

# A Network-on-Chip for Radiation Tolerant, Multi-core FPGA Systems

Justin A. Hogan, Raymond J. Weber and Brock J. LaMeres  
Electrical and Computer Engineering Department  
Montana State University  
Bozeman, MT 59717  
justin.hogan@msu.montana.edu  
raymond.weber@msu.montana.edu  
lamer@ece.montana.edu

**Abstract**— This paper describes research efforts to mitigate weaknesses in a TMR+spares radiation tolerant SRAM-based FPGA computer system. An existing 9-tile Microblaze architecture is reviewed and the desired improvements of fault-mitigated routing, fault location determination and performance enhancement via runtime-configurable hardware accelerators are discussed. Hamming encoding is proposed as a method for protecting the routing resources from radiation-induced single event upsets and as feedback to the computer's configuration control system to distinguish faults occurring in routing from those occurring within partially reconfigurable processing tiles. This is important as the recovery operation for each of these conditions is unique. Without the ability to distinguish routing faults from tile faults, routing faults are aliased as tile faults and unnecessary tile repair steps are taken. In addition to the protected routing with configuration control feedback, architecture for implementing TMR, processor-peripheral hardware accelerators is introduced.

## TABLE OF CONTENTS

1. INTRODUCTION .....	1
2. PREVIOUS WORK .....	2
3. SYSTEM LIMITATIONS .....	3
4. OUR SOLUTIONS .....	4
5. CONCLUSIONS .....	6
REFERENCES.....	7

## 1. INTRODUCTION

Field programmable gate arrays (FPGAs) have gained wide acceptance in aerospace applications as a result of their desirable combination of high performance, low cost, low power and design flexibility. In high bandwidth, computationally intensive applications FPGAs stand as a compromise between custom ASICs and traditional computer processors. FPGAs enable interfacing with high data rate instruments through the use of optimized custom logic cores, are capable of processing large data sets such as those generated by high-resolution imaging systems, and contribute to onboard data reduction through the use of real-time digital signal processing routines. This data reduction eases the onboard data storage requirements and reduces the amount of downlinked data. These attributes are particularly useful as payloads grow in complexity in an effort to maximize scientific value.

FPGAs, particularly SRAM-based devices, provide a unique flexibility to aerospace systems. The user design for these devices is stored in volatile static RAM cells, which must be initialized at power-up. A single FPGA can be used to implement multiple system functions by loading different configuration bitstreams based on current system needs or operating mode. This allows hardware sharing by non-concurrent processes that would otherwise require independent hardware systems resulting in an overall reduction in component count and system complexity. More advanced configuration features, such as active partial reconfiguration, allow specific portions of an FPGA to be reprogrammed without affecting the operation of the rest of the FPGA. This allows hardware peripherals to be instantiated on an as-needed basis resulting in power savings through an overall reduction in device resource utilization. Yet another benefit of FPGAs is the ability to change the implemented hardware design at any time during a system lifecycle in response to design errors, technology advancement or evolving mission requirements. In space systems this capability is highly advantageous as modifying hardware post-launch is problematic for obvious reasons.

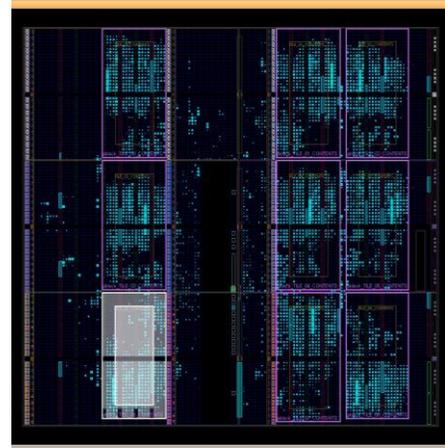
Aerospace environments present well documented challenges to commercial-grade SRAM-based FPGAs as the memory elements that give the devices their flexibility are themselves susceptible to faults induced by interactions with high-energy ionizing radiation. Radiation effects can generally be broken down into two categories based on the nature of their interactions with the device's constituent materials. Total ionizing dose (TID) is a material-dependent measure of energy deposition by ionizing radiation [1]. The materials of most interest in CMOS devices are the gate and field oxides as energy deposition can result in trapped charge and a gradual degradation of the characteristic electrical properties or an increase in leakage current [2]. As an ionizing particle transits through the gate oxide it leaves behind a track of electron-hole pairs. Differences in electron and hole mobility and a variety of charge trapping mechanisms result in persistent holes in the oxide. Over time, this charge builds up and affects transistor electrical properties, such as the threshold voltage [2]. These effects are strongly dependent on the thickness of the gate oxide and tend to diminish with decreasing oxide thickness. In deep sub-micron process nodes (< 45-nm), TID is less of a concern to FPGA device reliability in aerospace application

than the second category of radiation effects, known as single event effects (SEE).

Single event effects describe the transient effects related to radiation interactions. There are multiple subcategories of SEEs including single event transients (SET), single event functional interrupts (SEFI), single event upsets (SEU), and single event latchup (SEL) [3]. SETs, SEUs and SEFIs are closely related as they differ in where they occur spatially within the device. Whereas SEL represents a potentially damaging condition in which parasitic transistors are activated resulting in a high current draw, SETs, SEUs and SEFIs are considered to be soft errors correctable by writing valid data into the affected memory element. SETs occur when ionizing radiation induces a transient voltage in a combinational logic circuit. This erroneous signal only temporarily affects the combinational logic signals and dissipates before being latched into a system memory element [4]. If a SET is latched into a memory element it is known as an SEU. SEUs are faults affecting either user memory or configuration memory contents. SEFIs are a subcategory of SEUs that affect core device configuration functions including the configuration controller, clock management tiles, etc [3]. Single event effects are of primary interest due to their increasing rate of occurrence with current semiconductor device technology scaling trends [5,6]. For a more comprehensive review of single event effects consult [6].

## 2. PREVIOUS WORK

The focus of this research is to build upon the traditional fault mitigation techniques in an effort to increase the performance and reliability of SRAM FPGAs for aerospace applications. The approach to accomplishing this is to combine readback scrubbing, active partial reconfiguration, and TMR in a specific way to efficiently detect and mitigate radiation induced faults while minimizing fault recovery time. The FPGA is partitioned into discrete, partially reconfigurable processing resources. These are referred to as “tiles”, and they represent the granularity of the TMR implementation. Our current research system, implemented on a Xilinx Virtex-6 FPGA, consists of nine tiles each of which contains a Microblaze microprocessor. During normal operation, three tiles are active and constitute an active triad. The outputs of the active tiles are routed through a multiplexer to a majority voter to form a complete TMR system. In the background, and without impacting the operation of the active triad, a scrubbing routine maintains the spare tiles using active partial reconfiguration. An external configuration controller monitors the status of the TMR+spares system, controlling which tiles are active, performing configuration readback and scrubbing, and tracking the status of each of the tiles. Figure 1 depicts a TMR+spares system architecture.



*Figure 1 – This figure shows the FPGA floorplan for a 9-tile TMR+spares system. The FPGA fabric is partitioned into nine discrete elements each of which contains a Xilinx Microblaze soft processor.*

The configuration controller is responsible for detecting and recovering from faults in the system. The fault detection mechanism is dependent upon fault location. A SEU occurring in one of the active tiles is both detected and mitigated by the voter circuit. The voter circuit alerts the configuration controller to the faulted tile and by design prevents the fault from propagating to the system output. In this way the system tolerates faults in the active tiles. Upon detection of a faulted active tile, the configuration controller initiates a tile switch wherein a healthy spare tile is substituted for the faulted tile in the active triad. The tiles are synchronized and system operation is resumed. Active partial reconfiguration is then used to repair the faulted tile. After repair, the tile is reintroduced into the system as an available spare tile and the system is returned to a nominal state. The ability to bring healthy spare tiles online in place of faulted tiles, and repair faulted tiles in the background is advantageous in high fault rate environments. **It is generally faster to switch from a faulted tile to a healthy spare than it is to wait for the readback scrubber to correct the fault. Table 1 shows the approximate recovery times for each method. The times are design-dependent and vary according to bitstream length of the tiles. The recovery time estimate for readback scrubbing is a worst-case value for a system scrubbing all bits. This time could be improved by only attending to the configuration memory essential bits.**

Table 1—This table highlights the latency for various recovery techniques. Equivalent configuration clock speed is 1.3-MHz, 8-bit SelectMAP interface. The speed is limited by configuration data retrieval from the MicroSD storage device.

Recovery Method	Recovery time
Full configuration	~2.5-s
Tile PR	126-ms
Blind tile scrub rate (9 tiles)	1.1-s
Tile switch	200- $\mu$ s

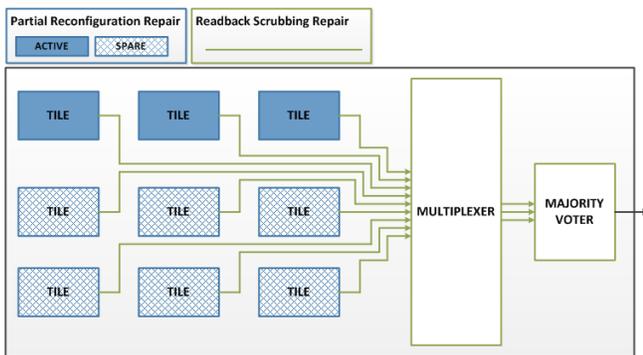


Figure 2—This figure shows the general architecture of a nine-tile TMR+spares system. Blue features represent system elements that are repaired via partial reconfiguration. Green elements are repaired by the scrubbing routine.

Since the fault rate is highly dependent on orbit, location within orbit, and recent solar activity, the number of available spare tiles can be adjusted accordingly. Reducing the number of maintained spares can provide a reduction in static power during low fault rate conditions. Similarly, the configuration memory scrubbing activity can be throttled based on observed or expected fault rates.

Spare tiles are maintained in a healthy state by a blind scrubbing routine performed by the configuration controller. This requires the partial bitstream for each of the tiles to be loaded at a user-defined scrub rate. A SEU occurring in the configuration memory for the routing between a tile and the voter will be aliased as a faulted tile, and the previous steps of repairing the tile will have been performed unnecessarily. Upon reactivation of that tile, the fault in the routing would be detected again. To prevent this from happening, the scrubber continuously checks the configuration memory of

the entire FPGA and cleans up faults affecting the routing which aren't fixed in the tile repair process. This prevents faults from accumulating in the static portion of the FPGA design.

The fault-tolerant research architectures are implemented on a custom research hardware platform shown in Figure 3. A Xilinx Virtex-6 (XC6VLX75T) device, referred to as the “main FPGA” plays host to the research systems. The configuration control and user interface functionality are implemented on a Spartan-6 (XC6SLX75) device, referred to as the “control FPGA”. A USB interface provides communication between the test platform and a host PC. The printed circuit board conforms to CubeSat form-factor requirements for eventual inclusion in a orbital or suborbital payload electronics stack.

The control FPGA self-configures at power-up via a Master-Serial connection to a Xilinx Platform Flash component. Once configured, the control FPGA reads the bitstream for the main FPGA from a MicroSD card and loads it over an 8-bit SelectMAP interface. This same interface is used to perform readback, scrubbing, active partial reconfiguration, and eventually fault injection on the main FPGA. A large GPIO bus between the two devices allows application data to be passed between the devices as well as any control signals necessary for system testing.

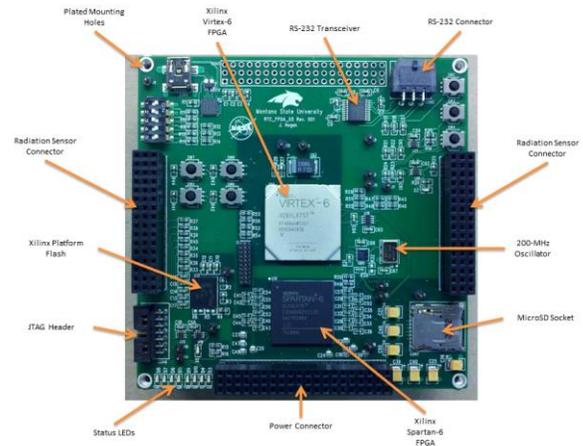


Figure 3—This figure shows the 4" by 4" custom FPGA board used as a test platform for radiation tolerant architectures. This board features a Virtex-6 and a Spartan-6. Virtex configuration is controlled by the Spartan FPGA via an 8-bit SelectMAP interface.

### 3. SYSTEM LIMITATIONS

The TMR+spares system has some weaknesses that can be improved through straightforward design enhancements. The first weakness is the inability to distinguish faults occurring in a tile from faults occurring in the routing. As the recovery mechanism is different for each, it is desirable to isolate fault locations. A second limitation is the lack of

inter-tile communication. Simply running soft processors in TMR reduces the FPGA to a simple microcontroller. With inter-tile communication, the processor tiles gain the ability to instantiate custom logic cores as peripheral devices to offload computationally intensive tasks. The ability to instantiate peripheral hardware accelerators makes far better use of the FPGA resources.

#### 4. OUR SOLUTIONS

Isolating fault location is necessary as it can be used as feedback to the configuration control system to more efficiently handle the fault recovery process. The technique we propose to accomplish this leverages the commonly used Hamming encoding process to add redundancy to the routing [15]. Hamming encoding is a technique for detecting and correcting single bit errors and detecting multiple bit errors by adding carefully calculated check bits to transmitted data. These check bits constitute a check code, which is used by the data receiver to determine if any bits were corrupted during transmission. In the case of a single bit upset, the Hamming decoding process, which is performed by the receiver, yields a binary code whose decimal value indicates which bit position was corrupted. Inverting the corrupted bit yields the original, uncorrupted data. The added bits are used to detect multiple bit errors and detect/correct single bit errors. Figure 4 provides some details on the generation of the Hamming code for a 32-bit word.

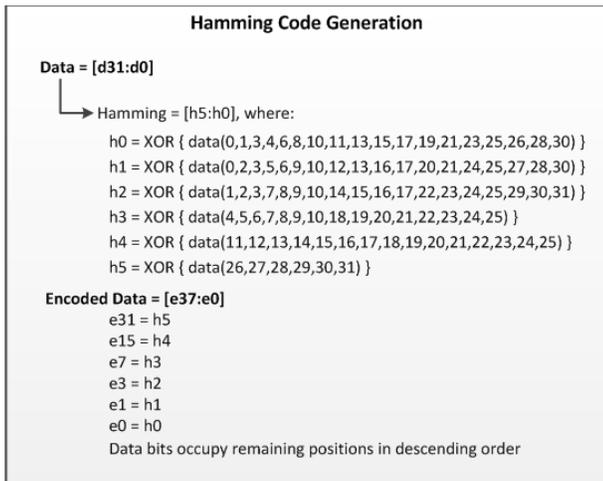


Figure 4—This figure shows the process for generating the check bits for the Hamming encoding process. Hamming bits are simply a series of even parity checks applied to particular bits such that the location of a bit upset can be determined by the decoder.

The use of Hamming codes to protect FPGA memory and register contents against various upsets has been previously demonstrated, e.g. [16]. In addition to using the Hamming code to recover the original data in the event of a single-bit upset, the detection of an upset in the Hamming decoding process implies that the error occurred between the encoder and the decoder. By placing an encoder immediately at the boundary of a processing tile, the amount of routing protected from faults is maximized. Faults occurring in a tile are detected by the majority voting process, while faults

occurring in the routing between a tile and the voter are detected by the Hamming decoding process. The system configuration controller uses this information to either initiate a tile swap in the event of a tile fault, or direct the scrubber to the affected routing region. The TMR architecture can tolerate multiple errors within the data path of a single active triad member, so the system controller can allow operation to continue in the presence of a detected routing upset with the understanding that the routing faults will, in time, be corrected by the scrubbing routine.

Table 2—This table shows the recovery options for various fault conditions.

Fault Location	Detection Method	Recovery Method
Tile	Majority voter circuit	Tile swap, background repair
Inter-tile routing	Hamming decoding circuit	Scrub

A routing protection test system was designed and implemented on a Xilinx Virtex-6 device. In this system, simple counters generated 32-bit data that was Hamming encoded using six check bits. Tile faults were simulated using a design level injection circuit that inverted a single data bit prior to encoding. Routing faults were simulated using a design level injection circuit that inverted any one of the 38-bit Hamming encoded vector bits. The routing faults were induced between the Hamming encoder and decoder circuits. Figure 5 depicts a block diagram of this system.

Each counter output is routed through a fault injection circuit capable of inverting individual bits determined by a control signal from an external processor. The data are input to Hamming encoder circuits, which generate the proper check codes used to protect the data in the routing between the source tile and the voter circuit. The data pass through another fault injection circuit used to simulate single-bit routing faults. Finally, the data are decoded and input into a majority voter circuit. If a single bit error is indicated by any of the check codes then an error in the routing is suspected to have occurred. If the data produced by a tile is deemed incorrect by the majority voting circuit, a tile fault is suspected to have occurred.

In the TMR+spares system this encoding scheme performs two important tasks: (1) it identifies faults occurring in the interconnect between two tiles allowing them to be corrected by the scrubbing routine, and (2) it mitigates the faults in the data allowing the system to continue normal operation. This preserves the integrity of the data used by the voting, and allows faults in the interconnect to be identified and corrected. The detection of multiple bit upsets in the interconnect is also useful because such a condition may interfere with proper system operation.

Multiple bit error detection allows a tile swap to be initiated in advance of computations being performed on corrupt data. Error detection and correction allows a greater understanding of the nature of faults occurring within the TMR+spares architecture.

Figure 5 shows the FPGA floorplan for the routing protection test system. In this system the resource utilization of the encoder/decoders is comparatively large when placed adjacent to the simple counter blocks. When used in a practical system the percentage increase in overhead will be significantly reduced. Additionally, the encoders/decoders themselves are susceptible to upset. The susceptibility can be reduced through application of fine-grained TMR, or they can be left unmitigated with the knowledge that should the encoder/decoder be faulted, the system simply reverts to being unable to resolve the location of the fault. The likelihood of this occurring depends on the area of the encoder/decoder circuitry.

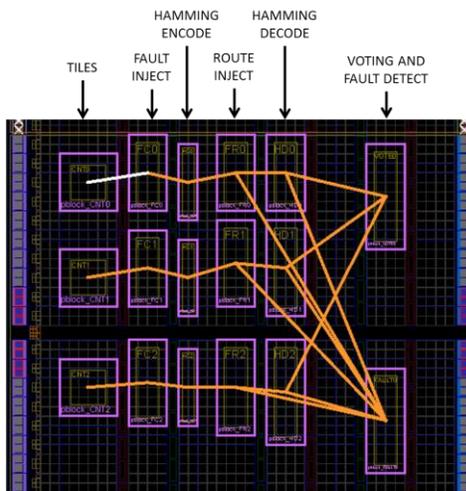


Figure 5—This figure depicts the FPGA floorplan for the TMR routing protection test system, noting that the resource utilization is extremely small for this simple system and all component blocks are grouped closely together.

Figure 6 shows the timing diagram for a routing fault simulation including the original data, the Hamming encoded data, the routing-faulted data, the Hamming decoded and corrected data, the Hamming check word representing which bit number was corrupted, and the final output of the TMR system. In this experiment, the routing fault was induced by performing an exclusive-OR operation of the Hamming encoded data with a fault mask bit vector. This is representative of an error occurring in a memory element along the routing path.

This is fundamentally different from errors occurring in configuration memory, but the effects of each manifest in the same way at the design level. This experiment shows that by encoding/decoding data over a routing path single event upsets occurring in the configuration memory along said path can be mitigated, and system operation can proceed as normal with the knowledge that the fault will eventually be corrected by a scrubbing routine. As single-bit upsets occurring in routing can be corrected, there is no need to take corrective action. In fact, such a system can tolerate single-bit upsets in the routing in each of the three TMR paths, and multiple-bit upsets in a single data path without requiring targeted recovery action. The system is able to tolerate a limited accumulation of errors, which may be useful in a high fault rate environment. The system presented is a simple proof-of-concept and is easily applied to the aforementioned 9-tile system, or other practical research systems.

The 9-tile system highlighted previously featured partially reconfigurable regions each containing a Microblaze soft processor. Each processor executed an identical counting program, which was used to test the general operation of the radiation tolerant architecture. This system was useful for developing the requisite reconfigurable computing tools such as SelectMAP device configuration, active partial reconfiguration, configuration memory blind scrubbing, and configuration memory readback. However, it is recognized that this is a very restricted use of the FPGA capabilities that make these devices so relevant to space applications.

The next step in this research is to increase the performance of the computer system by making use of the flexibility of the FPGA fabric. A research area currently being explored is runtime instantiation of custom hardware accelerators as peripheral devices to the Microblaze processors. This allows the processors to offload computationally intensive tasks to faster and more efficient computational resources rather than performing operations in software. As there are three microprocessors active at any time in a TMR system, it is also required that three hardware accelerators be instantiated, one for each processor. This presents a problem similar to the unprotected routing wherein fault location is indistinguishable. A fault occurring in a peripheral accelerator would eventually be detected as a tile fault by the voter circuitry. To recover, the configuration control system would have to repair both the accelerator and the processor tiles. Therefore, the hardware accelerators must be instantiated in a TMR arrangement.

Bus/Signal	X	O	1	2	3	4	5	6	7	8	9	10	
COUNT_0	0000	0000000	00000000	00000001	00000002	00000003	00000004	00000005	00000006	00000007	00000008	00000009	0000000A
COUNT_FAULT0	FFFF	FFFFFFFF	FFFFFFFFF3	00000000	00000001	00000002	00000003	00000004	00000005	00000006	00000007	00000008	00000009
COUNT_HC	3F7F	3F7FFFFF	3F7FFFFF3	3F7FFFFF4	000000000	000000007	000000019	00000001E	00000002A	00000002D	000000033	000000034	00000004B
ROUTE_FAULT	01	01	01										
ROUTE_FAULT0	3F7F	3F7FFFFF	3F7FFFFFEC	3F7FFFFF2	3F7FFFFF5	000000001	000000006	000000018	00000001F	00000002B	00000002C	000000032	000000035
COUNT_HD	FFFF	FFFFFFFF	FFFFFFFB	FFFFFFFC	FFFFFFFD	FFFFFFFE	FFFFFFF7	00000000	00000001	00000002	00000003	00000004	00000005
CHECK_CODE0	01	01	01										
COUNT	FFFF	FFFFFFFF	FFFFFFF9	FFFFFFFA	FFFFFFFB	FFFFFFFC	FFFFFFFD	FFFFFFFE	FFFFFFF7	00000000	00000001	00000002	00000003

Figure 6– This figure demonstrates a simulated routing fault. The original count, the faulted Hamming encoded data, the recovered data, the Hamming check code, and the TMR output values are depicted.

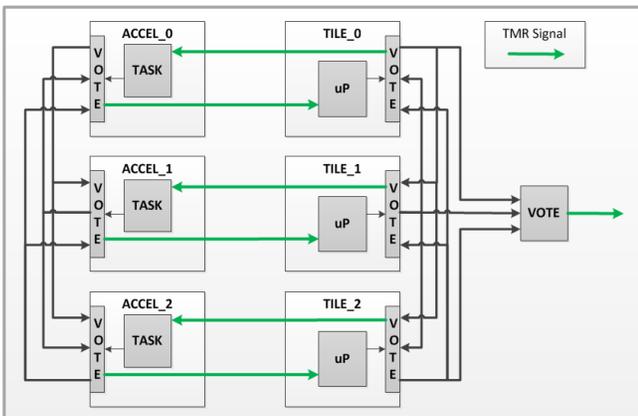


Figure 7– Proposed architecture for implementing TMR hardware accelerators as processor peripherals in a radiation tolerant FPGA system.

Figure 7 depicts the proposed architecture for implementing custom hardware accelerator cores as peripheral devices to a triad of TMR processor tiles. In this system, each tile contains a voter circuit connected to the other two like-tiles in the system. This allows the data passed between the accelerators and the processing tiles to be fault-mitigated using TMR. Since each tile contains a voter, the results of which can be passed to the configuration control system, faulted tiles can be identified and repaired efficiently. A faulted accelerator tile continues to feed correct data to its parent processor tile as the output is a majority vote from all three of the accelerators. Similarly, a parent processor tile feeds TMR-ed data to its accelerator allowing the accelerator to continue operating properly in the event of a processor fault. This arrangement isolates single faults allowing the rest of the system to continue operating nominally. Optionally, the TMR output from each tile can be used as feedback into the local logic to aide in the synchronization of newly repaired tiles. This TMR with feedback is commonly used in radiation tolerant FPGA applications [17]. The distribution of the voter to each of the tiles eliminates it as a single point-of-failure in the

system. Though the processor output is depicted going to an external voter circuit, any one of the tile-to-accelerator TMR signals could conceivably be passed to the device output rather than implementing another voter.

## 5. CONCLUSIONS

This paper describes research efforts to mitigate weaknesses in a TMR+spares radiation tolerant SRAM-based FPGA computer system. An existing 9-tile Microblaze architecture is reviewed and the desired improvements of fault-mitigated routing, fault location determination and performance enhancement via runtime-configurable hardware accelerators are discussed. Hamming encoding is proposed as a method for protecting the routing resources from radiation-induced single event upsets. In addition to fault mitigation, the presence of faults in the Hamming encoded data allows the computer’s configuration control system to distinguish faults occurring in routing from those occurring within the partially reconfigurable processing tiles. This is important as the recovery operation for each of these conditions is unique. Without the ability to distinguish routing faults from tile faults, routing faults were aliased as tile faults and unnecessary tile repair steps were taken.

In addition to the protected routing with configuration control feedback, architecture for implementing TMR, processor-peripheral hardware accelerators was introduced. This architecture allows the system to continue operating in the event of a single faulted tile. Each tile contains a voter circuit allowing TMR data to be passed between processor tiles and accelerators, and vice versa. This eliminates the voter as a single point-of-failure in the system at the cost of increased resource utilization. Future work in this area will move to a TMR+spare architecture which involves the addition of spare tiles to both the processor and peripheral triads, allowing a context switch from a faulted tile to a healthy tile. As in the 9-tile system, faulted tiles are then repaired in the background via active partial reconfiguration without interfering with foreground operations. Lastly, the

routing among the tiles will be Hamming encoded to further enhance the fault tolerance and increase the spatial awareness of fault locations.

## REFERENCES

- [1] A. Holmes-Seidel and L. Adams, *Handbook of Radiation Effects*. London, U.K.: Oxford Univ. Press, 2002.
- [2] T. Oldham and F. McLean, "Total ionizing dose effects in MOS oxides and devices," *IEEE Trans. Nucl. Sci.*, vol. 50, no. 4, pp. 483-499, Jun. 2003.
- [3] D. White, Considerations surrounding single event effects in FPGAs, ASICs, and processors, XAPP402 (v1.01), Xilinx Corp., 2012.
- [4] G. Wirth, M. Vieria, E. Neto, and F. Kastensmidt, "Single event transients in combinatorial circuits," *Integrated Circuits and Systems Design, 18<sup>th</sup> Symposium on*, pp. 121-126, Sept. 2005.
- [5] J. Benedetto, P. Eaton, D. Mavis, M. Gadlage, and T. Turflinger, "Digital single event transient trends with technology node scaling," *IEEE Trans. Nucl. Sci.*, vol. 53, no. 6, pp. 3462-3465, Dec. 2006.
- [6] P. Dodd and L. Massengill, "Basic mechanisms and modeling of single event upset in digital microelectronics," *IEEE Trans. Nucl. Sci.*, vol. 50, no. 3, pp. 583-602, June 2003
- [7] R. Lyons and W. Vanderkulk, "The use of triple-modular redundancy to improve computer reliability," *IBM Journal of Research and Development*, IBM, vol. 6, pp. 200-209, 1962.
- [8] C. Carmichael, Triple module redundancy design techniques for Virtex FPGAs, XAPP197 (v1.0), Xilinx Corp., 2001.
- [9] F. Kastensmidt, L. Sterpone, L. Carro, and M. Reorda, "On the optimal design of triple modular redundancy logic for SRAM-based FPGAs," *Proc. Conf. Design, Automation and Test Europe*, vol. 2, pp. 1290-1295, 2005.
- [10] B. Pratt, M. Caffrey, J. Carroll, P. Graham, K. Morgan and M. Wirthlin, "Fine-grain SEU mitigation for FPGAs using partial TMR," *IEEE Trans. Nucl. Sci.*, vol. 55, no. 4, pp. 2274-2280, Aug. 2008.
- [11] Z. Wang, L. Ding, Z. Yao, H. Guo, H. Zhou, and M. Lv, "The reliability and availability analysis of SEU mitigation techniques in SRAM-based FPGAs," *Eur. Conf. RADECS*, pp. 497-503, Sept. 2009.
- [12] P. Ostler, M. Caffrey, D. Gibelyou, P. Graham, K. Morgan, B. Pratt, H. Quinn and M. Wirthlin, "SRAM FPGA reliability analysis for harsh radiation environments," *IEEE Trans. Nucl. Sci.*, vol. 56, no. 6, pp. 3519-3526, Dec. 2009.
- [13] Virtex-6 FPGA memory resources user guide, UG363(v1.6), Xilinx Corp., 2011.
- [14] Virtex-6 FPGA configuration user guide, UG360 (v3.4), Xilinx Corp., 2011.
- [15] R. Hamming, "Error detecting and error correcting codes," *Bell System Technical Journal*, no. 2, pp. 147-160, 1950.
- [16] A. Frantz, M. Cassel, F. Kastensmidt, E. Cota and L. Carro, "Crosstalk and SEU-aware networks on chips," *IEEE Design and Test of Computers*, vol. 24, no. 4, pp. 340-350, July 2007.
- [17] N. Rollins, M. Wirthlin, M. Caffrey and P. Graham, "Evaluating TMR techniques in the presence of single event upsets," *Intl. Conf. Military and Aerospace Programmable Logic Devices*, Washington, D.C., Sept. 2003.



**Brock J. LaMeres (M'98-SM'09)** received the B.S. degree in electrical engineering from Montana State University, Bozeman in 1998, and the M.S. degree in electrical engineering from the University of Colorado, Colorado Springs in 2001, and the Ph.D. degree in electrical engineering from the University of Colorado, Boulder in 2005. He is currently an Associate Professor in the Department of Electrical and Computer Engineering at Montana State University, Bozeman where his teaching and research focus is in digital systems. Prior to coming to MSU, LaMeres worked for Agilent Technologies Inc., in Colorado Springs, CO as a hardware designer for digital test equipment



**Raymond Weber** received the B.S. degree in computer engineering in 2008 and the M.S. degree in electrical engineering in 2010 both from Montana State University, Bozeman. He is currently a Doctoral degree candidate in the Department of Electrical and Computer Engineering at Montana State University, Bozeman working on radiation tolerant computing systems.



**Justin Hogan** received the B.S. degree in electrical engineering from New Mexico State University in 2008 and the M.S. degree in electrical engineering in 2011 from Montana State University, Bozeman. He is currently a Doctoral degree candidate in the Department of Electrical and Computer Engineering at Montana State University, Bozeman working on fault tolerant, on-chip routing networks.

