Dynamic Reconfigurable Computing Architecture for Aerospace Applications

Brock J. LaMeres
406-994-5987
lameres@ece.montana.edu

Clint Gauer
406-994-6495
gauer33@gmail.com

Electrical & Computer Engineering Department, Montana State University, Bozeman, MT 59717

Abstract—This paper presents the design and prototyping of a computing architecture which dynamically reconfigures itself depending on the environment in which it resides. The system switches among three modes of operation (parallel processing, low power, and radiation tolerant) depending on an external radiation sensor and application input from the user. The system was prototyped on a Xilinx Virtex-5 FPGA to verify its feasibility when controlling a series of peripherals under the three modes of operation. This type of system is ideal for robust, real-time applications such as spacecraft control systems. [1] [2]

TABLE OF CONTENTS

1. INTRODUCTION

Triple Modulo Redundant (TMR) processing has been proven to be an effective method to mitigate Single Event Upsets (SEUs) in aerospace computing systems [1-3]. With the recent advances in the size and performance of Field Programmable Gate Arrays (FPGA), there is now a cost efficient path to implementing multiple processors in a single part while still meeting performance specifications [4,5]. The ability to dynamically reconfigure an FPGA to contain an arbitrary number of processors opens up many possibilities to tailor the operating characteristics of a processing system to meet various design objectives. This is especially of interest in aerospace applications where stringent design constraints such as increased reliability under radiation bombardment, power efficiency, and computation power are present [6,7].

In this work, we present an FPGA-based processing system that is dynamically reconfigured during operation to achieve different performance objectives. An FPGA part is selected which has enough logic resources to accommodate three soft processing cores plus additional glue logic. The system is designed to control a basic set of peripherals (keyboard, mouse, and LCD) as an initial proof of concept. The system is designed to seamlessly control the peripherals under each of the following three modes of operation:

Mode 1: Parallel Processing: The three processors are configured to perform non-redundant, parallel processing. This mode is used when the system is *not* in a harsh radiation environment and processing power is the primary design constraint. Processing tasks are assigned to a dedicated processor in order achieve the highest system performance.

Mode 2: Low Power: A single processor is implemented in the FPGA to reduce power consumption in environments where radiation is *not* present and power reduction is the primary design constraint. All processing is performed in a sequential manner just as in a traditional uni-processor computer system.

Mode 3: Radiation Tolerant: Three processors are configured in a triple modulo redundant configuration to perform error checking and correction using a voting and correction system. This configuration is used for the mitigation of single-event upsets in harsh radiation environments.

This approach assumes that an external radiation detector exists that can notify the FPGA-based processing system when radiation is present. When the system resides in an environment in which cosmic radiation is present, the system automatically reconfigures into *Radiation Tolerant Mode*. This mode of operation always has the highest priority. When the system is *not* in radiation, the system configures itself into either *Parallel Processing Mode* or *Low Power Mode* depending on user input. System-in-Package (SiP) technology has enabled the integration of low cost radiation sensors within the same package as traditional CMOS substrates to notify the computing system when radiation is present [8,9]. A number of sensing techniques have been demonstrated that can be used to monitor for radiation in VLSI systems including RADFETs [10], Total

1

[1] 978-1-4244-2622-5/09/$25.00 ©2009 IEEE.
[2] IEEEAC paper #1071, Version 1, Updated Dec 16, 2008
This research was carried out at Montana State University, Bozeman with funding from the Montana Space Grant Consortium (MSFC1-08-SD)

Ionizing Dose (TID) monitors [11], and built-in current sensors (BICS) [12,13]. This work assumes that a radiation sensor is incorporated within the same package as the FPGA to notify the processors when to reconfigure into *Radiation Tolerant Mode*.

The system is designed to continuously service the peripherals when switching between modes of operation and also when recovering from a faulted processor in radiation tolerant mode. This work presents the design and prototyping of our system on a Xilinx, Virtex-5 FPGA using the *PicoBlaze* soft processing core. This hardware has a well established development environment which enables the adoption of our system without any proprietary software. Our paper will describe the design and implementation of the system including the performance metrics in each of the three operating modes. We also present the implications of reconfiguration time on overall system performance. This work represents a new research effort that is being initiated at Montana State University, Bozeman on the design of radiation tolerant computing systems for aerospace applications based on reprogrammable fabrics. This project lays the initial foundation for more advanced research into the area of reconfigurable computing.

2. SYSTEM DESIGN

Our system is designed to control a set of basic peripherals as a proof of concept for operation in a real-time application. The peripherals are a PS2 Keyboard, PS2 Mouse, and a Liquid Crystal Display (LCD). The computing system continually monitors the input keyboard and mouse and writes the inputs to the LCD. The system hardware was implemented on a Xilinx ML507 evaluation board with a Virtex-5 FPGA. The system switches between modes of operation by reconfiguring the FPGA with different hardware using the *Xilinx SystemACE* and a 256Mb compact FLASH card which holds the different FPGA designs. An off-chip EEPROM is used to hold the program variable information during a reconfiguration event. Figure 1 shows the prototype system.

Fig. 1. Prototype system demonstrating the control of basic peripherals (keyboard, mouse, LCD) under different modes of operation using a Xilinx Virtex-5 FPGA.

Mode 1: Parallel Processing

The first mode of operation for this system is *parallel processing.* In this mode, the FPGA is configured to assign the various tasks of the application to dedicated microprocessors. Three PicoBlaze soft processors are used in this mode. One processor is dedicated to controlling the mouse, one is dedicated to controlling the keyboard, and one is used for writing to the LCD. The mouse and keyboard processors use interrupts to notify the LCD processor that new information has been received and needs to be displayed. The variable information for this application is stored to an off-chip EEPROM when a reconfiguration event is scheduled. The software in each of the three PicoBlaze processors is designed to read in this variable information upon start-up so that the system can continue operation in same state as a previous design upon reconfiguration. Figure 2 shows the block diagram of the parallel processing hardware configuration.

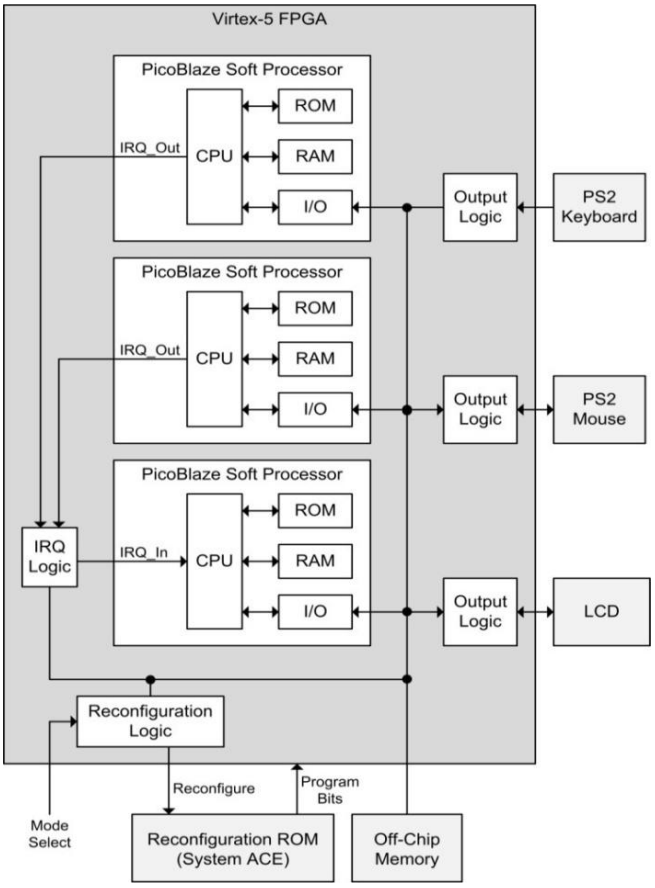

Fig. 3. Block diagram of parallel processing mode.

Xilinx ChipScope was used to monitor the internal signals on the FPGA to verify the operation of the system. Figure 3 shows a Chipscope screenshot of the address and instruction busses within the processors while in parallel processing mode. This screenshot shows the bus states at the moment when the processor controlling the mouse interrupts the processor controlling the LCD to indicate that new data has been received and needs to be written. The reset vector for the LCD control processor is 0x3FF.

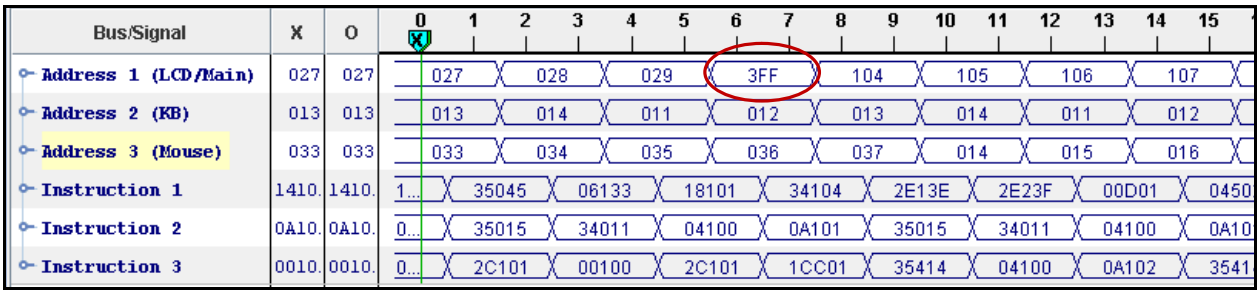

Fig. 3. Waveforms showing the parallel processing mode of operation. Address 0x03FF indicates that the processor controlling the mouse (3) has initiated an interrupt on the processor controlling the LCD (1).

Mode 2: Low Power

The second mode of operation for this system is *low power*. In this mode, the FPGA is configured to control all of the peripherals using a single processor. All of the peripheral control is implemented using a sequential program flow. This mode reduces power by using a single processor but sacrifices the speed advantages of having multiple dedicated processors. The block diagram for this mode is shown in figure 4.

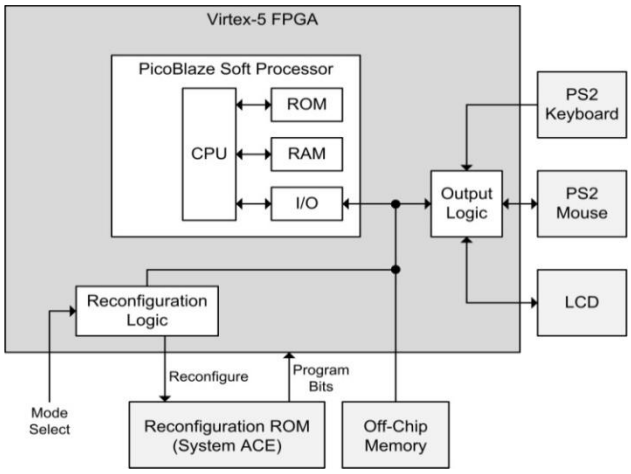

Fig. 4. Block diagram for low power mode.

Mode 3: Radiation Tolerance

The third mode of operation for this system is *radiation tolerant.* In this mode, the FPGA is configured to use three redundant processors and a TMR voting circuit to monitor for failures. For our initial prototype system, we performed TMR voting on the ROM interface within each of the three processors in addition to the I/O lines. The TMR circuitry also contains the logic for recovering from a fault. In this configuration, each of the three processors are executing the same program code that is duplicated in each of the three processors' memory. In this proof of concept, redundancy is only implemented on the processors. The TMR voting and recovery circuitry is not redundant and is implemented on the same FPGA as the three processors. This block diagram for this mode is shown in figure 5.

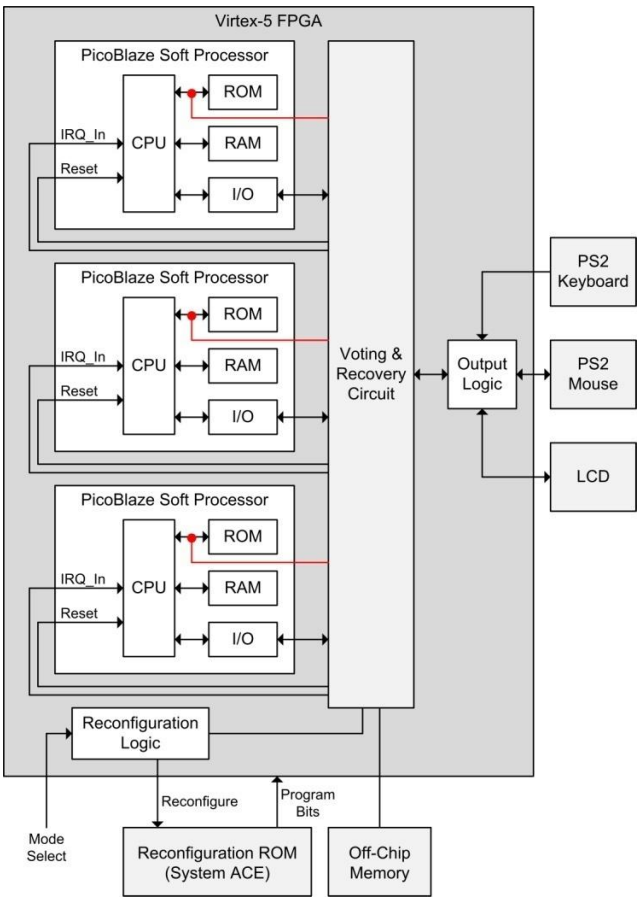

Fig. 5. Block diagram for radiation tolerant mode.

Reconfiguration

Each of the three configurations are self contained FPGA designs that are stored in the Xilinx SystemACE ROM component on the system board. Each of the three configurations contain the necessary logic to communicate to the SystemACE in order to prepare for a reconfiguration. A reconfiguration is triggered by one of two events. The first event is the detection of radiation by the integrated sensor. Note that this detection is not of an SEU itself, rather that radiation is present and the system needs to reconfigure to fault tolerant mode to prepare for the potential for SEUs. The second event is a change in

the user input indicating that either low-power or increased computation is the primary operating objective. The radiation sensor and user input are read by the FPGA as *Mode Select* lines. Anytime there is a change in the *Mode Select* lines, the system will finish its current task and then it will write any necessary variable information to an external EEPROM. The designs will then communicate with the SystemACE device to indicate which new configuration is to be used. Finally, the design will send the *Start Programming* signal to the SystemACE to begin the FPGA reconfiguration. Figure 6 shows the algorithmic flow chart of the reconfiguration procedure. This procedure is used in all three of the FPGA designs.

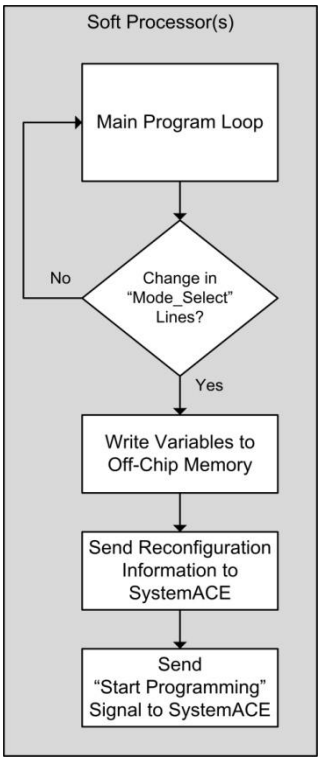

Fig. 6. Reconfiguration algorithm for all modes.

3. FAULT RECOVERY

In the *radiation tolerant* mode, the TMR circuit monitors the memory interface and I/O lines for faults. The TMR circuit also contains a state machine for the recovery of a faulted processor. When a fault is detected on one of the processors, the system attempts to recover the bad processor using the procedure flow-charted in figure 7.

Upon reset, each processor will read in its initial register values from the TMR/recovery system. All processors are reset at the same time in order to ensure they are synchronized and initialized to the beginning of the main program loop.

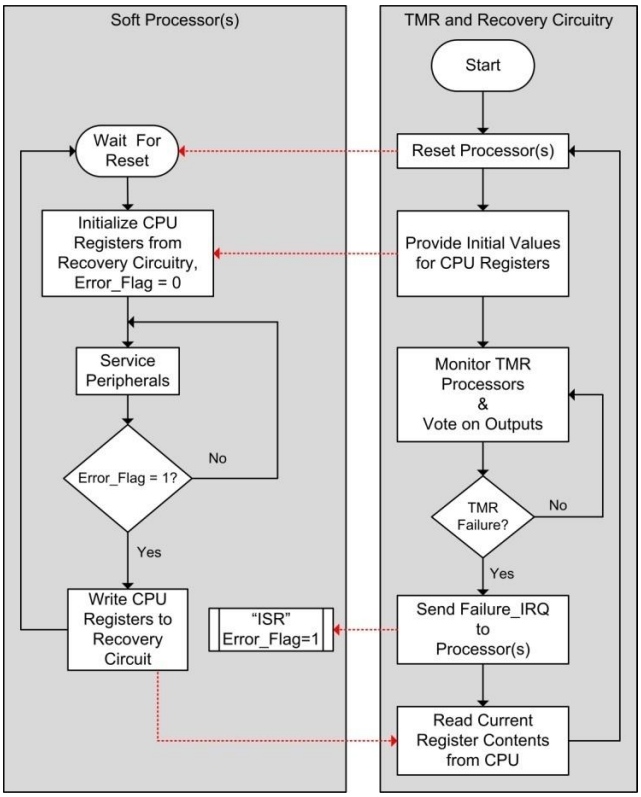

Fig. 7. Fault recovery process during radiation tolerant mode.

The processors then enter their main program loop which services the peripherals. At the end of the main program loop, an *Error_Flag* is checked to see if a TMR failure has been detected by the recovery circuit. If no failure has been indicated, the computer continues to execute its main program loop.

In the event that a failure is detected by the TMR circuit, the recovery state machine sends an interrupt to all processors. An interrupt service routine sets the *Error_Flag* indicating that a fault has been detected and processor recovery is necessary. When the main program loop checks the *Error_Flag* and sees it is asserted, it will then proceed to write all of its register and variable information to the off-chip memory and then wait for a reset. In this manner, the processors will complete their current peripheral tasks prior to beginning the recovery sequence. The TMR vote ensures that only valid information is written to the off-chip memory. The recovery state machine then resets all processors and each processor reinitializes to a known state.

Figure 8 shows a *ChipScope* screenshot of the three processors being reset during a recover cycle. This event occurred due to a fault detected on processor 1. Figure 9 shows operation during a continuous fault on processor 1 as would occur if physical hardware damage occured. The system performs its recovery sequence each time through the main program loop. The system will continue to operate with lower performance.

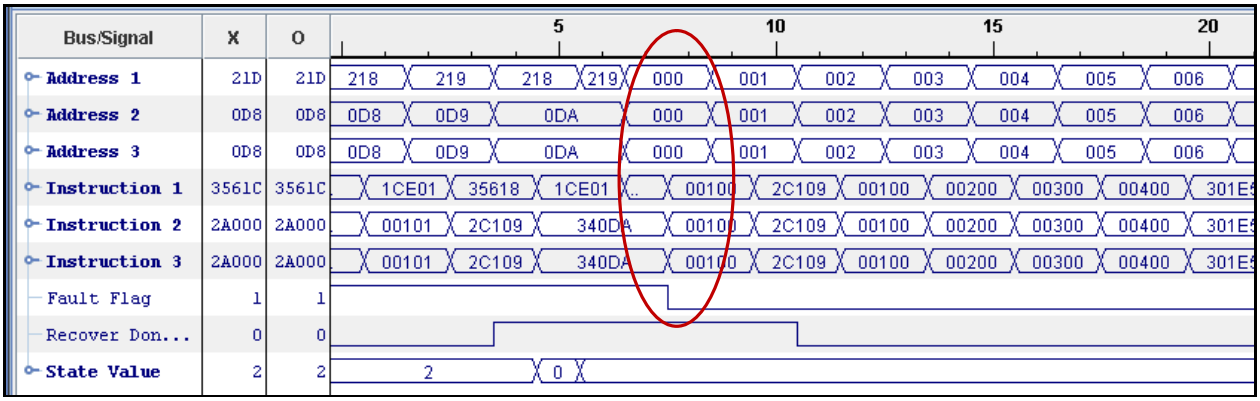

Fig. 8. *ChipScope* view of the processor busses during a reset event. This event is caused by a detected fault on processor #1 which triggers the recovery sequence.

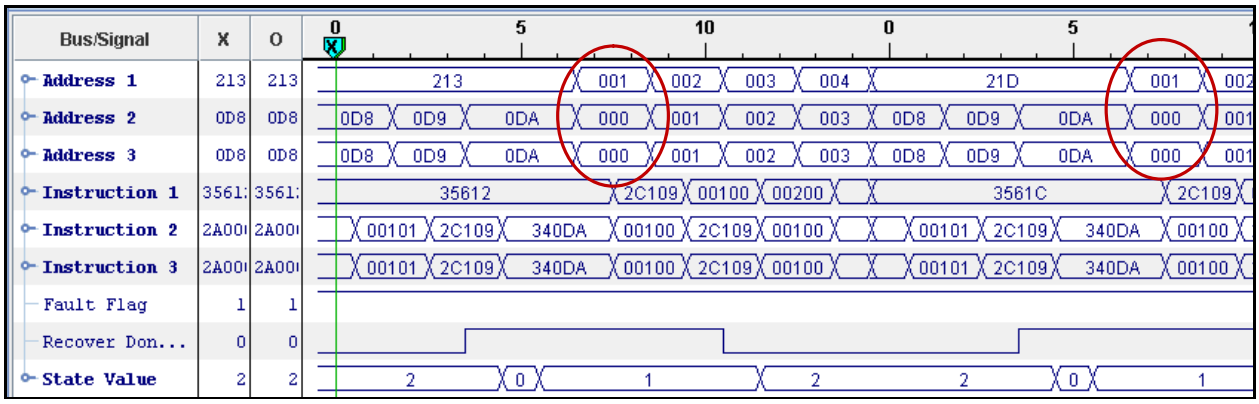

Fig 9. *ChipScope* view of the processor busses during a continuous fault on processor 1. The recovery sequence attempts to recover processor #1 each time through the main program loop. The peripherals are served each time through the program loop to ensure continuous system operation.

4. System Performance

One of the key timing parameters for our system is the time it takes for the recovery sequence to complete. (illustrated in figure 7). It was determined that the recovery of a faulted processor took 23.2ms on a Virtex-5 FPGA with a system clock of 100MHz. This time will vary for other designs depending on the amount of variable information that is stored off-chip during each recovery.

The time it takes to write to and read from the off-chip memory via IIC protocol is given as

$$t_{OCMW} = \frac{(n+2)\times 9}{f_{SCL}} + t_{WR} \times \left(\left\lceil \frac{n}{N} \right\rceil - 1 \right)$$

$$t_{OCMR} = \frac{(n+3)\times 9}{f_{SCL}}$$

where n is the number of bytes to read or write, f_{SCL} is the IIC clock rate, t_{WR} is the page write time, and N is the page size. For this design, f_{SCL} is 400 kHz, t_{WR} is 5ms and N is 16. During a fault recovery, t_{SM} is the amount of time it takes the recovery state machine to execute.

Another important timing parameter is the amount of time it takes to reconfigure the FPGA. This value is given as

$$t_{sysace} = \frac{k}{f_{SACLK}} + t_{RC}$$

where k is the number of MCU register writes to the system ace, f_{SACLK} is the clock frequency of the system ace, and t_{RC} is the time it takes the system ace to reconfigure the FPGA. For this design k is 4, f_{SACLK} is 33MHz, and t_{RC} is 504ms. This time is the same regardless of which configuration is being loaded due to the FPGA needing all logic to be configured each time it is programmed. For this work, we did not investigate partial reconfiguration. Table I shows the timing impact of the design and Table II shows the area impact of each of the three configurations used in this work.

TABLE I
TIMING IMPACT OF IMPORTANT SYSTEM EVENTS

System Event	Timing Dependencies	Total Time
Initial Power On	$t_{mouse} + t_{LCD} + t_{OCMW} + t_{sysace} + t_{OCMR}$	0.7ms + 1.67ms + 21.6ms + 504.0ms + 1.6ms = **529.6 ms**
Mode Config Change	$t_{OCMW} + t_{sysace} + t_{OCMR}$	21.6ms + 504.0ms + 1.6ms = **527.2ms**
Fault Recovery	$t_{OCMW} + t_{SMf} + t_{OCMR}$	21.6ms + 30.0ns+ 1.6ms = **23.2ms**

TABLE II
AREA USAGE FOR THREE CONFIGURATIONS

FPGA Resources	Resources Used		
	Parallel Processing	Low Power	Radiation Tolerant
Number of Slice Registers	261	102	355
Number of Slice LUTs	470	194	562
Number of occupied Slices	187	96	280
Number of LUT Flip Flop pairs	498	210	635

5. FUTURE WORK

This work was an initial effort at Montana State University to develop a knowledge base on reconfigurable computing for mitigating faults from space radiation. Future work will include more computationally intense aerospace algorithms, partial FPGA reconfiguration, scrubbing, and investigation into optimal TMR monitoring for a multi-processor system [1].

6. CONCLUSION

This paper presented the design and prototyping of a reconfigurable computing platform that switches between a parallel processing, low power, or radiation tolerant mode depending on its current environment and user input. The system is based on the existence of a radiation sensor which indicates whether radiation strikes are present. If radiation is detected, the system configures into a TMR architecture with circuitry to recover faulted processors. When not in a radiation environment, the system configures between a high performance, parallel processing mode or a single-processor, low power mode depending on user input regarding the application. The system was prototyped on a Xilinx Virtex-5 FPGA platform using the PicoBlaze soft processing core. Timing and area information on the design was presented to provide feasibility of this system for consideration in aerospace applications.

REFERENCES

[1] Melanie Berg, "Design for Radiation Effects", Military / Aerospace Programmable Logic Devices (MAPLD), Conference, Annapolis, MD, 2008.

[2] C. Carmichael, B. Bridgford, G. Swift, M. Napier, "A Triple Module Redundancy Scheme for SEU Mitigation of Static Latch-Based FPGAs", Military / Aerospace Programmable Logic Devices (MAPLD) Conference, 2004.

[3] J.W. Gambles, G.K. Maki, "Rad-Tolerant Flight VLSI From Commercial Foundries", *39th IEEE Midwest Symposium on Circuits and Systems*, vol 3, pp. 1227-1230, August 18-21, 1996.

[4] M. Stetller, "Radiation Effects and Mitigation Strategies for modern FPGAs", 10th annual workshop for LHC and Future experiments, Los Alamos National Laboratory, USA, 2004.

[5] J. Greco, G. Cieslewski, A. Jacobs, I.A. Troxel, A.D. George, "Hardware/software interface for high-performance space computing with FPGA coprocessors", 2006 IEEE *Aerospace Conference*, pp 10, March 4-11, 2006.

[6] M. Johnson, "Radiation Hardened, High Performance, Power Efficient Processing – An Objective of the NASA Exploration Systems Technology Development Program", *13th NASA Symposium on VLSI Design*, Post Falls, ID, June 5-6, 2007.

[7] S. Franklin, K. Jentung, B. Spence, M. McEachen, S. White, J. Samson, R. Some, J. Zsoldos, "The Space Technology 8 Mission", 2006 IEEE *Aerospace Conference*, pp 16, March 4-11, 2006.

[8] R. M. Lea, I. P. Jalowiecki, D. K. Boughton, J. S. Yamaguchi, A. A. Pepe, V. H. Ozguz, and J. C. Carson, "A 3-D stacked chip packaging solution for miniaturized massively parallel processing," *IEEE Trans. Adv. Packag.*, vol. 22, no. 3, pp. 424–432, Aug. 1999.

[9] M. Karnezos, "3D packaging: Where all technologies come together," in *Proc. IEEE/CPMT/SEMI 29th Int. Electron. Manufact. Technol. Symp.*, pp. 64–67, July 2004.

[10] A. Holmes Siedle and L. Adams, "RADFETs: A Review of the Use of Metal-Oxide-Silicon Devices as Integrating Dosimeters" Radiation Physics and Chemistry, 28, (2) 235 – 244, 1986.

[11] The Ridgetop Group Inc, *Datasheet: Sentinel Silicon Library, RadCell V_T Prognostic Cell* [Online], Available: www.ridgetop-group.com

[12] B. Gill, M. Nicolaidis, F. Wol, C. Papachristou, and S. Garverick, An efficient bics design for seus detection and correction in semiconductor memories," in Proceedings, Design, Automation and Test in Europe, pp. 592-597, March 2005.

[13] T. Calin, F. L. Vargas, and M. Nicolaids. Upset-tolerant cmos sram using current monitoring: Prototype and test experiments. *International Test Conference*, pages 45–53, 1995.

BIOGRAPHY

Brock J. LaMeres *(M'98) 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 Assistant Professor in the Department of Electrical and Computer Engineering at Montana State University (MSU), Bozeman. LaMeres conducts research in the area of radiation tolerant digital systems which is supported by the Montana Space Grant Consortium and NASA.*

Clint Gauer *is currently an M.S. candidate in the electrical engineering department at Montana State University, Bozeman. He works as a Research Assistant where his focus is on reconfigurable computing architectures for mitigating system level faults due to space radiation. Gauer received his B.S. in Computer Engineering from Montana State University, Bozeman in 2007.*