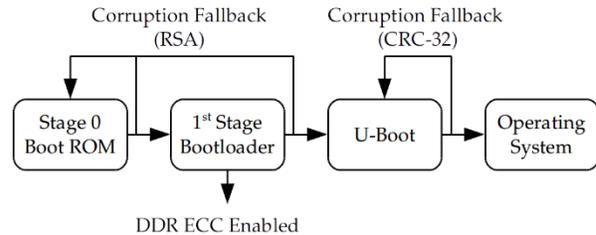
Figure 3: CSPv1 Boot Sequence

The first precaution in the boot-up sequence is a method of verifying the correctness of boot images contained in the non-volatile memory. The CSP repurposes the RSA authentication features of the Zynq to check boot images. The intended purpose of this feature is tamper-prevention; however, it has proven very capable of verifying the correctness of non-volatile memory contents. Additionally, it is possible to use fallback (an arbitrary number of times), if the image from which the Zynq is trying to boot is corrupted.

After the verification of a boot image, it is necessary to ensure that this image will remain uncorrupted as it moves to the volatile memory and is used by the processor. Memory errors are logged and corrected through several mechanisms at every level of the memory hierarchy. At the lowest level, both the L1 and L2 caches in the Zynq can trigger interrupts on parity failure. In main memory, ECC can be enabled (halving the usable memory capacity) which will automatically detect two bits, and correct one bit, of error per word in the DDR. Using error-detection and error-correction extensions to the kernel, these DDR faults can also be monitored and managed by software. Finally, in the NAND flash, there are several features which will likely ensure that no images, or scientific data, are corrupted. The Zynq controller for the NAND device has built-in support for ECC, and its reliability is augmented by bad-block support.

During normal operation (after boot-up), there are several measures to ensure correct program execution. In addition to the external supervisory circuit, the Zynq has three internal watchdogs (one for each ARM core, and one which monitors device hardware) which can be used to detect and correct faults. Additionally, several

fault-tolerant software techniques, including temporal redundancy and program health monitoring, are to be used on flight systems.

4.3 FPGA Software Reliability Assurance

One of the challenges of deploying an FPGA system in space is that its SRAM-based memory architecture makes it susceptible to high-energy particles which can cause corruption of the memory cells. A Single-Event Upset (SEU), a common occurrence in space environments, can lead to bit flips in configuration or data memory of the FPGA which can eventually lead to data errors or device failure if left unmitigated. This hazard is also an issue for the CSP; however, corrective measures have been included to prevent catastrophic system failure. The solution is configuration scrubbing, the process of quickly repairing these configuration bit upsets in the FPGA before they can accumulate to the point of rendering the device inoperable. Several techniques for scrubbing have been deployed in space missions [14], but each has tradeoffs with respect to detection and correction time, coverage of upset patterns, and fault localization.

The configuration scrubber within CSP uses a *readback* strategy that continuously reads configuration data frame by frame (FPGA configuration data is organized into *frames,* with each frame consisting of 101 words or 3232 bits). Each configuration frame is compared to the *golden* frame contents stored in main memory. The golden frame data is generated by the vendor's bitstream-generation tools. A frame mask is also used to mask configuration bits within a frame that are known to change during operation. If a difference exists between a non-masked bit within the readback frame and the golden frame, the configuration frame is overwritten with the contents of the golden frame. The location (frame, word, and bit) of any upsets within the configuration data is logged for post-processing and error analysis.

This scrubber only checks the configuration frames associated with the logic portions of the configuration bitstream (Block 0 frames). The configuration frames associated with internal block memory or BRAM are not scrubbed, since their contents change with normal functionality. In addition, BRAMs can be protected from SEUs by utilizing the built-in support of error correction coding (ECC).

One of the unique features of the CSP scrubber is that it supports partial reconfiguration of the FPGA logic. The scrubbing process is paused during partial reconfiguration to prevent the scrubber from "fixing" the configuration memory. Partial reconfiguration is then performed to update the configuration data with the new hardware circuitry. Before the scrubber resumes, the golden frame data and mask information associated with the partial reconfiguration is updated to reflect a change in the hardware, so that future scrubbing will not undo the partial reconfiguration. Once scrubbing resumes, the new partial configuration data is embedded in the golden file and will be used to verify the contents of the FPGA circuitry.

4.4 System Reset Methodology

The supervisor circuit on CSPv1 will reset the Zynq if it fails to put out a heartbeat signal. However, the exact strategy of determining what should cause the Zynq to stop outputting its heartbeat is a considerable research topic and design decision. Nominally, the heartbeat should stop only when the Zynq has encountered a fault for which it is completely unable to recover without assistance from the external circuit.

In order to achieve this outcome, the CSP uses a multi-level heartbeat. The externally facing heartbeat (which is accepted by the optionally RadHard supervisor from Intersil) is sourced from the FPGA, and is internally connected to a number of different pseudo-heartbeats from the different parts of the Zynq device. Some of these pseudo-heartbeats come from the ARM side of the chip (these are application health information), and there are also pseudo-heartbeats from the different instantiated FPGA hardware elements. The intent of this scheme is that if some internally monitored element fails, its manager will attempt to reset or reinitialize it, and if that is not possible, the external heartbeat will stop, triggering a full system reset.

5 SUMMARY AND CONCLUSIONS

In this paper, we introduced the CHREC Space Processor (CSP) as a new approach based upon the concept of hybrid space computing. By featuring COTS devices to perform the critical data processing, and simpler RadHard devices to monitor and manage the COTS devices, and augmenting them both with novel fault-tolerant computing techniques, CSP can maximize performance and reliability while minimizing energy consumption and cost, providing unprecedented computing capability in space.

CSPv1, the first production version of CSP, is designed to be selectively populated in various combinations of RadHard and COTS on the same PCB board, permitting a spectrum of possible hybrid boards for different mission requirements. It is also designed to be scalable in application scope — by the use of multiple CSP cards in a system, most any computing performance requirement can be accommodated, from CubeSats to large space systems.

CSPv1 boards and their corresponding evaluation boards are currently being tested by various industry, government, and university groups partnered in CHREC. Among these tests, CSPv1s (and the Zynq chip on which they are based) are being evaluated, or queued to be evaluated, for ionizing-radiation-induced upset performance, thermal and vacuum performance, and shock and vibration performance. The boards are currently scheduled to fly on two upcoming missions through NASA Goddard. One is a technology mission (STP-H5/ISEM) on the International Space Station in which two CSPs will be paired together to evaluate hardware performance and collaborative algorithms in a space setting. The second (CeREs) is a heliophysics science mission in which a single CSP will be performing data processing in a small NASA satellite.

6 ACKNOWLEDGEMENTS

7 REFERENCES

1. Brown, G.R., "Radiation Hardened PowerPC 603e Based Single Board Computer," Proc. of 20th Digital Avionics Systems, Oct. 2001.

2. Berger, R.W. et. al., "RAD750 SpaceWire-Enabled Flight Computer for Lunar Reconnaissance Orbiter," Proc. of International SpaceWire Conference, Dundee, Scotland, Sep. 2007.

3. Lovelly, T. M., Cheng, K., Garcia, W., and A. D. George, "Comparative Analysis of Present and Future Space Processors with Device Metrics," Proc. of Military and Aerospace Programmable Logic Devices Conference (MAPLD), San Diego, CA, May 19-22, 2014.

4. Barnaby, H.J., "Total-Ionizing-Dose Effects in Modern CMOS Technologies," IEEE Transactions on Nuclear Science, vol. 53, no. 6, pp. 3103-3121, Dec. 2006.

5. Howard, J.W., Carts, M.A., Stattel, R., Rogers, C.E., Irwin, T.L., Dunsmore, C., Sciarini, J.A., and K.A. LaBel, "Total dose and single event effects testing of the Intel pentium III (P3) and AMD K7 microprocessors," Proc. of Radiation Effects Data Workshop, pp. 38-47, 2001.

6. Fabula, J., and H. Bogrow, "Total ionizing dose performance of SRAM-based FPGAs and supporting PROMs," Proc. of Military and Aerospace Programmable Logic Devices Conference (MAPLD), Laurel, MD, Sep. 2000.

7. Ceschia, M., Violante, M., Reorda, M.S., Paccagnella, A., Bernardi, P., Rebaudengo, M., Bortolato, D., Bellato, M., Zambolin, P., and A. Candelori, "Identification and classification of single-event upsets in the configuration memory of SRAM-based FPGAs," IEEE Transactions on Nuclear Science, vol. 50, no. 6, pp.2088-2094, Dec. 2003.

8. Reinhardt, S.K. and S. Mukherjee. 2000. "Transient fault detection via simultaneous multithreading." SIGARCH Comput. Archit. News 28, 2 (May), 25-36. 2000.

9. Shye, A., Blomstedt, J., Moseley, T., Reddi, V.J., and D.A. Connors, "PLR: A Software Approach to Transient Fault Tolerance for Multicore Architectures," IEEE Transactions on Dependable and Secure Computing, vol. 6, no. 2, pp.135-148, April-June 2009.

10. Carmichael, C., Fuller, E., Blain, P., and M. Caffrey, "SEU mitigation techniques for Virtex FPGAs in space applications," Proc. of Military and Aerospace Programmable Logic Devices Conference (MAPLD), Laurel, MD, Sep. 1999.

11. Garvie M. and A. Thompson, "Scrubbing Away Transients and Jiggling Around the Permanent: Long Survival of FPGA Systems Through Evolutionary Self-Repair," Proc. of International On-Line Testing Symposium, July 12-14, 2004.

12. Ostler, P., Caffrey, M., Gibelyou, D., Graham, P., Morgan, K., Pratt, B., Quinn, H., and M. Wirthlin, "SRAM FPGA reliability analysis for harsh radiation environments," IEEE Trans. on Nuclear Science, vol. 56, no. 6, 2009.

13. Wilmot, J. "Implications of responsive space on the flight software architecture." Proc. of AIAA Responsive Space Conference, 2006.

14. Martin, Q. and A.D. George, "Scrubbing optimization via availability prediction (SOAP) for reconfigurable space computing," Proc. of IEEE Conference on High-Performance Extreme Computing (HPEC), Waltham, MD, Sep. 10-12, 2012.