An FPGA-based Radiation Tolerant SmallSat Computer System

Connor R. Julien
Montana State University
540 Cobleigh Hall
Bozeman, MT 59717
303-552-6813
connor.julien@msu.montana.edu

Brock J. LaMeres
Montana State University
533 Cobleigh Hall
Bozeman, MT 59717
406-994-5987
lameres@ece.montana.edu

Raymond J. Weber
Flat Earth Inc.
985 Technology Blvd
Bozeman, MT 59718
406-994-5989
raymond.weber@msu.montana.edu

Abstract— **This paper describes the design and testing of a radiation tolerant, low-cost computer system for use in small satellites. The computer is implemented on a modern Field Programmable Gate Array, which enables the novel fault mitigation strategy to be deployed on a commercial part, thus reducing the cost of the system. Using a modern processing node also provides increased computational performance and power efficiency. Our computer system has been in development at Montana State University for the past 8 years and has undergone a series of technology demonstrations to increase its technical readiness level. These include high energy particle bombardment at the Texas A&M Radiation Effects Facility, 8 high altitude balloon flights to 30km, and two sounding rocket flights to altitudes greater than 120km. This computer is scheduled for a demonstration onboard the International Space Station at the end of 2016 and for two stand-alone small satellite missions in low earth orbit in 2017. This paper will describe the design, testing, and characterization of our computer system in addition to the qualifications underway to prepare for the upcoming orbital demonstrations.**

TABLE OF CONTENTS

1. INTRODUCTION

A radiation tolerant, field programmable gate array (FPGA) based computing system has been under development at Montana State University (MSU) that will meet the future needs of space computing. Radiation effects in space are detrimental to computers used in essential space applications. Shielding and radiation hardened processors provide some relief from this design issue but are expensive and can be mission specific. Cost, reliability and scalability drive the research performed by MSU where FPGAs provide a cheaper, low power option for processing.

Modern processing nodes (less than 45nm) have been shown to have inherent resilience to total ionizing dose (TID) effects, with commercial parts achieving greater than 500krad of tolerance simply due to the thin gate oxides and relatively deep isolation trenches found in CMOS devices. However, the use of modern processing nodes makes commercial parts uniquely susceptible to single event effects (SEEs) due to high energy ionizing radiation. As a result, a computer system wishing to exploit the power efficiency, low cost, and TID immunity of a modern FPGA must have an SEE mitigation strategy at its core.

The SEE fault mitigation approach in our computer involves breaking a commercial FPGA fabric into redundant tiles, each with the characteristics that they can fully contain the circuit of interest and also be individually reprogrammed using partial reconfiguration. In our computer, a tile contains a full computer system based on a Xilinx MicroBlaze soft processor. At any given time, three of the tiles run in triple modular redundancy (TMR) with the rest of the tiles reserved as spares. The TMR voter can detect faults in the active triad by voting on the tile outputs. A configuration memory scrubber continually runs in the background and can detect faults in the configuration memory of both the active and inactive tiles. In the event of a fault in the active triad, the damaged tile is replaced with a known good spare and foreground TMR operation continues. The damaged tile is repaired in the background by reinitializing its configuration memory through partial reconfiguration (PR). Faults detected in inactive tiles by the scrubber are also repaired in the background and reintroduced as spares. This approach mitigates SEEs in the FPGA circuit fabric in addition to SEEs in the configuration memory.

Prototyping of this radiation tolerant computer system began in 2007. Since then the design has been tested in front of a cyclotron and has flown on multiple high altitude balloon and sounding rocket flights. To continue progressing the maturity of the system, testing aboard the International Space Station (ISS) will be performed in late 2016. The specific version of the computer system that will be tested aboard the ISS is called Artemis. NanoRacks LLC, a partner of NASA, provides the integration and test environment aboard the ISS. Two other versions of the radiation tolerant system will be

978-1-5090-1613-6/17/$31.00 ©2017 IEEE

incorporated as the experimental payload on two 3U CubeSats to be launched from the ISS in 2017.

The radiation tolerant computing system is comprised of a stack of printed circuit boards (PCBs) that fit a 1U CubeSat form factor. Each PCB contains a major sub-system of the computer including a power regulation system, processing systems, data logging system and an interface system. To meet requirements for upcoming missions, the system can fit into the required chassis for testing on the ISS or, when paired with an avionics system, a satellite mission.

The Artemis flight unit has been delivered to NanoRacks for its mission to the ISS. At the time of this paper, Artemis is set to launch to the ISS on HTV-6 out of Japan in December of 2016. HTV-6 is operated by the Japanese Aerospace Exploration Agency (JAXA) and will launch from Tanegashima Space Center in Japan to resupply the ISS. Data from this mission should be available in early 2017.

This paper communicates the specific motivations for this project including the need for fault tolerant space computing as well as describing radiation effects on space computers. Prior work in the field of radiation tolerant computing as well as MSU's past work is explained. The system design for Artemis is fully described as well as the details for Artemis's mission with NanoRacks to the ISS. Characterization and test results of Artemis are discussed. Future planned demonstration missions for the radiation tolerant computer are also described. For the duration of this document, the name Artemis is used to describe the specific version of the radiation tolerant computer system that will be tested in low earth orbit (LEO) on the ISS.

2. MOTIVATION

Space Computing Needs

As humans continue to develop technology used in space, a large demand for radiation tolerant computing systems has arisen. Every few years, the National Aeronautics and Space Administration (NASA) releases a Technology Roadmap with the intention of guiding space technology development. The objectives of flight computing in Technology Area (TA) 11 are to develop "effective radiation hardening technologies and processing approaches for extreme environments" using part-level hardening and software based fault tolerance [1]. The biggest challenges to meeting this objective are maintaining high processing power, having low power requirements and enhancing reliability for space missions. Many of these requirements are met by using FPGAs which are power efficient and reconfigurable. MSU's approach has been to use commercial off the shelf (COTS) FPGAs to meet the needs of NASA's flight computing technology.

Radiation Effects on Space Computers

The most dangerous radiation to space computers is high energy ionizing radiation. These particles have large amounts of kinetic energy some of which can be transferred to a device as the particle strikes the substrate. If the energy transferred to the material exceeds the materials band gap energy then an electron could be excited from the valence band to the conduction band, creating an electron-hole pair [3]. Electron-hole pairs are the main source for radiation induced faults in space electronics. Total Ionizing Dose (TID) and Single Event Effects (SEE) are two types of radiation effects that can cause failures in electronic devices.

Failures caused by TID occur when the device is hit with low energy protons and electrons (greater than 30MeV/AMU) [2]. As the substrate of the device is struck by the particle, an electron-hole pair forms. Most failures caused by TID occur because the electron-hole pair gets trapped in the insulating material of the electronic device [12]. If the electron-hole pair is trapped in the gate oxide of a transistor, a conduction channel can form, causing the transistor to be in a permanently active state. Trapped charge from TID can also occur in the regions between transistor devices, causing leakage current between the devices. These TID effects may not affect the functionality of the device, but they do cause excessive power consumption and will lead to failure of the device over time [4].

Single Event Effects are produced when a high-energy particle strikes the diffusion region of a device. As the particle passes through the device, a path of electron-hole pairs are produced. Although the electrons and holes may recombine quickly, if the charge carriers effect a certain part of the circuit, unwanted logic level transitions could be induced. Unlike device damaging TID, SEEs do not cause permanent damage to the device. However, they can cause erroneous operation or even full system crashes due to unexpected logic transitions. When an SEE causes one of these logic changes it is referred to as a Single Event

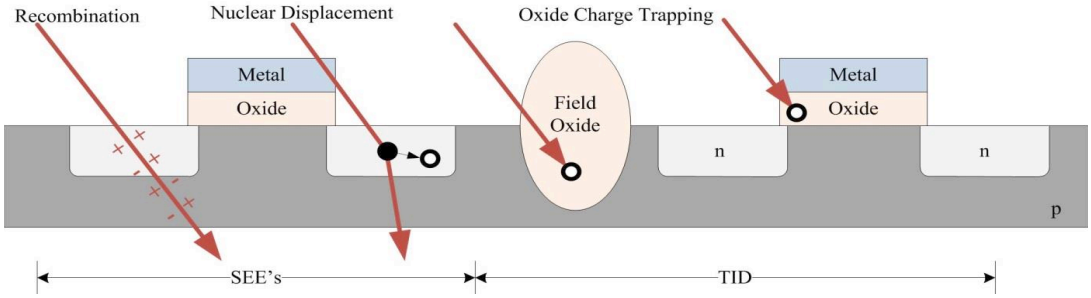


Figure 1. MOSFET cross-section showing radiation faults by SEEs and TID.

Transient (SET). When an SET occurs and the transient is latched by a flip-flop or other memory element it is called a Single Event Upset (SEU). These changes in logic state are referred to as a "bit-flip" [5]. SEUs affect both bipolar junction transistors (BJTs) and metal-oxide-semiconductor field-effect transistors (MOSFETs). If SEUs are detected and corrected by resetting the device, no permanent damage is done. However, if an SEU causes memory corruption or permeates a logic shift through multiple levels it is called a Single Event Functional Interrupt (SEFI). SEFIs can lead to more behavior problems as well as power degradation. SEFIs are also harder to recover from, requiring a full system power cycle or re-initialization. Figure 1 above shows the types of radiation faults, both TID and SEEs, on a MOSFET.

Modern day integrated circuits exhibit small feature sizes, creating less of a concern for TID damage. The oxide thickness is so small it becomes statistically improbable that a charge will get trapped. Modern FPGAs are achieving TID tolerance levels of greater than 300krad when implemented in the 65nm process node and as much as 600krad when implemented in a 22nm node [6]. The small diffusion regions in current devices cause SEE susceptibility because a radiation strike can carry enough energy to change the state of a device [7].

A Necessity for Fault Tolerant Computing

The faults caused by radiation in space paired with the desires and needs of the space community provides strong evidence that fault tolerant computing is a necessity. Fault tolerant computing research at MSU aims to provide a solution to the problems faced by technology used to perform complicated, long duration space missions.

3. Prior Work

Montana State University's Contribution

For the last eight years, MSU has been researching and developing an FPGA architecture to mitigate SEEs in space computing. From prototyping with evaluation boards to launching the system aboard sounding rockets, the technology has progressed in flight heritage and technology maturation. MSU's approach to mitigating SEEs and the history of technology maturation for the radiation tolerant system are described in the subsections below.

Montana State University's Approach—At the core of MSU's approach to radiation tolerance are modern COTS FPGAs. The small feature sizes in a 28nm process FPGA gives inherent TID immunity. FPGAs are a low cost, readily available solution over other radiation hardened processors. The reconfigurable nature of FPGAs is extremely advantageous to MSU's novel architecture. FPGAs also provide a lower power consumption option while maintaining or increasing computational power. This project utilizes FGPAs from Xilinx Inc., an American semiconductor manufacturing company. Xilinx FPGAs are frequently used for space and military applications so they carry more inherent robustness which suits the needs of space computing.

To mitigate SEE faults in the FPGA, TMR+Scrubbing is expanded, creating more reliability. MSU's methodology involves creating redundant circuitry in the fabric of the FPGA. Each redundant circuit, referred to as a tile, is contained within a specific region in the FPGA fabric, allowing that region to be reconfigured with partial reconfiguration (PR). PR is an advanced FPGA design technique. PR of an FPGA involves modification of an active FPGA design which re-programs a region of the FPGA while the rest of the fabric continues to run in normal operation [8]. Each tile contains a 32-bit RISC architecture soft processor called the MicroBlaze (Xilinx). The experimental FPGA contains nine tiles, three of which are actively running in a common TMR configuration. This leaves six spare tiles, all identical to the three that are currently running. Figure 2 below shows the floorplan of the Artix-7, each pink box representing a tile. The TMR voter is used to determine when one of the three active tiles receive a fault. The voter checks the outputs of the three active tiles and detects a fault if the outputs do not agree. When the TMR voter determines a fault

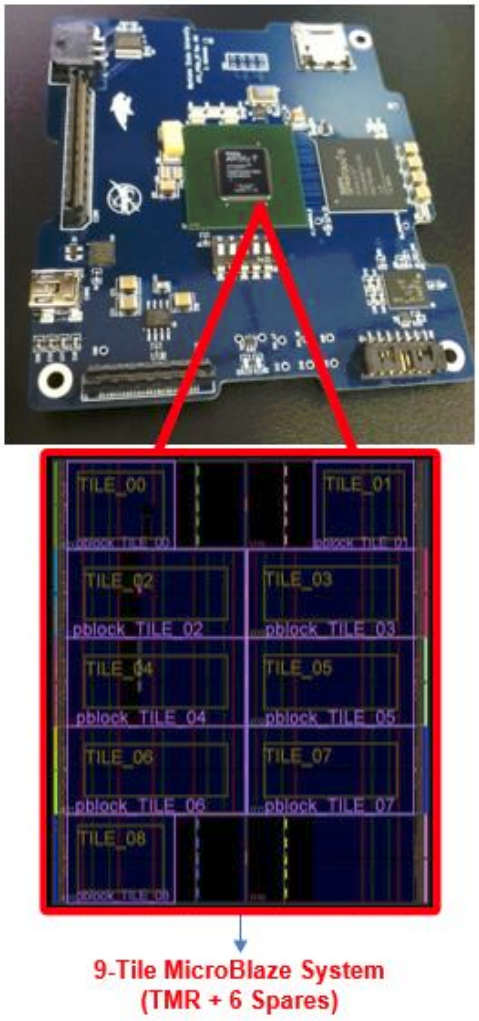


Figure 2. The FPGA board (above) and the Artix-7 Fabric Layout (below).

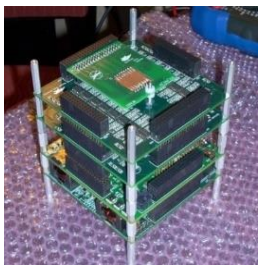
has occurred on an active tile, the faulted tile is marked and the control FPGA initializes a new tile that was originally a spare. The new tile is synced with the other two active tiles and TMR operation resumes. In the background, the control FPGA performs a PR of the faulted tile which reinitializes its configuration memory. As soon as the tile is partially reconfigured, it is listed as a new possible spare tile. A configuration memory scrubber also runs in the background and can detect faults, triggering a new active tile and causing a PR of the faulted tile.

Technology Maturation—A history of the maturation of the radiation tolerant computing system at MSU is presented in this section. NASA uses a measurement system called the Technology Readiness Level (TRL) to provide consistent comparison of maturity between different technologies. This scale system begins with TRL-1, where basic principles are reported, all the way to TRL-9, where the technology has been flight proven on successful missions.

The radiation tolerant computer started at TRL-1 in 2007. Between 2008 and 2010, it advanced to TRL-2 and TRL-3. The prototype to achieve TRL-3 involved connecting prototype boards to commercial Xilinx FPGA evaluation boards. Figure 3 shows the proof of concept prototype used to reach TRL-3.

Figure 3. Prototype Hardware used to Demonstrate TRL-3.

Two tests at the Texas A&M Radiation Effects Facility allowed the computer system to attain TRL-4. To perform these tests, the system was developed into a 4" x 4" x 4" stacked PCB cube form factor in 2010 (Figure 4). The two tests occurred between 2010 and 2011. A mounting test fixture was used to position the computer system in front of a cyclotron that bombarded the system with krypton ions at 25MeV/AMU. Testing with the cyclotron and confirming the integration of the different system components allowed for validation in a laboratory environment and TRL-4.



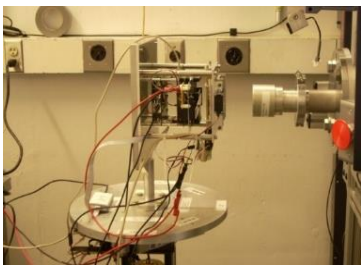


Figure 4. Prototype Hardware used for Cyclotron Testing and Demonstration of TRL-4.

TRL-5 was demonstrated between 2011 and 2013 on multiple high altitude balloon flights. To fly on these balloons, certain form factors had to be adopted and housing for the computer system needed to be developed. A power board was implemented to regulate power from battery packs into the system. A data logging feature was also added. Six of the eight balloon flights were performed by the BOREALIS program run by the Montana Space Grant Consortium (MSGC). These balloon flights generally flew to 90,000 feet out of southwest Montana. The other two flights were performed by NASA's Columbia Scientific Balloon Facility. These flights attained altitudes of 120,000 feet and took place in New Mexico. These flights showed that the system can survive the harsh environment of space, helping it achieve TRL-5.

Figure 5. View from High Altitude Balloon Flight to Demonstrate TRL-5.

In 2014, the computer system was flown on a sounding rocket to achieve TRL-6. The rocket vehicle was SL-9 operated by UP Aerospace LLC which attained an altitude of 408,000 feet (Figure 6). To prepare for this flight, the computer system was again redesigned to fit a 1U CubeSat form factor. This redesign readied the system for sounding rocket flights as well as future ISS and satellite missions. Attaining TRL-6 means that the system and subsystems were validated in an end-to-end environment. TRL-7 was attempted to be validated, however a few hardware malfunctions prevented full system operation.

Figure 6. SL-9 Integration and Launch to Demonstrate TRL-6.

In March of 2015 the system was flown on another sounding rocket out of Wallops, Virginia. This rocket flight required much more rigorous testing before launch. Multiple vibration and pressure tests were performed after the integration phase. A much more solid deck platform was used to interface the radiation tolerant computer to the sounding rocket (Figure 7). This flight did not advance the TRL of the computer system due to failures in the data logging system during vibration testing and in the battery system during launch.

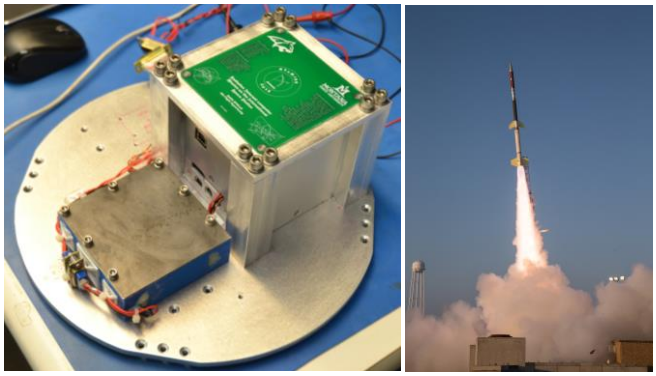


Figure 7. Hardware and Terrier-Orion Rocket Launch.

A Necessity for Space Orbit Testing

Now that TRL-6 has been achieved, the system must be exposed to more rigorous environments. This system is designed to operate in the harsh radiation of space, which is impossible to simulate on Earth. Testing with a cyclotron does not reproduce the space environment accurately. Even with the aluminum lid of the FPGA removed, the FPGA may not see enough energy to witness an SEE. High altitude balloon flights expose the system to harsher environments that are unachievable during ground testing. However, the short duration of these flights as well as the lower altitude still does not provide the system with the correct radiation environment. Although sounding rocket altitudes can reach a high radiation environment, these rocket flights typically only last a duration of fifteen to twenty minutes, even less than the balloon flights.

Due to these restrictions and in order to progress the TRL of this radiation tolerant computer system, it must be fully tested in LEO for a long duration of time. The most obvious platform choice is to test the system integrated into a CubeSat that will orbit the Earth for months to years. Another opportunity would be to test the system on the ISS, where the radiation environment is adequate to validate the operation of the system.

4. NanoRacks Mission Overview

To achieve the space orbit required to attain a certain fault rate, Artemis will be tested on the ISS. This test is made possible by NanoRacks LLC, which operates an internal payload system on the ISS called the NanoRacks Platform [9]. This section covers the mission overview along with the technical requirements of a NanoRacks internal payload (also called a NanoLab).

On board the ISS, a rack locker on the Japanese Experiment Module (JEM) contains two NanoRacks Platforms (Figure 8). The purpose of these platforms is to interface the NanoLab modules both structurally and electrically to the ISS. As an experiment is built into a NanoLab form factor, the NanoRacks Safety Data Template is submitted and approved. A fit check and function test are performed during delivery. Then the NanoLab is launched to the ISS on a resupply mission by SpaceX, Orbital ATK or the Japanese Aerospace Exploration Agency (JAXA). Once the cargo vehicle has docked with the ISS, an astronaut will move the NanoLab to the NanoRacks Platform and plug it into a USB port. The NanoLab will be powered and begin operations, showing up as a USB mass storage device to the NanoRacks Platform. During operation, data can be retrieved from a NanoRacks ground station in Houston, Texas using Remote Desktop Connection to access the USB hub on the NanoRacks Platform aboard the ISS. NanoRacks simply emails the data files to the customer once they are downloaded from the ISS. The NanoRacks ground station can download files from the ISS at a rate of 3 MB/s [9-10]. When the mission is complete, an astronaut removes the NanoLab and stores it for its return trip to Earth. Once the vehicle has landed, NanoRacks returns the NanoLab to the customer.

As described in the next section, Artemis was developed into a 1U NanoLab for its mission to the ISS. NanoRacks upholds several functional requirements for interfacing the NanoLab to the NanoRacks Platform on the ISS. For the 1U form factor, the Artemis NanoLab could not exceed 100mm x 100mm x 100mm in size [9]. A variety of companies supply CubeSat chassis that meet the specifications of a 1U CubeSat, including NanoRacks and Pumpkin. The maximum mass of

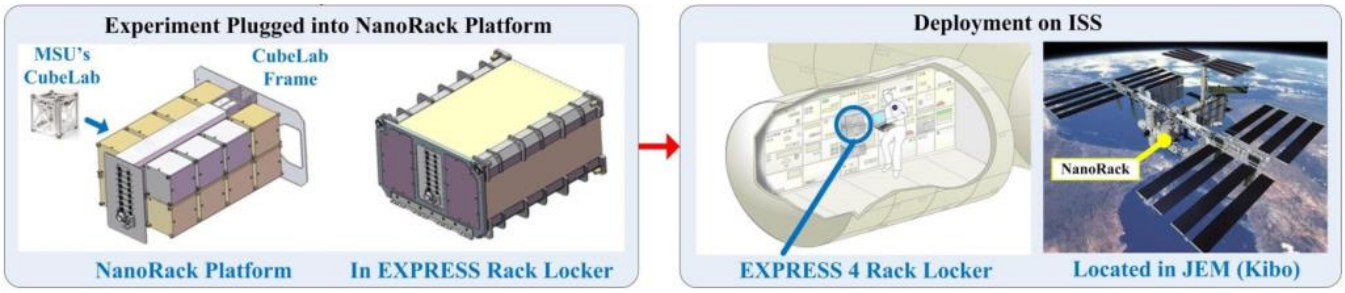


Figure 8. NanoRacks Mission Concept.

the Artemis NanoLab could not exceed 1,000 grams. The USB connection from the Platform to the NanoLab provides 2W of power at 5VDC. Special care must be taken to ensure the USB port spacing, dimensions and orientation of the NanoLab matches the NanoRacks Platform USB port. Additional requirements involve following safety standards like ensuring the NanoLab does not contain any pyrotechnics and that the outer walls of the NanoLab chassis pass a sharp edge test [9-10].

5. System Design

As described in the Prior Work section, this radiation tolerant computing system has undergone numerous iterations. After confirming the breadboard design was functioning as expected, the system was implemented into a stack of printed circuit boards (PCBs). Each board contained a specific subsystem, described in the Hardware subsection below. To prepare the computer system for the ISS, each PCB had to be modified to fit the form factor required by NanoRacks as described in the previous section. The following subsections describe each PCB, the FPGA digital design for both FPGAs and the software developed for Artemis.

Hardware

The Space Science and Engineering Lab (SSEL) at Montana State University has flight heritage using the CubeSat chassis from Pumpkin that fits the form factor required by NanoRacks. The EE program at MSU works closely with the SSEL and therefore decided to utilize their expertise by using the Pumpkin CubeSat chassis for the ISS mission. Each PCB was modified to fit inside the Pumpkin chassis, following SSEL's PCB board outline. This allowed for the stacked computer system to reside completely inside the chassis, isolating it from other environments. The following subsections describe the design of each of the three PCBs used in Artemis.

ISS Interface Board—The bottom board of the PCB stack is an unpopulated version of the data interface board (Figure 9). The only purpose of this board is to provide a USB connector in the mechanically correct place for Artemis to interface with the NanoRacks Platform on the ISS. Two wires from the USB port break out VCC (+5V) and GND to the power board. All four wires from the USB port (VCC, GND, D+ and D-) are also broken out through wires to the data interface board on the top of the PCB stack.

Power Board—The power board (Figure 10) has been under development at MSU for the last six years. Originally the board could receive a wide range of input voltages, typically 9V-30V [11]. When the power board was revised for the Artemis mission, the high voltage input capability was removed in order to save space on the PCB. It also was not necessary to include a high range of voltage input because the ISS supplies a steady 5V to Artemis during its mission. Therefore, the input voltage for the Artemis power board was reduced to 4-13 Volts [12]. The power board uses linear regulators along with two Texas Instrument (TI) power

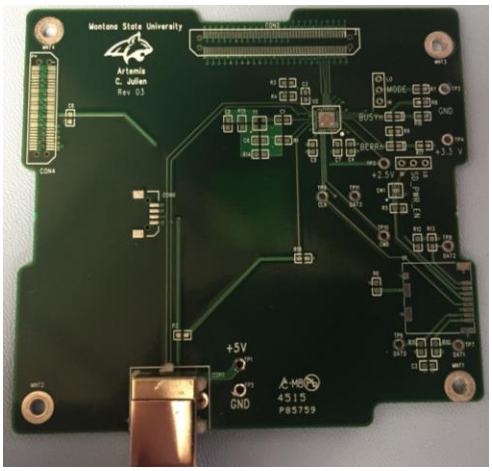


Figure 9. ISS Interface Board.

controllers to produce and monitor eleven voltage rails used by the rest of the Artemis system.

The power board produces six digital voltage rails and five analog voltage rails. The TI power controllers require a 3.3V supply voltage for operation [13]. A TI LMR14203 buck voltage regulator is used to produce this voltage. The LMR14203 is specifically designed for an unregulated voltage source over a wide range of voltage inputs (4.5-42V) [14]. A TI LMR62014 step up voltage regulator is used to generate a 15V, biasing voltage rail for use with an attached experiment. This regulator was chosen for this rail because it can prevent a runaway condition on a sensor by only providing small amounts of current [15]. The remaining nine rails are generated by TPS62130 step-down converters. This regulator is used in a buck-boost topology to generate the -3V rail [16]. The TPS62130 is used as a normal buck regulator to generate the 3.3V, 3.0V, 1.8V, 1.0V and 0.95V rails along with three 2.5V rails. Table 1 below provides a summary of these voltage rails, which regulator is used to generate each rail and what system each rail is used for.

Table 1. Artemis Power Board Voltage Rails

Voltage Rail	Regulator	System
3.3V	LMR14203	Power controller supply voltage
15V	LMR62014	Bias voltage for exp.
3.3V	TPS62130	FPGA digital I/O
2.5V	TPS62130	FPGA digital I/O
1.8V	TPS62130	FPGA configuration
1.0V	TPS62130	FPGA core voltage

0.95V	TPS62130	FPGA core voltage
3.0V	TPS62130	Amplifier power
-3.0V	TPS62130	Amplifier power
2.5V	TPS62130	Front comparator
2.5V	TPS62130	Back comparator

Two TI power controllers, UCD90124A's [13], provide a variety of sequencing and monitoring features. A 12-bit ADC can be used to monitor power-supply voltage, current or temperature inputs. Another 26 GPIO lines can be used as power enables, resets or to control LEDs. Initial configuration of the controllers is performed over a provided TI USB-GPIO device cable. Once the devices are connected to a computer, the TI Fusion Digital Power software can be used to flash the device with the correct configuration. Upon power up of the power board, the configuration parameters stored in flash memory are loaded onto the devices and the power rails enabled. The power controllers can also communicate with the Spartan-6 FGPA on the FPGA board using the PMBUS serial interface [17]. PMBUS is built on the I2C specification and is similar to SMBUS. This allows the Spartan-6 to collect data from the power controllers such as voltage, current, temperature and system runtime. A red LED for each power supply is attached to GPIO on the power controllers. In the event of a faulted rail, the controller will turn on the red LED. One green LED is also attached to each power controller. These LEDs are only on when all the power rails on each respective controller are operating normally.

TI Fusion allows for complete control of power on and power off rail sequencing. The Artix-7 FPGA requires a power on sequence of V_{CCINT}, V_{CCBRAM}, V_{CCAUX} and V_{CCO} [18]. Therefore, the 0.95V, 1.8V and 3.3V power rails on the power board are brought up in that order. The remaining power rails turn on at the same time as the 3.3V rail. All the power rail outputs as well as the PMBUS signals run to the power connector to be distributed to other boards in the computer stack.

FPGA Board—The FPGA board (Figure 11) is connected above the power board and provides control of the entire Artemis system along with housing the main device under test. The Spartan-6 FPGA controls the system while the Artix-7 is the experiment FPGA (details below). Most of the other hardware on the FPGA board supports the FPGAs. An IM Flash module stores the initial bitstream configuration for the Spartan-6. The flash memory is configured through a JTAG cable during benchtop testing. An SD card contains the full and partial bitstreams for the Artix-7. A bank of five switches provides the configuration select for both FPGAs. The Spartan-6 operates in Master Serial/SPI mode while the Artix-7 operates in the Slave SelectMAP configuration. To configure properly the correct switches must be set to on or off for each FPGA. On power up, the flash memory configures the Spartan-6. A green LED is attached to the DONE pin of the Spartan which goes high upon a successful program, lighting the LED. Once the Spartan-6 is running, one of its first tasks is to program the Artix-7. The Spartan-6 pulls the full configuration bitstream from the SD card and configures the Artix-7 over the SelectMAP port. Another green LED is attached to the DONE pin of the Artix-7, which turns on after the Artix has been successfully programmed. The two FPGAs are provided a 100 MHz clock from an oscillator positioned next to the Spartan-6.

A Maxim MAX6627 digital temperature sensor is used to monitor the die temperature of the Artix-7. The Artix contains an in-chip diode connected transistor that is connected to the MAX6627 from a GPIO pin. The temperature sensor sends data to the Spartan-6 over a 3 pin SPI bus. The Spartan displays the Artix die temperature in the GUI and prints it as part of a data packet.

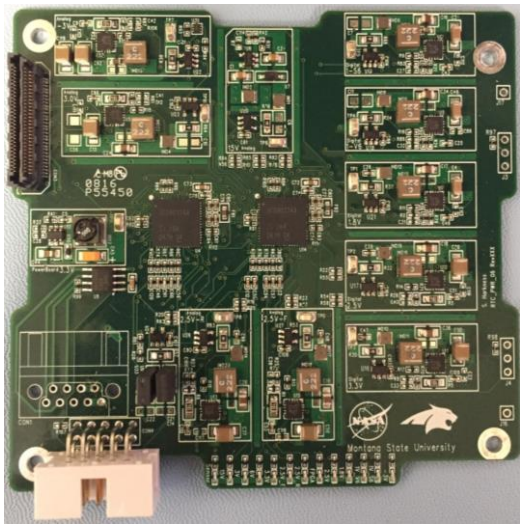


Figure 10. Power Board.

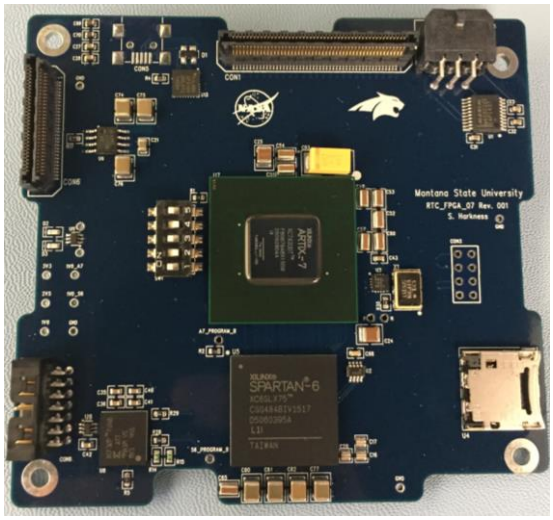


Figure 11. FPGA Board.

The remaining FPGA board features provide connections to other systems or ground support equipment. A JTAG header allows the FPGAs and flash memory to be configured over JTAG, making benchtop testing easier. There is a micro-USB connection (currently unused) as well as an RS-232 connector for the GUI. The power connector brings regulated power rails from the power board below to provide the FPGA board with its five voltage sources (see Table 1). The sensor connector provides 64 signal lines that are connected to pins on the Spartan-6 FPGA. These signals lines will be used in future developments for connections to radiation sensors or other peripherals. Eleven signal lines used by the data board also run through the sensor connector.

Data Board—The data board (Figure 12) is connected on top of the FPGA board with both the power and sensor connectors. This board provides the dual functionality of storing data on an SD card while being able to access the data as if it were a removable USB device. The MAX14502 USB-to-SD card reader produced by Maxim provides this functionality. During normal operation, the MAX14502 is in card reader mode. This allows the Artemis data SD card to be viewable as a removable storage drive (similar to a flash drive) when connected to a computer by USB. During a data file write, the MAX14502 is in pass-thru mode allowing the control FPGA to write a data file to the SD card. The Spartan-6 communicates with the SD card over a 4-pin SPI with chip selects and other control signals. While the chip is in pass-thru mode, the SD card does not show up as a USB drive. The Maxim chip is supported by a 19.2 MHz oscillator.

As previously mentioned, a power line, ground line and two signal lines are broken out from the bottom ISS interface board and connected to the top data board. The VCC line is used by the MAX14502 to show that Artemis is connected to a computer as a USB device. The two signals, D+ and D-, are used for USB communication.

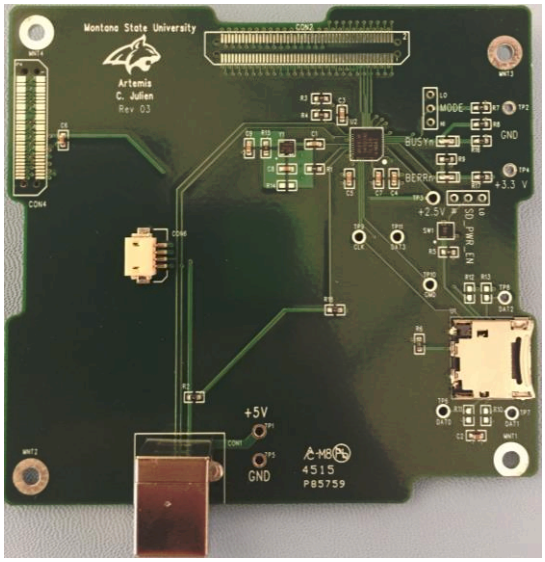


Figure 12. Data Board.

FPGA Digital Design

Both the Spartan-6 and the Artix-7 FPGAs are designed with VHDL using Xilinx design suites. These design suites synthesize and implement VHDL modules and generate a bitstream that is used to configure the FPGA.

Artix-7—The Artix-7 digital design involves part of the TMR voter, a multiplexer and the nine redundant tiles that each run a MicroBlaze soft processor (see Software, below). These systems make up the TMR+Scrubbing+Spares configuration. Care must be taken when designing the floor plan of the Artix-7 because for each tile to be partially reconfigurable it needs to adhere to clocking and other resource boundaries on the FPGA. Each box in Figure 13 is a tile that contains a MicroBlaze. Unlike Figure 2, this image shows the Artix-7 fully routed.

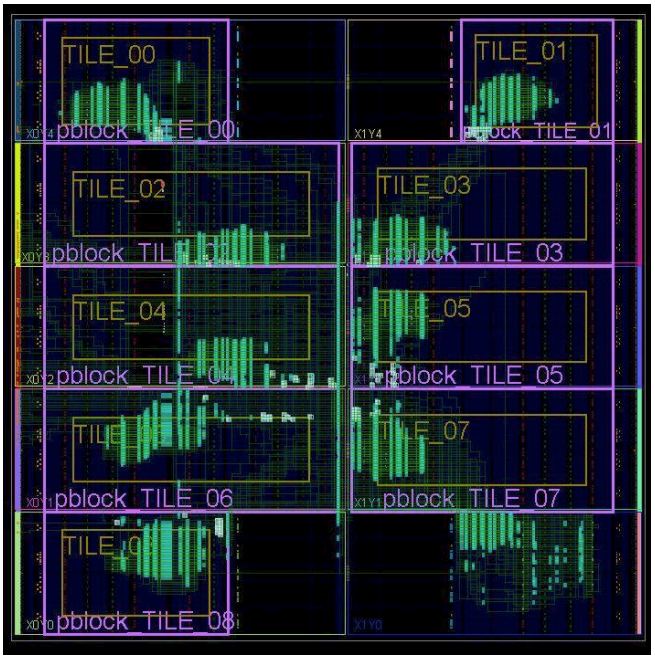


Figure 13. Routed Artix-7 Fabric Floorplan.

A Xilinx Clocking Wizard IP is used to create the different clock speeds used by the different components of the system from the 100 MHz clock speed that is an input to the FPGA from the oscillator [19]. The clock wizard breaks the 100 MHz clock into a 10 MHz clock for the TMR voter and the multiplexer. A custom clock divider breaks the 10 MHz clock into an even slower 156 kHz clock used by the each of the nine tiles.

Each of the nine tiles are simple VHDL components. The clock and two resets, an auxiliary reset and an external reset, are the only inputs. A MicroBlaze soft processor is dropped into each tile (see Software). The output is a fourteen-bit standard logic vector.

The output of each tile runs to the multiplexer component. An active tile signal from the Spartan-6 tells the MUX which three of the nine tiles are currently active. The MUX then

outputs just the output of the three active tiles. These three tile outputs are inputs to the TMR voter and are also routed to the Spartan-6 FPGA to be used in the GUI and printed in the data file.

The TMR voter module compares the three active tile outputs with each other. If one of the outputs is different from the other two, the voter recognizes that the tile has been faulted and provides a two-bit bus (Health_Tile), designating which of the three active tiles is faulted. The health tile signal is also routed to the Spartan-6 FPGA, which uses the information to determine what operation should be performed next.

Spartan-6—The digital design modules on the Spartan-6 include the SD card controller and the Artix-7 configuration state machine. The main feature of the Spartn-6 is a single MicroBlaze soft processor dropped into the fabric that runs ControlOS, discussed in the Software section below.

A Digital Clock Manager Xilinx IP Core is used to break the input 100 MHz clock into four different clocks used by various parts of the system. A 200 MHz clock is produced for use by the MicroBlaze. A 100 MHz clock is available for future implementation of a radiation sensor. A 20 MHz clock is used by the SD card controller and the Artix-7 configuration state machine. Finally, an 8 MHz clock is used by the soft error mitigation (SEM) controller core.

Software

The software that runs on the MicroBlaze soft processors on both FPGAs is written in the C language and developed with the Xilinx Software Development Kit. An executable file (.elf) is generated when the program code is compiled. This executable file is associated to its specific MicroBlaze in the digital design of each FPGA.

Artix-7 Software—Nine MicroBlaze soft processors, one on each tile, run on the Artix-7. For Artemis, a simple counter program is run on the MicroBlaze. The MicroBlaze has a fourteen-bit output. The counter starts at zero and counts to 16,383 (2^14) before rolling over. The MicroBlaze sends this count value to the MUX in the FPGA fabric.

Spartan-6 Software—The Spartan-6 has one MicroBlaze incorporated into its design that runs a custom operating system called ControlOS. ControlOS manages and runs the systems main tasks and functions. ControlOS also runs a serial GUI that can be displayed on a computer screen with a serial console.

Upon configuration of the Spartan-6 and once ControlOS starts running it performs a pre-operational self-test (POST). During the POST, the timers, interrupts and various other GPIO peripherals are initialized. Upon successful initialization of an item in the POST it turns green in the GUI (Figure 14). The last POST task is to perform the initial configuration of the Artix-7. Once the POST is passed, normal operation proceeds.

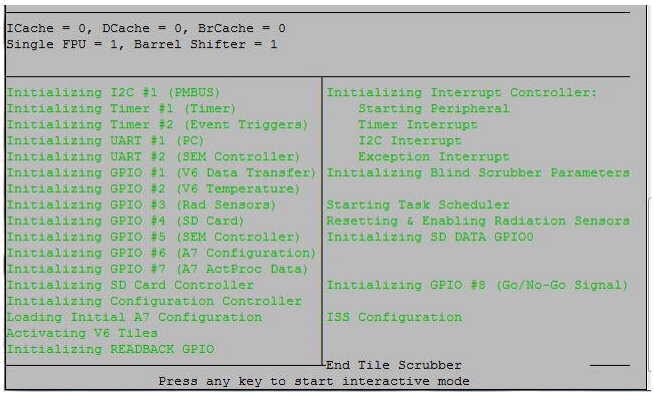


Figure 14. Artemis GUI POST Screen.

ControlOS operates as a first-in-first-out scheduling kernel. Sixteen tasks can be scheduled at once with a configurable runtime duration. Each task is set with three parameters: a pointer to the task function, a period (in seconds) and an enable/disable. This allows for full customization of the tasks that Artemis performs. Mission specific considerations include how often a data file should be written, how often a fault should be injected and how often the system performs a blind scrub. Table 2 below shows each of the main ControlOS tasks and how often they are run for the Artemis mission.

Table 2. Artemis Tasks and How Often They Are Performed.

Task	How often task is run
Move Tile	1 second
Repair Tile	1 second
Update Power Measurement	20 minutes
Update Power Logs	20 min 5 secs
Active Tiles Update	5 seconds
Write Data File	12 hours
Watchdog Update	30 minutes
Fault Injection	11 hours
Blind Scrubber	7 hours

Partial reconfiguration on the Artix-7 is completely controlled by ControlOS. In an event of a fault, the voter on the Artix alerts ControlOS on the Spartan. ControlOS immediately brings up a new tile, and then performs a PR of

the faulted tile. To perform this feature, ControlOS first determines which tile is faulted by correlating which tiles were active and which active tile was faulted. Each tile is a reconfigurable region in the fabric that has a partial bitstream used to reconfigure that particular region. Each partial bitstream has a start address and length associated with it and are stored after the initial Artix-7 full bitstream on the SD card. Once the faulted tile number has been determined, ControlOS pulls the tiles partial bitstream from the SD card using the address and length for that tile and programs that region of the Artix-7 using the SelectMAP configuration. Upon a proper PR of a tile on the Artix-7, the DONE pin should toggle, toggling the attached green LED.

An important feature of ControlOS is that it can perform a simulated fault injection into the Artix-7 system. Simulating a fault will help characterize the computer system and ensure that the fault tolerant features are still working correctly. To inject a fault, ControlOS partially reconfigures one of the tiles. When partial reconfiguration occurs, the tile is essentially reset. This causes the tiles output to no longer match that of the other two active tiles. The voter recognizes the problem and ControlOS brings on a new tile and repairs the faulted one. Figure 15 shows the GUI screen during normal operation. In this case the first three tiles are active. Figure 16 shows the GUI screen during a tile fault. In this image, tile five has just been faulted (red) and tile eight was brought online to run with tiles six and seven. This fault injection method can also be used with "bad bitstreams". Instead of partial reconfiguring with the same tile partial bitstream, a known corrupted bitstream in used to re-program the tile. In this case the voter also sees it as a faulted tile, triggering the repair process. The "bad" bistreams are also contained on the SD card. Table 3 shows the memory map of the SD card containing the clean initial and partial bitstreams as well as the corrupted full and partial bitstreams.

Table 3. Artemis Bitstream Start Addresses and Lengths

Tile	Hex Start Address	Hex Length
Full – "Good"	x00000400	x00947A5C
0	x00948000	x000C1530
1	x00A09600	x000AF2B0
2	x00AB8A00	x00109210
3	x00BC1E00	x0010C470
4	x00CCE400	x00109210
5	x00DD7800	x0010C470
6	x00EE3E00	x0011E6F0
7	x01002600	x0010C470
8	x0110EC00	x000C1530
Full – "Bad"	N/A	N/A
0	x01B17E00	x000C1530
1	x01BD9400	x000AF2B0
2	x01C88800	x00109210
3	x01D91C00	x0010C470
4	x01E9E200	x00109210
5	x01FA7600	x0010C470
6	x020B3C00	x0011E6F0
7	x021D2400	x0010C470
8	x022DEA00	x000C1530

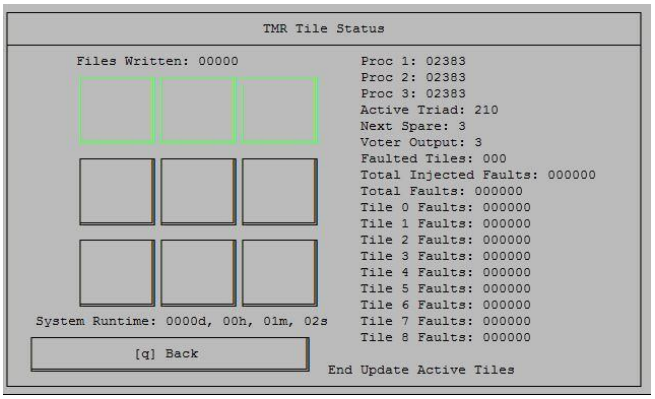


Figure 15. Artemis GUI Tile Status Screen Showing Normal Operation.

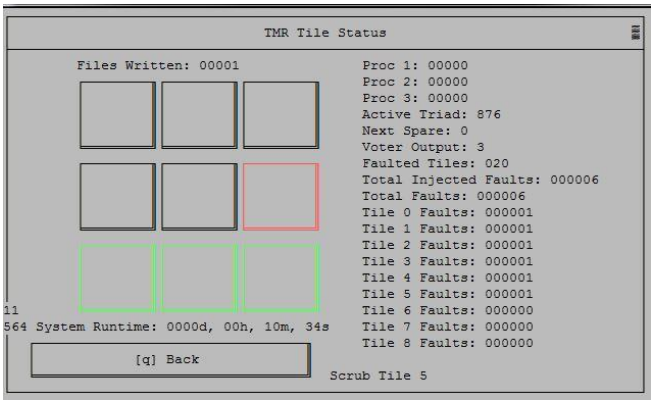


Figure 16. Artemis GUI Tile Status Screen During a Tile Fault.

The blind scrubber task is run once per file write, about every seven hours for the Artemis mission. The blind scrub is performed in the same way initial configuration of the Artix-7 is performed over the SelectMAP port. Blind scrubbing is performed to reset the static part of the design, including the voter and multiplexer, in case they have been effected by radiation.

For Artemis to log data files, ControlOS directs the accumulation of data, the operation of the data SD card on the data board as well as actually writing a file to the SD card. ControlOS communicates with the MAX14502 SD-to-USB card reader chip on the data board to ensure it is in the proper mode. When a file is to be written, ControlOS becomes the master which places the MAX chip into pass-thru mode. ControlOS then gives the MAX chip a done signal, allowing it to revert to card-reader mode. Best practice includes power

cycling the SD card when switching between card reader and pass-thru modes. Power cycling the SD card prevents the MAX14502 from trying to communicate with the SD card while it is transitioning to pass-thru mode [12].

6. Test Results

Test Procedures and Characterization

As the subsystem PCBs of Artemis were assembled, safe-to-mate testing was performed. Once the PCB stack was completely built, the FPGA digital design and software were developed. Once Artemis was complete testing of the ISS configuration began

Duration testing was performed in the weeks leading up to the Artemis hardware delivery. This involved running the stack for as long as possible. If a bug was found, a fix would be developed and tested on an identical development unit of Artemis. If the bug fix was successful, the new configuration would be transfer to the Artemis flight unit and would repeat the test cycle.

Thermal testing of the experimental FPGA was also performed. A bug found in ControlOS showed that the Artix-7 temperature was not printing properly in the data files. Further investigation showed that ControlOS was receiving the temperature value properly from the temperature sensor. ControlOS also provided the temperature value properly to the GUI, but the reason it will not print in a data file has yet to be determined. To estimate a temperature on the Artix-7, an external temperature probe was attached to the top of the Artix-7. Airflow as restricted around Artemis during thermal testing to try and simulate restricted airflow environment on the ISS. This thermal testing revealed that the Artix-7 external temperature never exceeded 45 degrees Celsius which is 80 degrees Celsius below the maximum recommended temperature. Further thermal testing and characterization will need to be performed for the stand-alone satellite missions.

Throughout duration, thermal and functional testing the average nominal current draw of Artemis was characterized as 0.402 Amps. This is within the power requirement provided by NanoRacks.

On-Orbit Data

With Artemis delivered to our flight providers, a waiting game is played. In the best-case scenario, Artemis will be delivered to the ISS in early December by HTV-6 launched from Japan. If this is the case, on-station data will be available a few weeks after launch. By the time of the presentation of this paper we hope to have received and analyzed data from Artemis in orbit on the ISS.

7. Future Work

While awaiting data from Artemis on the ISS, the radiation tolerant system is being developed and integrated for a satellite mission. Two versions of this mission are being developed, dubbed RadSat-u and RadSat-g (Figure 17-18). RadSat will be a 3U CubeSat that will be launched from the NanoRacks CubeSat Deployer (NRCSD) on the ISS. With the provided support from the Space Science and Engineering Laboratory (SSEL) at MSU, the ground operations will be conducted from MSU in Bozeman Montana. RadSat-u will incorporate a new version of the radiation sensor developed by an MSU senior capstone design team. RadSat-g has been selected for launch by the NASA CubeSat Launch Initiative (CSLI) in 2017.

RadSat will be comprised of an avionic stack and the experimental payload stack. The avionic stack is comprised of PCBs that have flown on previous satellites developed by SSEL. Batteries charged with solar panels will provide unregulated power to the electrical power system (EPS). A command and data handling board (C&DH), a radio board and a multifunctional interface board (MFIB) make up the rest of the avionics stack. The experimental payload will be a version of the radiation tolerant computer discussed in this paper. Commands and data will occur between the MFIB and the payload while power is provided by the EPS board.

Depending on RadSat's orbit, the satellite should be in a harsh radiation environment for 12 months. This mission will finally give the radiation tolerant computer system enough time in a radiation environment to advance to TRL-8 or TRL-9.

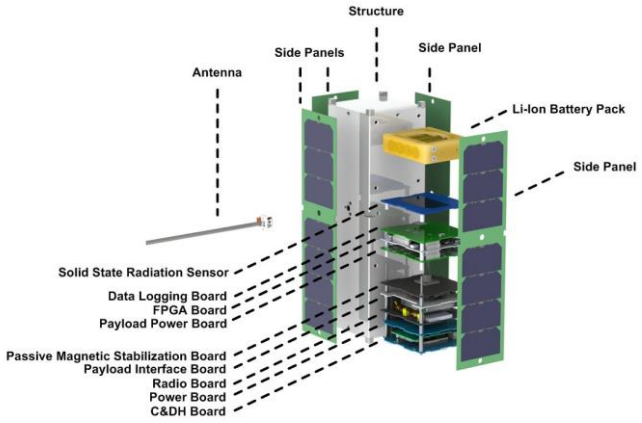


Figure 17. Exploded CAD Model of RadSat.

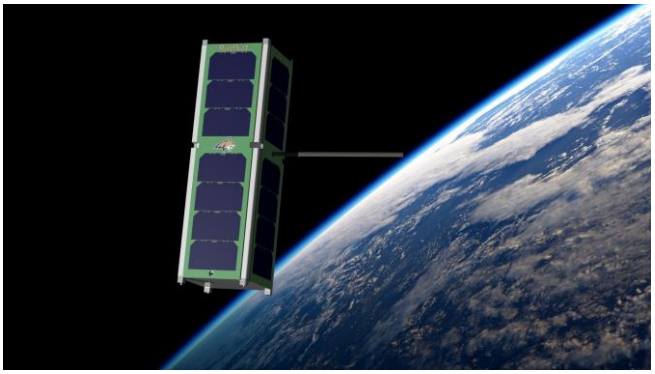


Figure 18. CAD Model Rendering of RadSat.

8. CONCLUSION

This paper presented research and work performed to further the development of an FPGA based radiation tolerant computing system. MSU's fault detection architecture utilizes the partial re-configurability of an FPGA to mitigate SEEs in the device. Although technology maturation opportunities have been prevalent, the systems true test lies in long duration testing in harsh radiation environments like those experienced in low earth orbit. The research team at MSU will eagerly await the launch, integration and data collection phases of the Artemis mission on the ISS.

ACKNOWLEDGEMENTS

This work is and has been supported by NASA Cooperative Agreements NNX10AN32A, NNX10AN91A, NNX12AM50G, NNX14AL03A, NNX13AR03A and NNX16AI75A. The authors would also like to thank NASA CSLI, Montana Space Grant Consortium and the Space Science and Engineering Lab for their expertise and support.

REFERENCES

[1] "NASA Space Technology Roadmaps – Modeling, Simulation, Information Technology & Processing Roadmap, Technology Area 11", [Online], Available: www.nasa.gov/offices/oct/home/roadmaps/index.html , Nov. 2010.

[2] B. J. Lameres, et. al. "RadSat - Radiation Tolerant SmallSat Computer System". AIAA/USU Conference on Small Satellites. 2015.

[3] A. Bar-Lev, Semiconductor and Electronic devices, 3rd ed. Prentice Hall, 1993.

[4] H. Barnaby, "Total-ionizing-dose effects in modern cmos technologies," Nuclear Science, IEEE Transactions on, vol. 53, pp. 3103–3121, 2006.

[5] A. Holmes-Siedle, L. Adams, "Handbook of radiation Effects", 2nd edition, New York, Oxford University Press, 2002.

[6] Fabula, J., et al., "The Total Ionizing Dose Performance of Deep Submicron CMOS Processes", Military and Aerospace Programmable Logic Devices (MAPLD) Conference, Annapolis, MD, Sept 15-18, 2008.

[7] Quinn, H. et al., "High-Performance Computing for Airborne Applications", Nuclear and Space Radiation Effects Conference (NSREC), Denver, CO, July 19, 2010.

[8] "Vivado Design Suite User Guide: Partial Reconfiguration". Xilinx. April 2016.

[9] "NanoRacks Internal Platforms 1A/2A and NanoLab Modules Interface Control Document". NanoRacks LLC. February 2014.

[10] "Interface Control Document between CubeLab Modules and the NanoRacks Platform". Space Systems Laboratory, University of Kentucky. January 2011.

[11] R. J. Weber, "Reconfigurable hardware accelerators for high performance radiation tolerant computers," Ph.D. dissertation, Montana State University, May 2014.

[12] S. A. Harkness, "Experiment Platform to Facilitate Flight Testing of Fault Tolerant Reconfigurable Computer Systems," M.S. Thesis, Montana State University, May 2015.

[13] UCD90124 Datasheet, "12-Rail Sequencer and System Health Monitor with Fan Control". Texas Instruments. September 2010.

[14] LMR14203 Datasheet, "LMR14203 42Vin, 0.3A Step-Down Voltage Regulator in SOT-23". Texas Instruments. April 2013.

[15] LMR62014 Datasheet, "LMR62014 20Vout, 1.4A Step-Up Voltage Regulator in SOT-23". Texas Instruments. April 2013.

[16] "Using the TPS6215x in an Inverting Buck-Boost Topology" Application Report. Texas Instruments. June 2013.

[17] PMBus Specification Version 1.1, Feb. 5, 2007. [Online.] Available: http://pmbus.org/specs.htm

[18] "Artix-7 FPGAs Data Sheet: DC and AC Switching Characteristics". Xilinx. March 2015.

[19] "Clocking Wizard v5.1". Xilinx. April 2015.

BIOGRAPHY

Connor R. Julien *received the B.S. degree in electrical engineering from Montana State Univ., Bozeman in 2015 and is currently an M.S. degree candidate in electrical engineering at Montana State Univ., Bozeman with an expected graduation date of May 2017. He is currently a Graduate Research Assistant in the Department of Electrical and Computer Engineering at Montana State University (MSU), Bozeman where his focus is on reconfigurable computing architectures to detect and repair faults caused by high energy radiation.*

Brock J. LaMeres *(M'98-SM'09) received the B.S. degree in electrical engineering from Montana State Univ., Bozeman in 1998, and the M.S. degree in electrical engineering from the Univ. of Colorado, Colorado Springs in 2001, and the Ph.D. degree in electrical engineering from the Univ. 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 teaches and conducts research in the area of digital systems. Prior to joining the faculty at MSU in 2006, he worked as a Hardware Design Engineer for Agilent Technologies in Colorado Springs from 1999 to 2006. LaMeres' research is sponsored by NASA, the National Science Foundation, the Montana Space Grant Consortium, the National Space Grant Consortium, and the Office of Naval Research.*

Raymond J. Weber *(S'07-M'15) received the B.S. degree in computer engineering in 2008, the M.S. degree in electrical engineering in 2010 and the Ph.D. in engineering from Montana State University in 2014. He is currently a Design Engineer at Flat Earth Inc. in Bozeman MT where his work is focused on digital signal processing algorithm design and implementation.*