A RADIATION TOLERANT COMPUTER MISSION TO THE INTERNATIONAL SPACE STATION

by

Connor Russell Julien

A thesis submitted in partial fulfillment
of the requirements for the degree

of

Master of Science

in

Electrical Engineering

MONTANA STATE UNIVERSITY
Bozeman, Montana

April 2017

TABLE OF CONTENTS

TABLE OF CONTENTS — CONTINUED

TABLE OF CONTENTS — CONTINUED

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

LIST OF FIGURES – CONTINUED

Figure Page

LIST OF FIGURES – CONTINUED

Figure Page

LIST OF FIGURES – CONTINUED

Figure Page

LIST OF FIGURES – CONTINUED

Figure Page

ABSTRACT

The harmful effects of radiation on electronics used in space poses a difficult problem for the aerospace industry. Memory corruption and other faults caused by the harsh radiation environment are difficult to mitigate. The following Masters of Science thesis describes the design and testing of a radiation tolerant, low-cost computer system to meet the increasing demand of fault tolerant space computing. The computer is implemented on a modern Field Programmable Gate Array (FPGA), which enables a novel fault mitigation strategy to be deployed on a commercial part, thus reducing the cost of the system. Using modern processing nodes as small as 28nm, FPGAs can provide increased computational performance and power efficiency. Common mitigation techniques like triple modular redundancy and memory scrubbing are expanded by utilizing partial reconfiguration on the FPGA and by introducing extra spare processors. Our computer system has been in development at Montana State University for the past 10 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 currently being demonstrated onboard the International Space Station and will be the payload for two stand-alone small satellite missions in low Earth orbit in 2018. This Masters of Science thesis presents improvements to the system by moving the design to a new, low power FPGA with a new processor synchronization method. This thesis will present the design, testing, and characterization of the computer system along with conveying data collected by the experiment on the International Space Station.

INTRODUCTION

A radiation tolerant, Field Programmable Gate Array (FPGA) based computing system has been under development at Montana State University (MSU) for the last ten years that promises to 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) radiation effects, with commercial parts achieving greater than 500krad of tolerance simply due to the thin gate oxides and relatively deep isolation trenches found in modern devices. However, the use of modern processing nodes makes commercial parts uniquely susceptible to single event effects (SEEs) due to high energy ionizing radiation. 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 processing tiles. Each tile can fully contain the circuit of interest and be individually reprogrammed using partial reconfiguration (PR). 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. 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.

The radiation tolerant computing system is comprised of a stack of printed circuit boards (PCBs) that fit a 1U ($100mm^3$) 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. This form factor supports testing on the ISS or, when paired with avionics systems, a satellite mission.

Prototyping of this radiation tolerant computer system began in 2007. Since then, the system has been tested in front of a cyclotron and has flown on both high-altitude balloons and sounding rockets. To continue progressing the maturity of the system, testing aboard the International Space Station (ISS) is currently being performed. 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 incorporated as the experimental payload on two 3U CubeSats to be launched from the ISS in 2018.

The following thesis covers Artemis, a flight unit of the computer being demonstrated on the ISS. The Artemis unit launched to the ISS on the HTV-6 mission

out of Japan on December 9th of 2016. HTV-6 is operated by the Japanese Aerospace Exploration Agency (JAXA) and launched from the Tanegashima Space Center in Japan to resupply the ISS. Data from this mission is currently being collected and will be discussed in a later chapter.

The following chapters communicate 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 described. The system design for Artemis as well as the details for its mission to the ISS are explained. 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 system is referred to as the Radiation Tolerant Computing System (RTCS) and the name Artemis is used to describe the specific version of the RTCS that is being tested in low Earth orbit on the ISS.

MOTIVATION

Space Computing Requirements

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 for 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]. Maintaining high processing power, low power consumption and enhancing reliability for space missions are the biggest challenges to meeting this objective. Designing modules to perform onboard processing is also becoming more popular. Data acquisition and compression before downlink are important functional requirements when working with telemetry budgets and increased processing power. 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 that implement a novel SEE mitigation strategy to meet the needs of NASA's flight computing technology.

Radiation Effects on Space Electronics

Total Ionizing Dose (TID) and Single Event Effects (SEEs) 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. A visualization of TID effects in a MOS device is shown on the right in Figure 2-1 below. Most failures caused by TID occur because the electron-hole pair gets trapped in the insulating material of the electronic device [3]. 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].

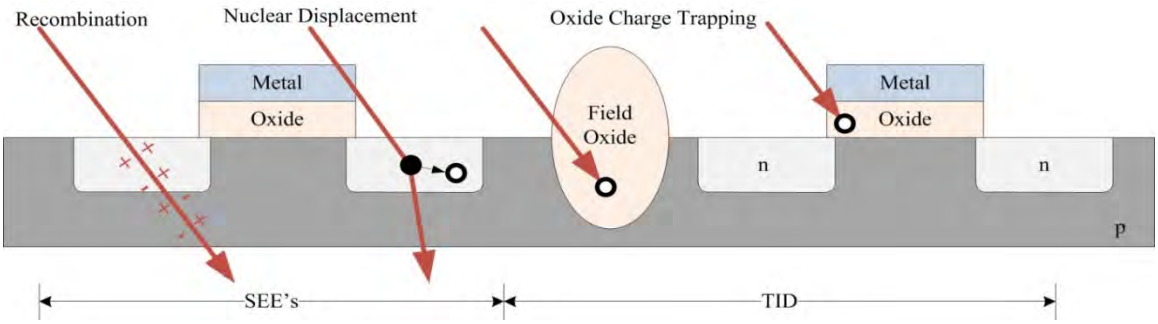

Figure 2-1. MOSFET cross-section showing radiation faults by SEEs and TID [5].

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 a voltage pulse to propagate through the circuit it is referred to as a Single Event Transient (SET). When an SET occurs and the transient is latched by a flip-flop or another memory element it is called a Single Event Upset (SEU). These changes in logic state are referred to as a "bit-flip". SEUs affect both bipolar junction transistors (BJTs) and metal-oxide-semiconductor field-effect transistors (MOSFETs) [6]. 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. A Single Event Latchup (SEL) is also a SEE. An SEL occurs when two transistors are stuck in states that short power to ground, causing excessive current draw and permanently damaging the device. Figure 2-1 above shows the effects of SEEs on a MOSFET.

High energy ionizing radiation is detrimental to space electronics. Ionizing radiation has sufficient energy to break electrons free from atoms and molecules. The general concern is over radioactive particles that are traveling at relativistic speeds. Alpha and Beta particles traveling at relativistic speeds can cause soft errors but are generally mitigated by shielding. Gamma radiation can potentially pass through shielding but with energies of a few hundred keV, they don't have sufficient energy to affect digital hardware. The main concern for radiation damage comes from galactic cosmic rays (GCR). These cosmic rays originate outside the galaxy and provide a

continuous radiation environment that permeates interplanetary space [6] [7]. Galactic cosmic rays are made up of 85% protons, 14% alpha particles and 1% heavier nuclei. These particles may have energies extending up to 1 GeV. These particles are highly charged which makes them densely ionizing. Table 2-1 shows the energy levels of various radioactive particles. High energy ionizing 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, an electron could be excited from the valence band to the conduction band, creating an electron-hole pair [8].

Table 2-1. Energies of various particles [3].

Particle Type	Energy Range
Trapped protons and electrons	≤ 100 MeV
Alpha particles	5 MeV
Solar protons	≤ 1 GeV
Cosmic rays	≥ 1 GeV

The magnetic fields lines making up the magnetosphere also trap low energy charged particles [9]. These particles are diverted from their path and become trapped in the Van Allen Belts, shown in Figure 2-2 [7]. Trapped particles spiral around the magnetic field lines as they drift around the Earth. Particles trapped in the Van Allen belts have been known to cause soft errors in electronics but are usually mitigated by certain amounts of shielding [10]. A region off the coast of Brazil called the South Atlantic Anomaly (SAA) is created due to the magnetic field lines being offset from Earth's axis of rotation, show in Figure 2-3. This area is where the radiation belts are

closest to Earth. Radiation intensity can increase by an order of magnitude when traversing the SAA, causing damaging effects on electronics [7].

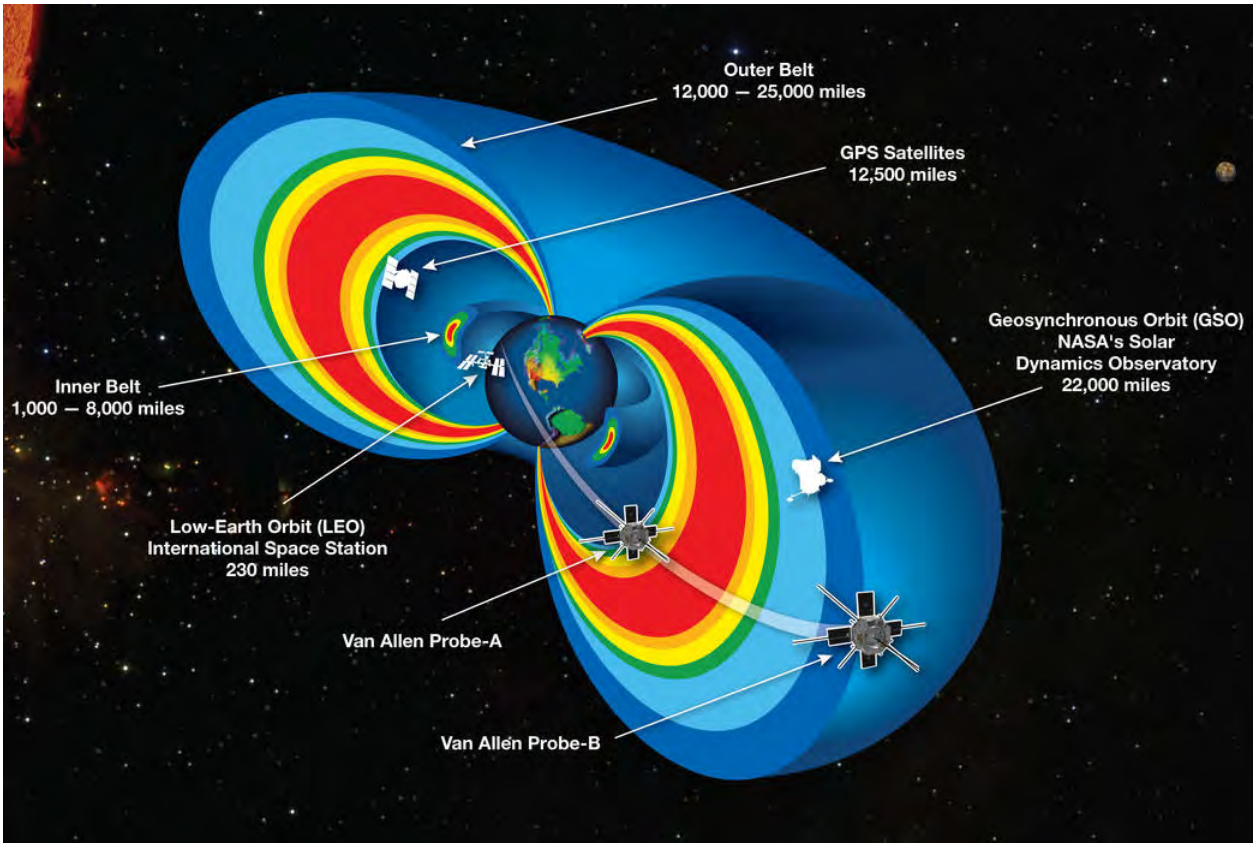

Figure 2-2. Visualization of the Van Allen belts compared to satellite orbits [11].

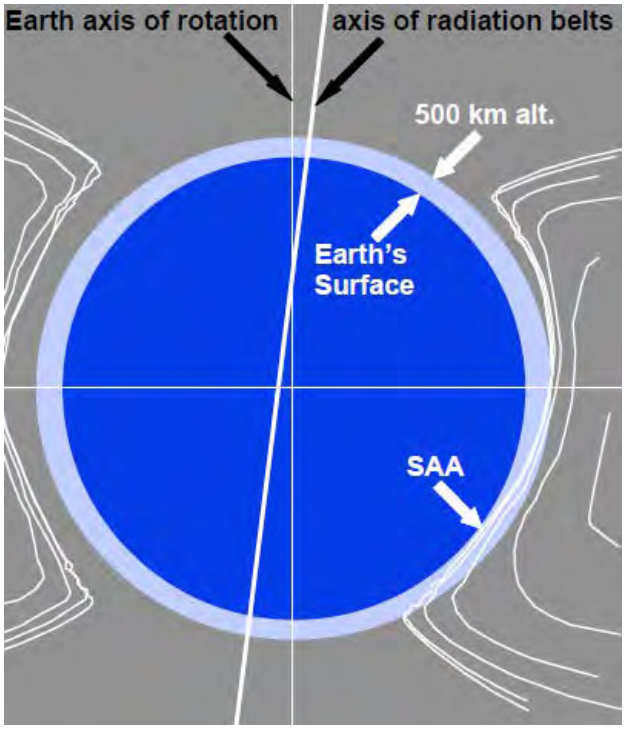

Figure 2-3. Radiation belts and location of the South Atlantic Anomaly [7].

Common Radiation Mitigation Techniques

Techniques used to reduce the harmful effects of radiation include shielding and creating radiation hardened electronics as well as enacting specific types of fault tolerant architectures.

Shielding

Shielding is effective in preventing alpha and beta particles from reaching electronics but gamma particles, neutrons and heavy ions are not affected by the thickness of the shielding. Figure 2-4 shows that more than doubling the shielding thickness reduces the proton dose received by less than a factor of two [9]. When a strict mass budget is being followed, it is impractical to fly thicker shielding when it will be

relatively ineffective at reducing the GCR spectrum. Shielding can even cause cascading particles of secondary radiation when the structure is bombarded with a GCR.

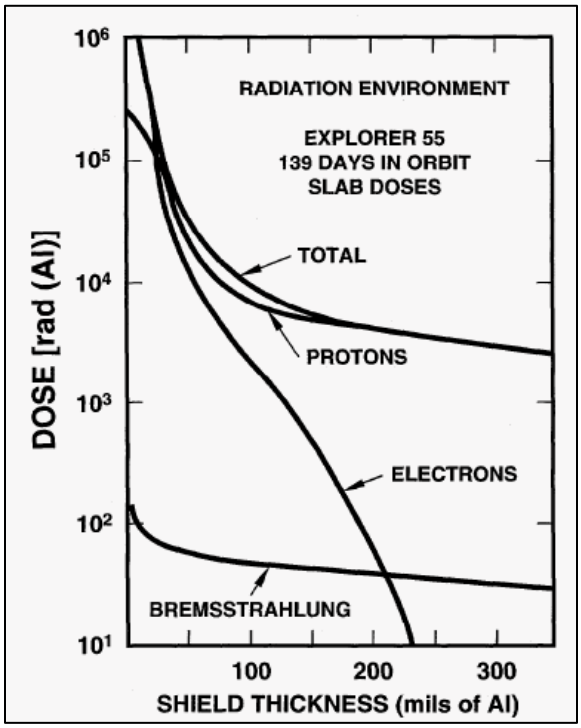

Figure 2-4. Total dose as a function of aluminum shielding thickness [9].

Material Hardening

A silicon device solution to the radiation effects problem is radiation hardening by process (RHBP). Hardening devices with RHBP involves changing the steps involved in the fabrication of the silicon device. Silicon on insulator (SOI) technology is an example of hardening by process. This technology involves adding an insulator layer below the silicon junction (Figure 2-5). The insulating layer can be silicon dioxide or sapphire (also called Silicon on Sapphire(SOS)). Developed to reduce a devices parasitic capacitance,

SOI and SOS devices also decrease charge trapping by greatly reducing the gate oxide thickness [6]. Device isolation is also a benefit of this technology.

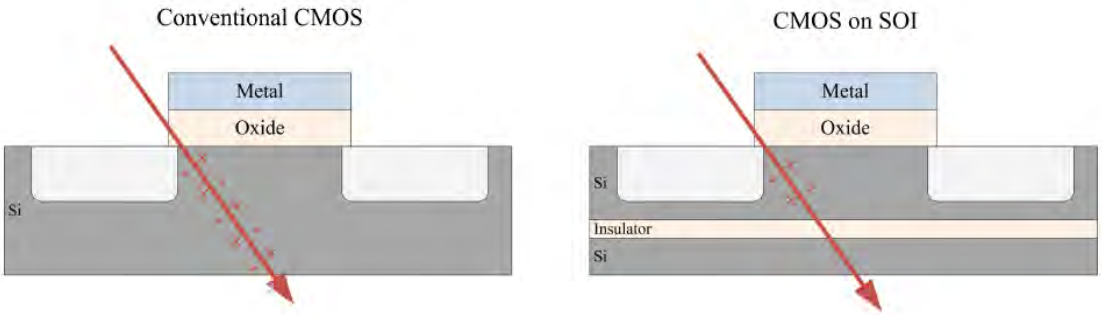

Figure 2-5. Conventional CMOS transistor compared to a silicon-on-insulator device [5].

Another way to prevent charge from getting trapped in the insulating layer is known as radiation hardening by design (RHBD). This technique encompasses using isolated transistors and guard rings to provide conduction paths (Figure 2-6). Charge induced by radiation flows through the conduction path instead of remaining trapped in the insulating material [12].

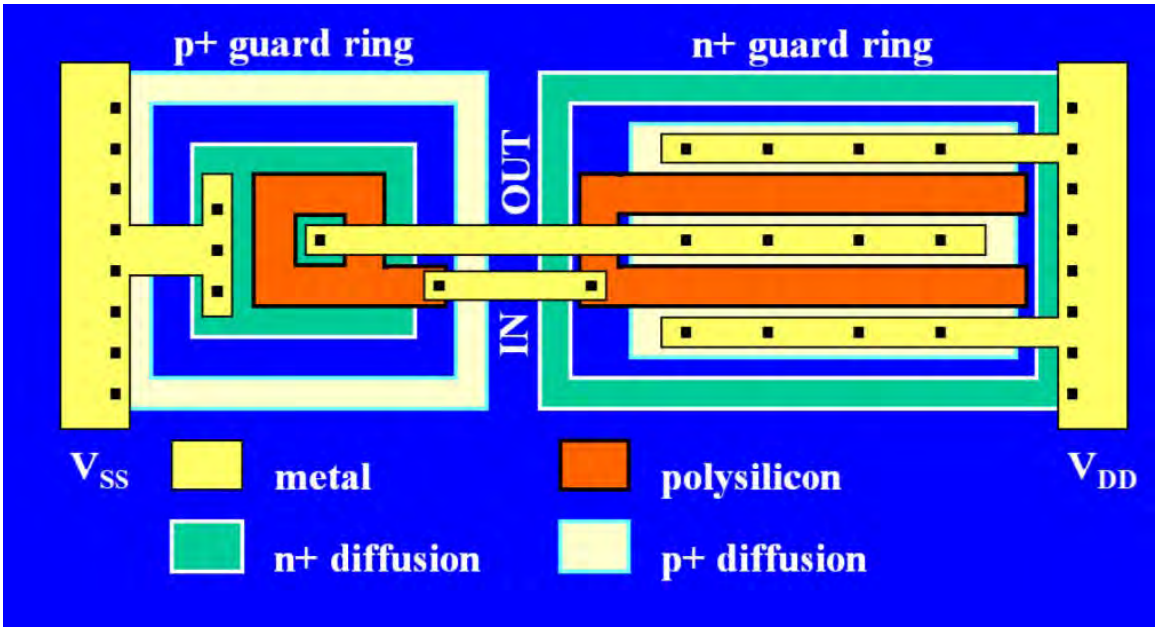

Figure 2-6. Radiation Hardening by Design inverter using isolated transistors with guard rings [13].

Architectural Hardening

Shielding, RHBP and RHBD provide extended tolerance for TID effects on electronics. However, SEEs can still cause faults in the system. Therefore, architectural methods are utilized to help mitigate SEEs. The conventional approach is to use circuit redundancy and memory checking to recognize SEE faults and correct them.

Triple Modular Redundancy (TMR) is a common SEE mitigation technique. A design utilizing TMR has three copies of the required circuit with the outputs fed through a voter circuit. If one of the outputs is different than the majority output, a fault may have occurred in one of the circuit copies. The two other circuits produce a valid output while the faulted circuit is repaired and TMR operation can continue.

Memory scrubbing is a process where the configuration memory of a device is compared to a known good copy. This “golden copy” would ideally be stored in a

radiation-tolerant memory such as PROM or EEPROM. Blind memory scrubbing involves re-writing the memory location with the golden copy even if no memory fault has occurred. Alternatively, readback memory scrubbing checks the memory location for corruption, determining if a memory scrub is necessary. Readback scrubbing has the potential to save time by only having to perform a read and determine that a write is not necessary. Readback scrubbing can also potentially be a way of accounting for radiation faults that occur in memory when blind scrubbing would just correct the fault without recognizing that a fault existed.

Triple modular redundancy and memory scrubbing (TMR+Scrubbing) are generally used together to mitigate radiation effects [14]. Using TMR makes the device immune to single upsets in the control or data blocks, protecting user or configuration memory. Memory scrubbing corrects bit flips in the configuration memory, keeping the number of upsets as low as possible.

Existing Radiation Hardened Processors

Most existing radiation hardened processors are used by the aerospace industry and the military. The Mongoose-V built by Synova, Inc for example, is a radiation hardened flight computer used by New Horizons. The Mongoose-V costs $20,000-$40,000 and runs at 15 MHz [15]. One of the most popular radiation hardened processors is the RAD6000 made by BAE Systems. This processor costs between $200,000 and $300,000 and can run at a maximum of 25 MHz. The RAD6000 was used on the Spirt and Opportunity Mars rovers. The RAD750 succeeded the RAD6000 and

costs around $200,000 but can run at much higher speeds of 200 MHz. Two RAD750's are currently used on the Curiosity Mars rover.

Limitations

The cost of these radiation hard systems greatly exceeds commercial processors with similar processing speeds. While large aerospace and defense companies may have the funds to use these expensive processors, small organizations like startups and Universities require cheaper options. For reference, the system designed in this thesis only costs between $10,000 and $20,000. Radiation hardened processors also lag commercial devices in performance by around ten years. Figure 2-7 shows a performance comparison of commercial processors and radiation hardened processors.

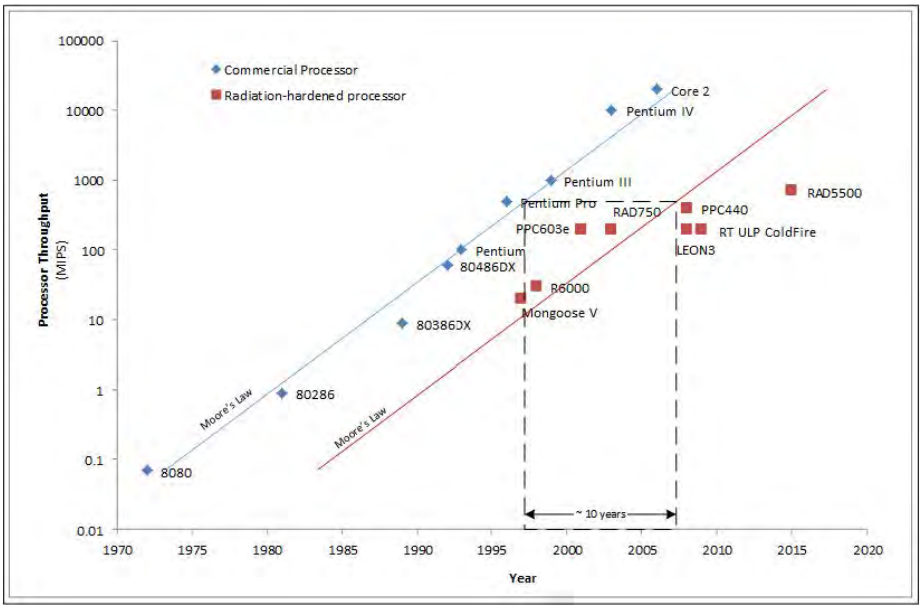

Figure 2-7. Performance comparison of radiation hard processors and commercial processors in the last 50 years [16].

Using FPGAs in the Harsh Environment of Space

An FPGA is a programmable logic device that can be configured to meet application requirements. Programmed by the customer, not the manufacturer, FPGAs use a grid of logic blocks that are routed together to create logic gates and complex combinational functions. Field programmable gate arrays can be used to implement any function that an application-specific integrated circuit (ASIC) could perform but with reduced design time. Floor planning an FPGA design improves the resource allocation to help meet timing considerations.

Functionality of FPGAs

Modern FPGA designers like Altera and Xilinx include numerous functions that play a part in reducing radiation effects. Design techniques utilizing a soft error mitigation (SEM) controller and partial reconfiguration (PR) are advantageous in space demonstrations.

An SEM controller provides a solution to detect and correct soft errors in the configuration memory of an FPGA. These controllers contain error injection, error detection, error correction and error classification functionality. They aim to reduce or eliminate the amount of soft errors caused by SEEs to keep system reliability high. Part of the SEM functionality involves the ability to classify configuration memory errors as “essential” or “non-essential”. Without this function, every error in configuration memory would be classified as essential, potentially causing a system wide mitigation response. Error correcting codes (ECCs) and cyclic redundancy checks (CRCs) are used

to detect the errors in memory. The SEM controller injects faults occasionally to ensure that the rest of the error correction functionality is operating properly [17].

Partial reconfiguration of FPGAs allows a specific region to be reprogrammed without interrupting the rest of the device. The fabric of the FPGA is divided into reconfigurable logic and static logic. A PR is performed by downloading a partial bitstream file to the FGPA which contains information destined for a reconfigurable logic region (Figure 2-8). That region is reconfigured without interrupting other reconfiguration regions or the static logic operations currently in progress [18]. Once the reconfigurable region has been reprogrammed, it can be reset, and then enters normal operation. Partial reconfiguration is important for radiation mitigation because it can reset a region on the FPGA to repair a fault without interfering with the rest of the device.

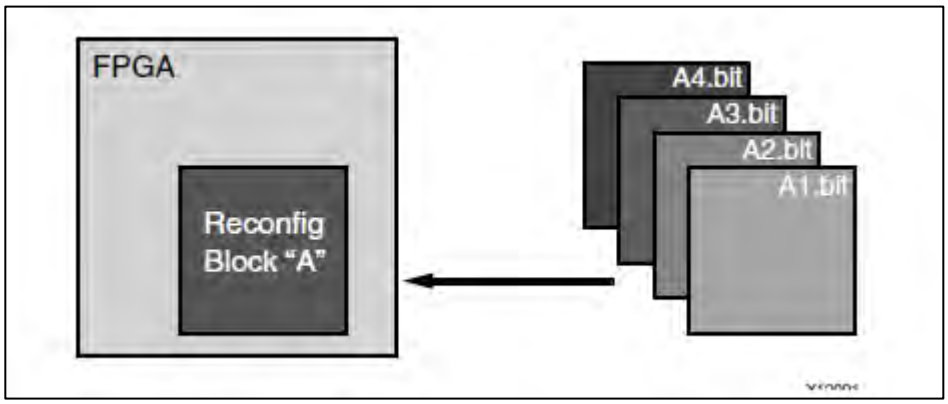

Figure 2-8. A visualization of Xilinx partial reconfiguration with partial bitstreams [18].

Radiation Effects on FPGAs

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 [19]. However, the small diffusion regions in current devices cause increased SEE susceptibility because a radiation strike can carry enough energy to change the state of a device [20]. Configuration memory or design memory can be effected, causing the corresponding logic element to malfunction. Single event effects can cause bit flips and other data corruption. Additionally, the interconnect between logic blocks can also be effected by SEEs, either adding or removing a connection. Figure 2-9 provides a visualization of these errors.

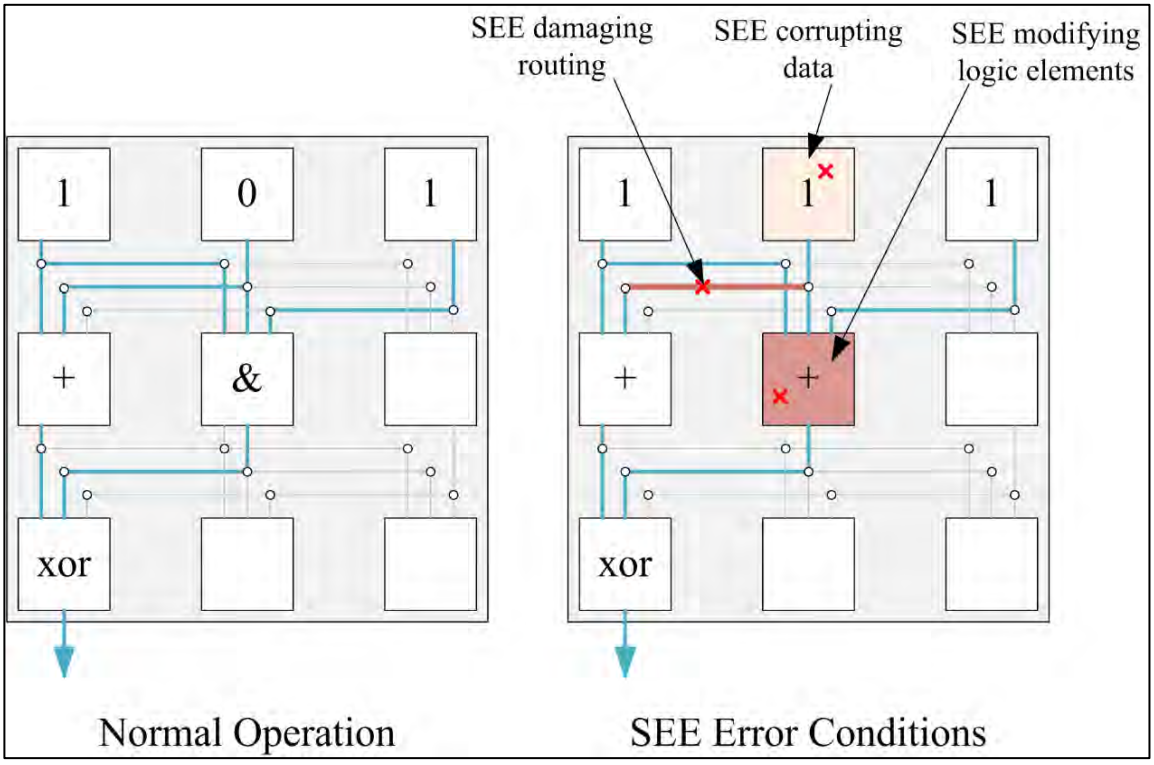

Figure 2-9. Normal FPGA operation compared to operation under SEE errors [5].

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.

MONTANA STATE UNIVERSITY'S CONTRIBUTION

For the last nine 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 RTCS 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. The RTCS uses FPGAs from Xilinx Inc., an American semiconductor manufacturing company. Xilinx FPGAs are generally used for space and military applications so they carry more inherent robustness which suits the needs of space computing.

Mitigation of SEE faults is performed by expanding TMR+Scrubbing to create 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. 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. This functionality is referred to as TMR+Scrubbing+Spares. Figure 3-1 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 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.

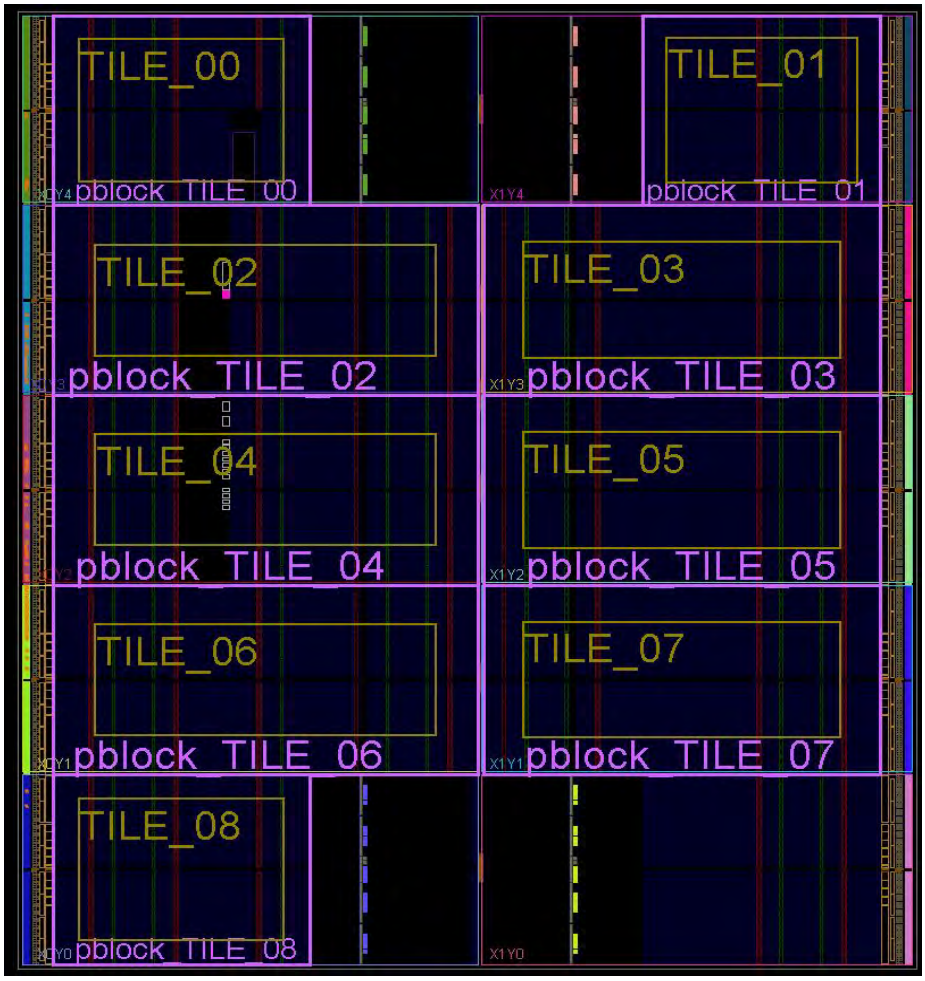

Figure 3-1. Artix-7 floorplan showing reconfigurable regions.

Technology Maturation

A history of the maturation of the RTCS 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. Table 3-1 below provides a description of the different TRLs.

Table 3-1. A description of the nine technology readiness levels defined by NASA [21].

TRL	Description
1	Basic principles observed and reported
2	Technology concept and/or application formulated
3	Analytical and experimental critical function and/or proof-of-concept
4	Component and/or breadboard validation in laboratory environment
5	Component and/or breadboard validation in relative environment
6	System/subsystem model or prototype demonstration in a relative environment (ground or space)
7	System prototype demonstration in a space environment
8	Actual system completed and "flight qualified" through test and demonstration (ground or space)
9	Actual system "flight proven" through successful mission operations.

The RTCS 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 3-2 shows the proof of concept prototype used to reach TRL-3.

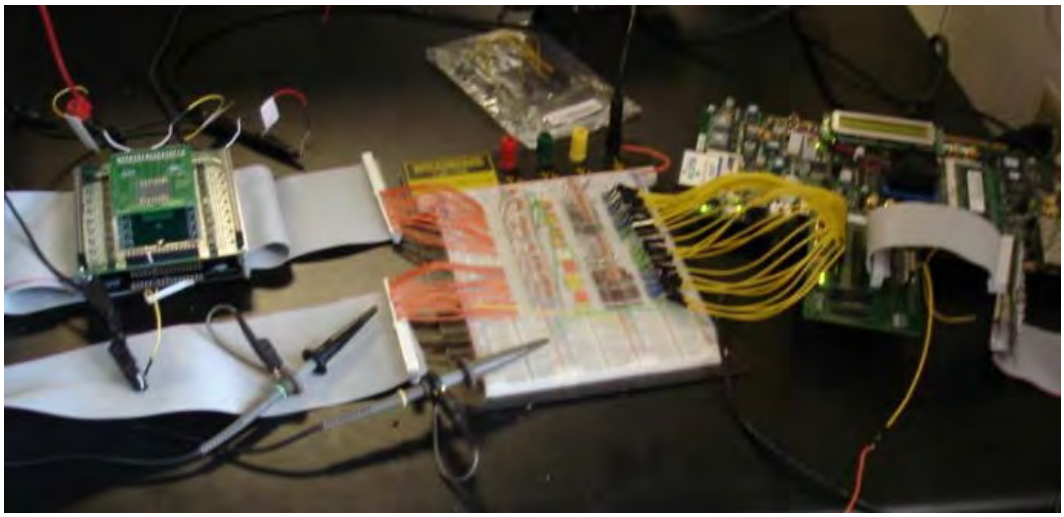

Figure 3-2. Breadboarded prototype of the RTCS.

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 3-3, left). 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 (Figure 3-3, right). Testing with the cyclotron and confirming the integration of the different system components allowed for validation in a laboratory environment and TRL-4.

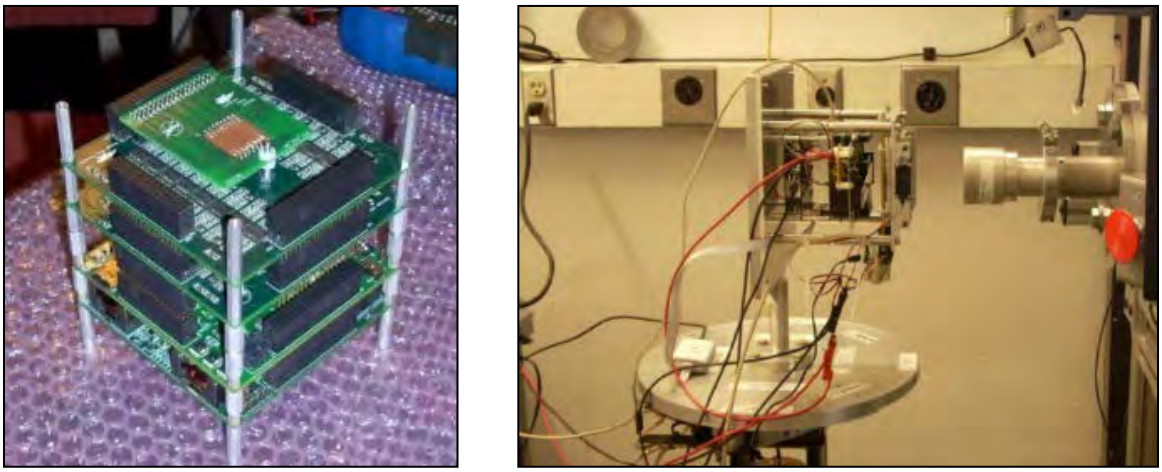

Figure 3-3. RTCS stack (left) and cyclotron testing of the RTCS (right).

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 3-4 shows two of these high-altitude balloon flights.

Figure 3-4. High altitude research balloon flights.

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 3-5). To prepare for this flight, the computer system was again redesigned, this time 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 3-5. Up Aerospace sounding rocket demonstration.

In March of 2015 the system was flown on another sounding rocket out of Wallops, Virginia (Figure 3-6). This rocket flight required much more rigorous testing before launch. Multiple vibration and pressure tests were performed after the integration phase. A more solid deck platform was used to interface the RTCS to the sounding rocket.

Figure 3-6. Wallops sounding rocket (left) and view from the rocket in orbit (right).

Prior Work

Prior work on this project at MSU has included numerous Master's and Doctorate students that continue to develop and enhance this project. Todd Buerkle was involved with designing and testing the ionizing radiation detector [22]. Jennifer Hane developed the fault tolerant FPGA architecture and the interface to the radiation sensor [23]. Justin Hogan worked on modeling reliability of various architectures [24]. Raymond Weber designed the power board and ConrolOS, the operating system that runs on the Spartan-6 [5]. Sam Harkness redesigned the PCBs to fit the 1U form factor as well as developing the data file writing system [3].

A Necessity for Orbital 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 than 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 to progress the TRL of the RTCS, it must be fully tested in low Earth orbit (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.

ARTEMIS 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 [25]. This section covers the mission overview along with the technical requirements of a NanoRacks internal payload (also called a NanoLab).

Mission Overview

As an experiment is built into a NanoLab form factor, the NanoRacks Safety Data Template must be submitted and approved. This document contains material lists and CAD files along with information about potentially hazardous aspects of the NanoLab.

NanoRacks has several options for delivery of the flight unit. For their ISS internal payloads, the flight unit can be and is usually shipped to NanoRacks. Another option is for the flight unit to be hand delivered to NanoRacks in Houston, Texas. A functional test is then performed on the flight unit. This test involves plugging in the NanoLab to a mockup of the Platform/computer system NanoRacks employs on the ISS. NanoRacks leaves the system running for around two days to make sure it is functioning properly. NanoRacks then packages and delivers the NanoLab to Lockheed Martin, who gets it onto the launch vehicle. Then the NanoLab is launched to the ISS on a resupply mission by SpaceX, Orbital ATK or the Japanese Aerospace Exploration Agency (JAXA).

On board the ISS, a rack locker on the Japanese Experiment Module (JEM) contains two NanoRacks Platforms, shown on the left side of Figure 4-1. The purpose of these platforms is to interface the NanoLab modules both structurally and electrically to the ISS. 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.

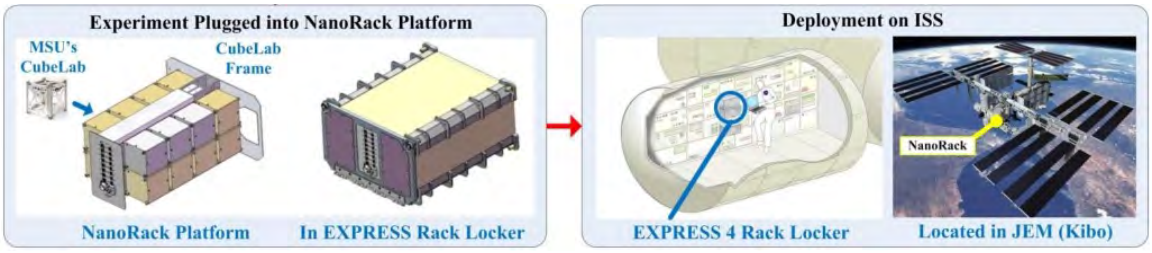

Figure 4-1. NanoRacks mission concept [26].

Concept of Operations

As soon as the NanoLab is plugged into the USB port, it will receive power and begin operations. Artemis will show 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 accesses the USB hub once a week and emails the data files to the customer once they are downloaded from the ISS. When the mission is complete, an astronaut will remove the NanoLab and store it for its return trip to Earth. Once the vehicle has landed, NanoRacks returns the NanoLab to the customer.

Mechanical Requirements

NanoRacks upholds several physical requirements for interfacing the NanoLab to the NanoRacks Platform on the ISS. For the 1U form factor, the NanoLab could not exceed 100mm x 100mm x 100mm in size [25]. A variety of companies supply CubeSat chassis that meet the specifications of a 1U CubeSat, including NanoRacks and Pumpkin. The maximum mass of the NanoLab could not exceed 1,000 grams. Special care must be taken to ensure the USB port spacing, dimensions and orientation of the NanoLab matches the NanoRacks Platform USB port. The USB placement must be on one side of the chassis, as shown in Figure 4-2, and must follow the correct dimensions to ensure the proper fit in the locker. Figure 4-3 shows the side view of the NanoRacks Platform and contains dimensions between USB ports where up to eight NanoLabs can be plugged in. 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 [25] [27].

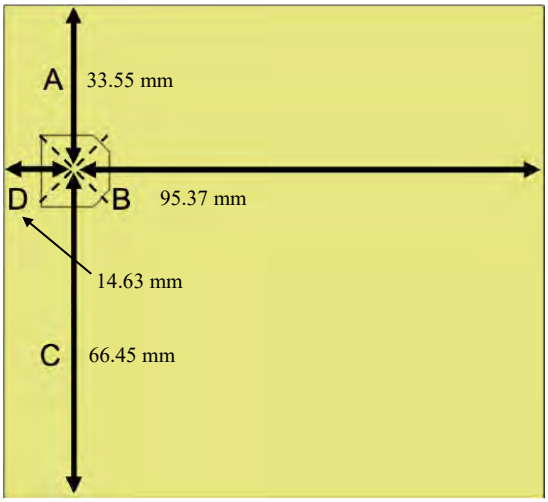

Figure 4-2. NanoLab USB connector placement dimensions [25].

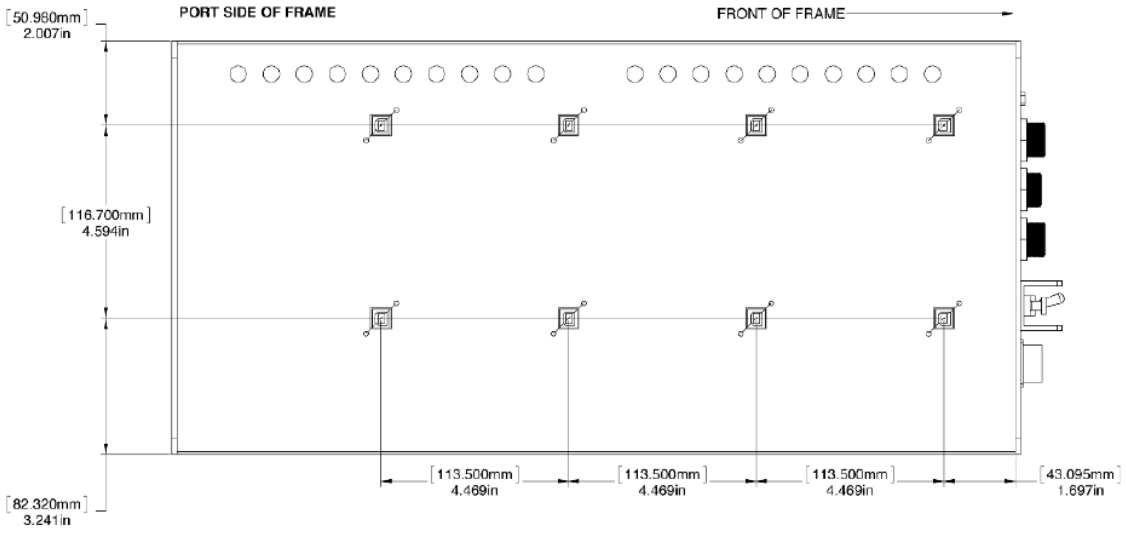

Figure 4-3. Platform 1A and 2A side-view with the front panel to the right [25].

Electrical Requirements and Data Handling

The USB connection from the Platform to the NanoLab provides 2W of power at 5V DC. The D+/D- signal pair for the USB also runs into the platform from the NanoLab. The USB connection inside the Platform is connected to a USB hub which

attaches to a Windows XP embedded computer. The Software Toolkit for Ethernet Lab-Like Architecture (STELLA) allows NanoRacks ground control to operate a DOS command terminal on the embedded computer. This connection allows the ground station to transfer files between the NanoLab, the Platform and the NanoRacks ground station. The NanoRacks ground station can download files from the ISS at a rate of 3 MB/s [25]. Data files are emailed to the client after being download from the ISS.

SYSTEM DESIGN

The following chapter describes a version of the RTCS being tested on the ISS called Artemis, which is the focus of this thesis. 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 section below. To prepare the computer system for the ISS, each PCB had to be modified to fit the form factor required by NanoRacks. The following sections describe each PCB, the FPGA digital design for both FPGAs and the software developed for Artemis. Specific subsections will bring attention to functionality or design changes made as part of this thesis project.

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 ECE 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. Figure 5-1 shows a block diagram of the Artemis system. The following subsections describe the design of each of the PCBs used in Artemis.

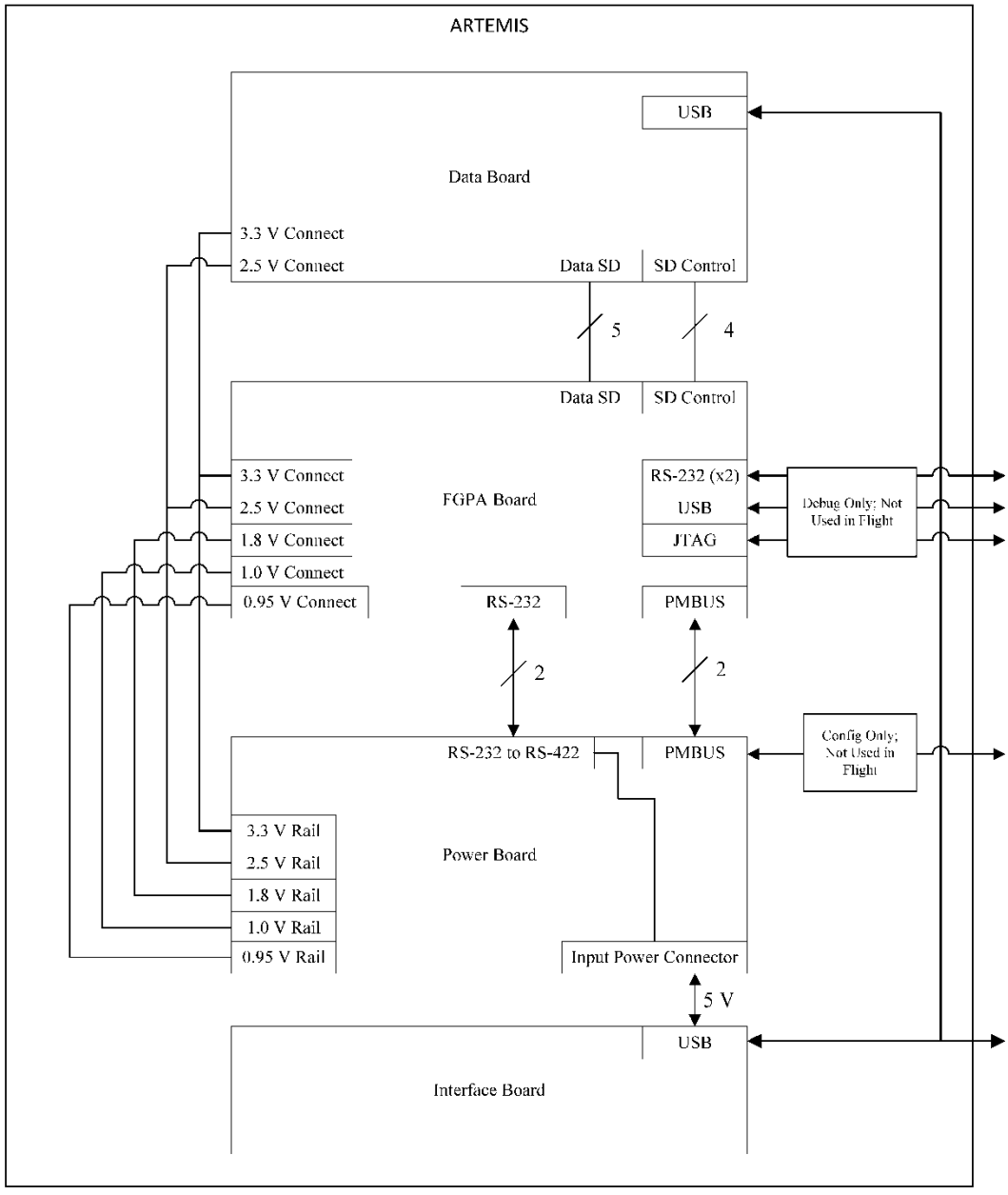

Figure 5-1. Artemis system block diagram.

Interface Board

The bottom board of the PCB stack is an unpopulated version of the data interface board (Figure 5-2). The 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.

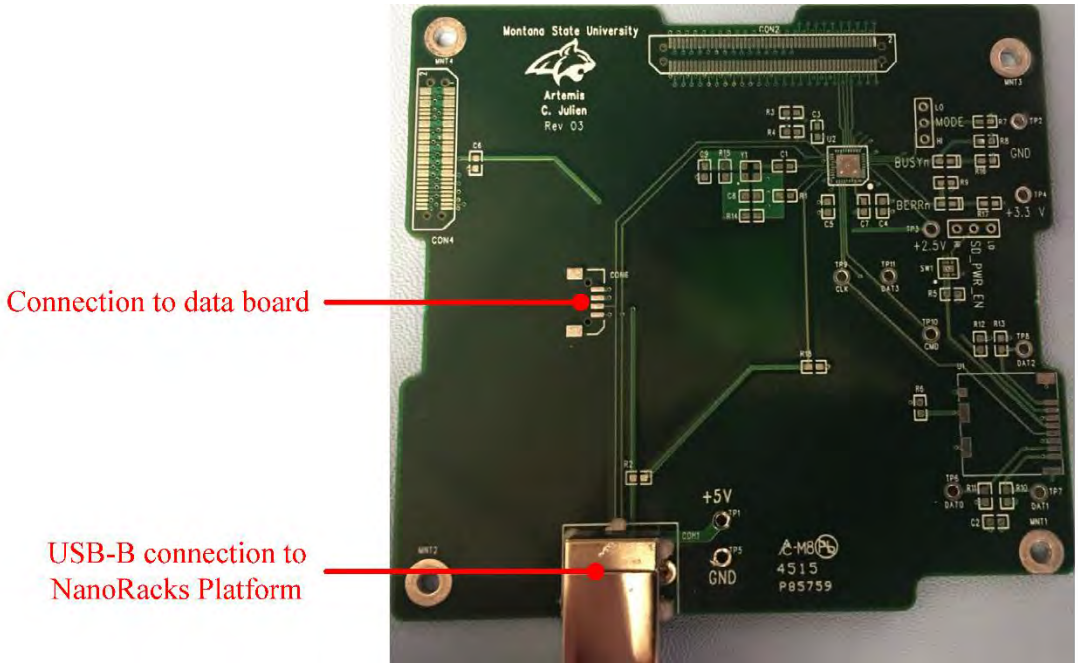

Figure 5-2. Interface Board with important components labeled.

Power Board

The power board (Figure 5-3) has been under development at MSU for the last six years. Further information on the power board can be found in Raymond Weber's dissertation [5]. Originally the board could receive a wide range of input voltages,

typically 9-30 V. When the power board was revised for the Artemis mission, the high voltage input capability was removed to save space on the PCB. It also was not necessary to prepare for a high range of voltage inputs because the ISS supplies a steady 5V to Artemis during its mission. Therefore, the input voltage range for the Artemis power board was reduced to 4-13 Volts [3]. The power board uses linear regulators along with two Texas Instrument (TI) power controllers to produce and monitor eleven voltage rails used by the rest of the Artemis system.

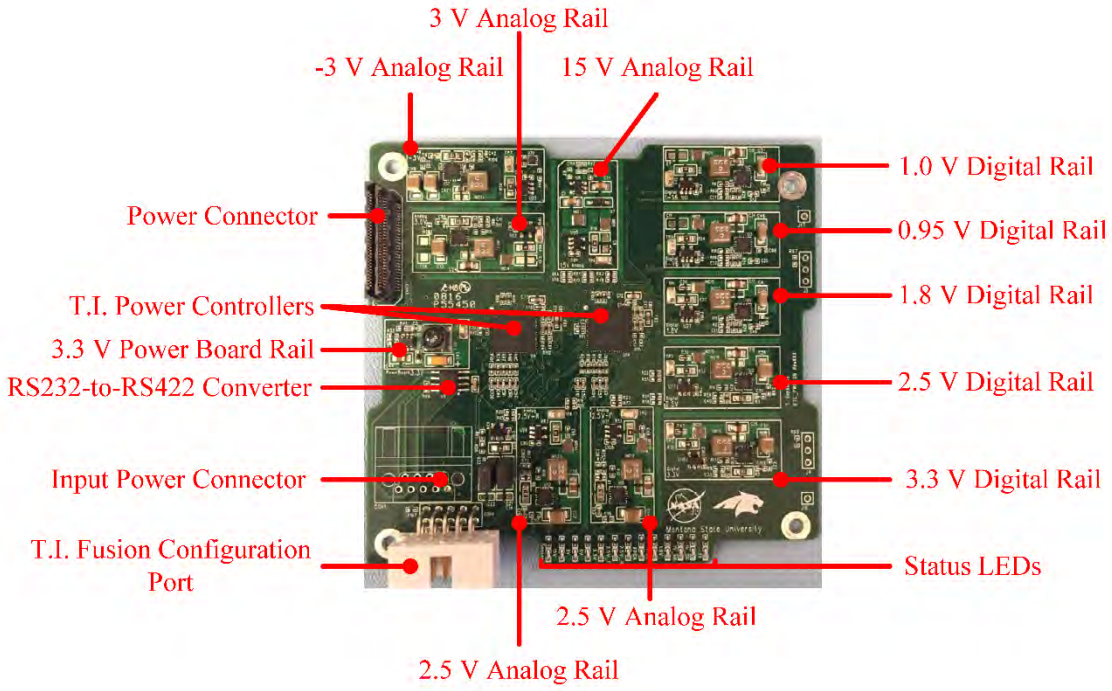

Figure 5-3. Power board with important components labeled.

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 [28]. 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) [29]. A T.I. 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 [30]. 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 [31]. 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 5-1 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 5-1. The Artemis power rails, regulator for each rail and the system it supplies.

Voltage Rail	Regulator	System
3.3V	LMR14203	Power controller supply voltage
15V	LMR62014	Bias voltage for exp. (analog, unused on Artemis)
3.3V	TPS62130	FPGA I/O (digital)
2.5V	TPS62130	FPGA digital I/O (digital)
1.8V	TPS62130	FPGA configuration (digital)
1.0V	TPS62130	FPGA core voltage (digital)
0.95V	TPS62130	FPGA core voltage (digital)
3.0V	TPS62130	Amplifier power (analog, unused on Artemis)
-3.0V	TPS62130	Amplifier power (analog, unused on Artemis)
2.5V	TPS62130	Front comparator (analog, unused on Artemis)
2.5V	TPS62130	Back comparator (analog, unused on Artemis)

Two TI power controllers, UCD90124A's [28], 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 [32]. 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 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 VCCINT, VCCBRAM, VCCAUX and VCCO [33]. 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

Previous revisions of the FPGA board included the previous versions of the Virtex FPGA. The Virtex-6 FPGA was the most recently used experimental FPGA. However, this FPGA draws a considerable amount of power. For the Artemis mission, the power budget forced the FPGA to be changed to the low power Artix-7. A new FPGA board was designed by Samuel Harkness that included the Artix-7 as the experiment FPGA. This section describes the design of the Harkness's FPGA board while the Artix-7 FPGA digital design developed by Connor Julien is described in the FPGA Digital Design section below.

The FPGA board (Figure 5-4) 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. 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 Maser 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 [34]. Table 5-2 and Table 5-3 below shows the configuration modes and switch values for the Artix-7 and Spartan-6 respectively. Figure 5-5 provides a block diagram of the FPGA board systems and I/O.

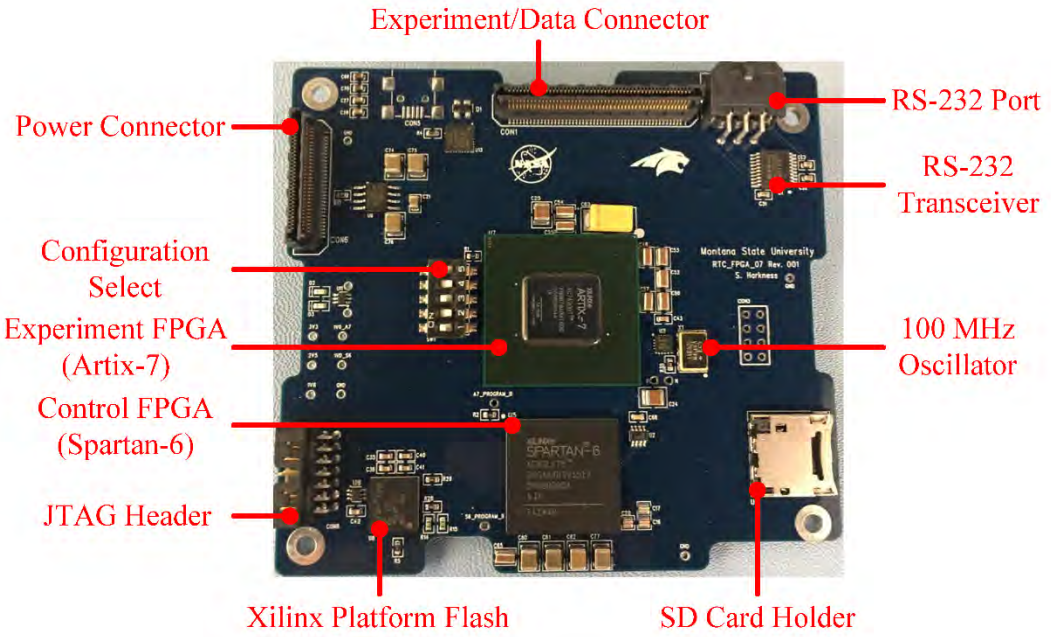

Figure 5-4. FPGA board with important components labeled.

Table 5-2. Configuration modes for the Artix-7 FPGA.

Configuration Mode	M [2:0]	Bus Width	CCLK Direction
Master Serial	000	x1	Output
Master SPI	001	x1, x2, x4	Output
Master BPI	010	x8, x16	Output
Master SelectMAP	100	x8, x16	Output
JTAG	101	x1	Not Applicable
Slave SelectMAP	110	x8, x16, x32	Output Input
Slave Serial	111	x1	Input

Table 5-3. Configuration modes for the Spartan-6 FPGA.

Configuration Mode	M [1:0]	Bus Width	CCLK Direction
Master Serial/SPI	01	1,2,4	Output
Master SelectMAP/BPI	00	8,16	Output
JTAG	XX	1	Input (TCK)
Slave SelectMAP	10	8,16	Input
Slave Serial	11	1	Input

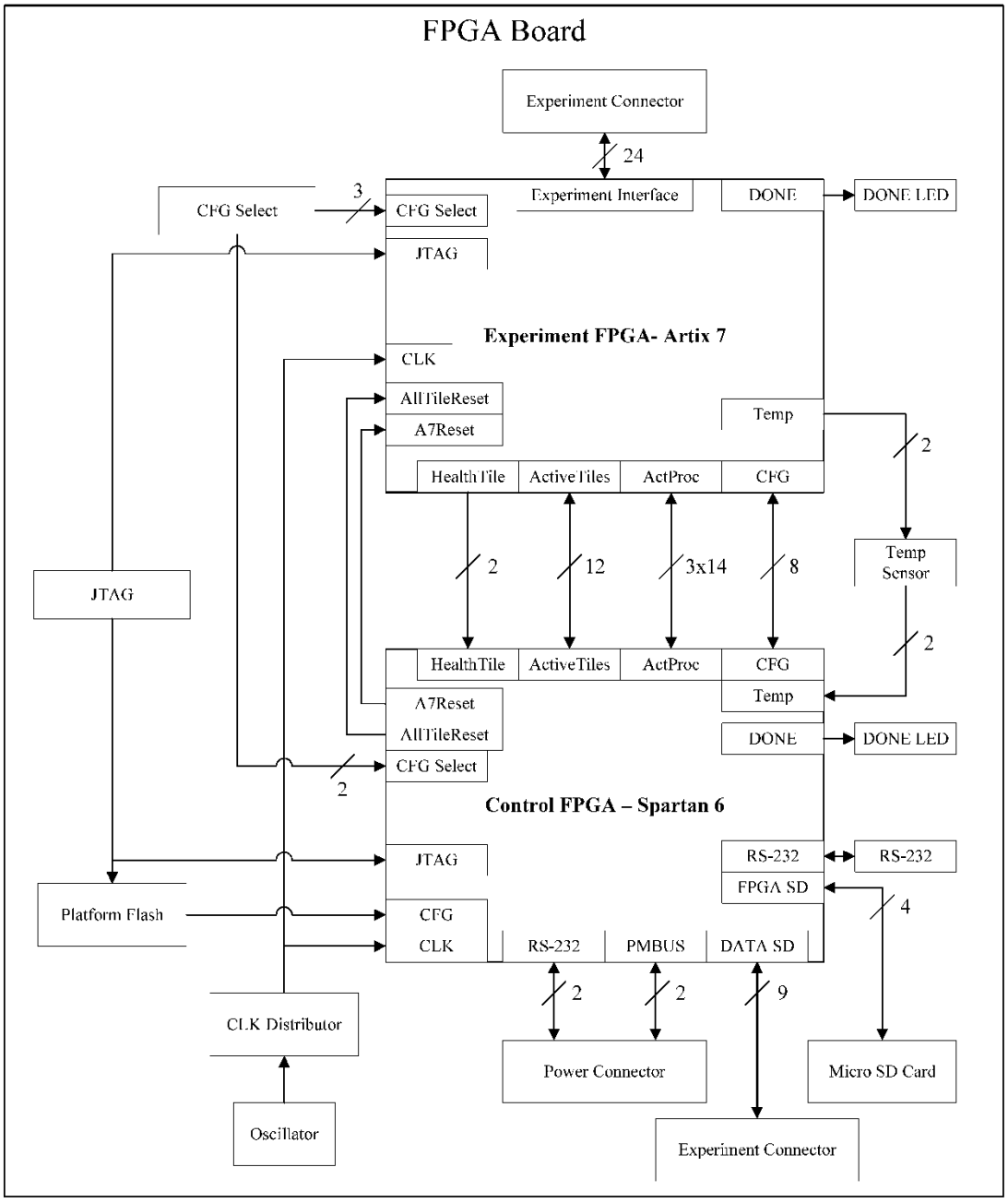

Figure 5-5. FPGA board block diagram.

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. This temperature sensor was chosen because it measures the junction temperature every five seconds and has a low power consumption, typically drawing 200 μA [35]. 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.

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 5-1). 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 is connected on top of the FPGA board with both the power and sensor connectors. The data board (Figure 5-6) 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.

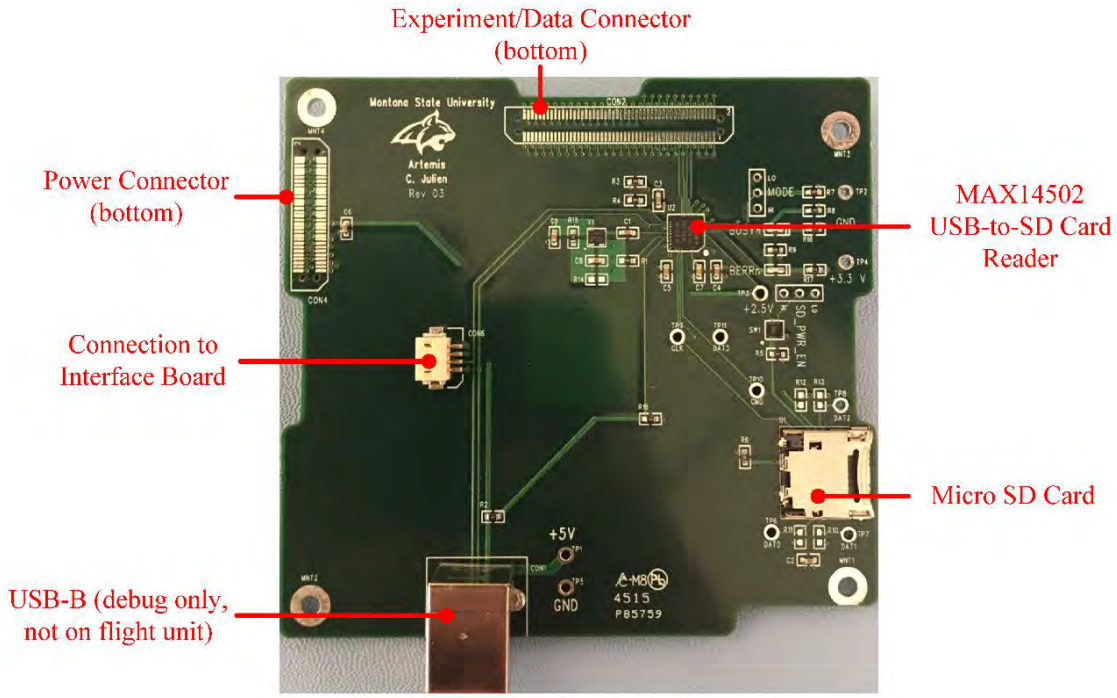

Figure 5-6. Data Board with important components labeled.

As previously mentioned, a 5 V 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.

Full Hardware Stack

The full Artemis hardware stack consists of the interface board, power board, FGPA board and data board stacked from bottom to top. The base plate of the chassis holds threaded rods in place that the boards are slipped onto and separated with spacers. The rods are secured to the chassis wall with standoffs. Figure 5-7 and Figure 5-8 below show the computer aided design (CAD) models while Figure 5-9 and Figure 5-10 show the Artemis flight unit.

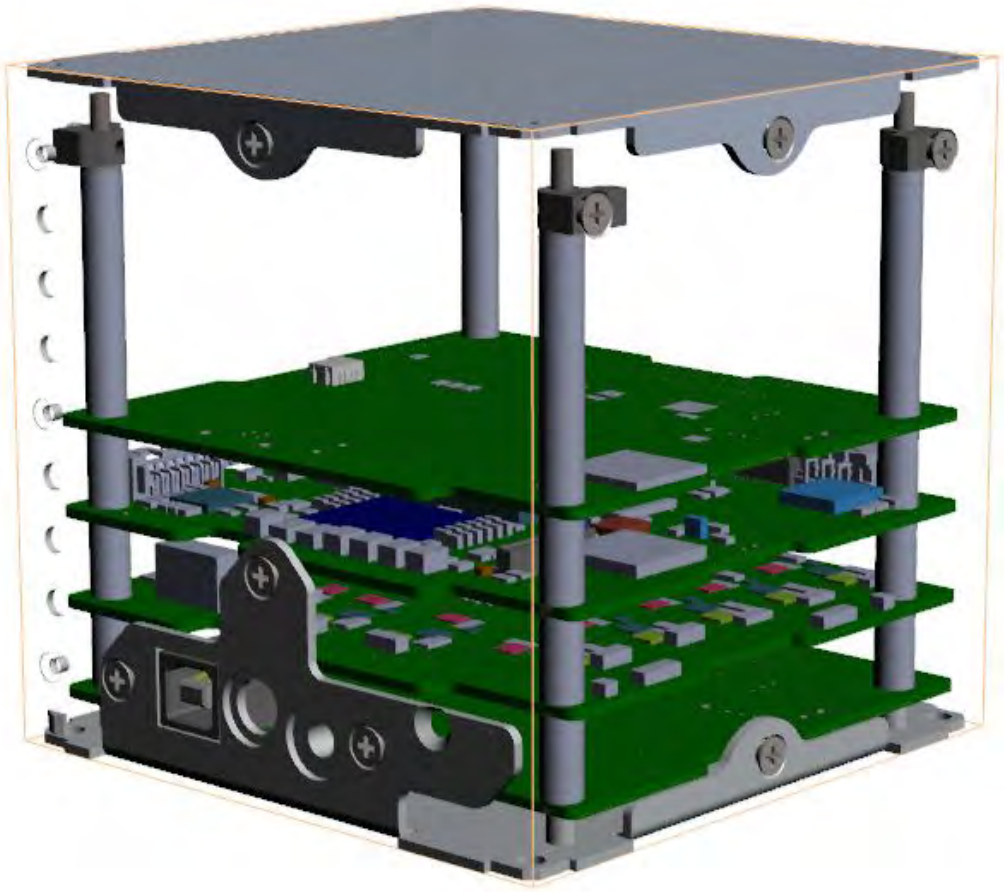

Figure 5-7. Artemis model with chassis side walls hidden.

Figure 5-8. Artemis model showing all chassis components in place.

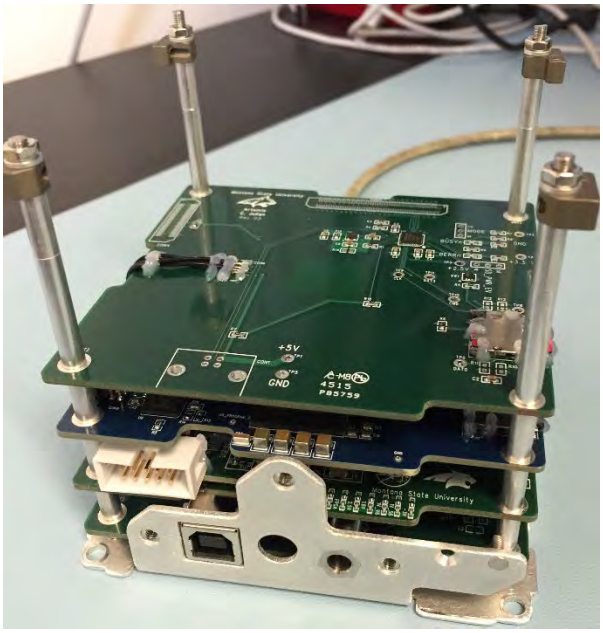

Figure 5-9. Artemis without the chassis side walls or top.

Figure 5-10. Artemis contained fully in the chassis.

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. The Spartan-6 digital design was designed before this thesis and the Artix-7 digital design was adapted and further developed for this thesis project.

Artix-7

The digital design for the experimental FPGA was first developed on a Virtex-4 FPGA. Most of the recent development has been performed on a Virtex-6 FPGA. To

meet the low power requirements on the ISS, the design was moved on to an Artix-7 FGPA. 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 5-11 is a tile that contains a MicroBlaze. Unlike Figure 3-1, this image shows the Artix-7 fully routed.

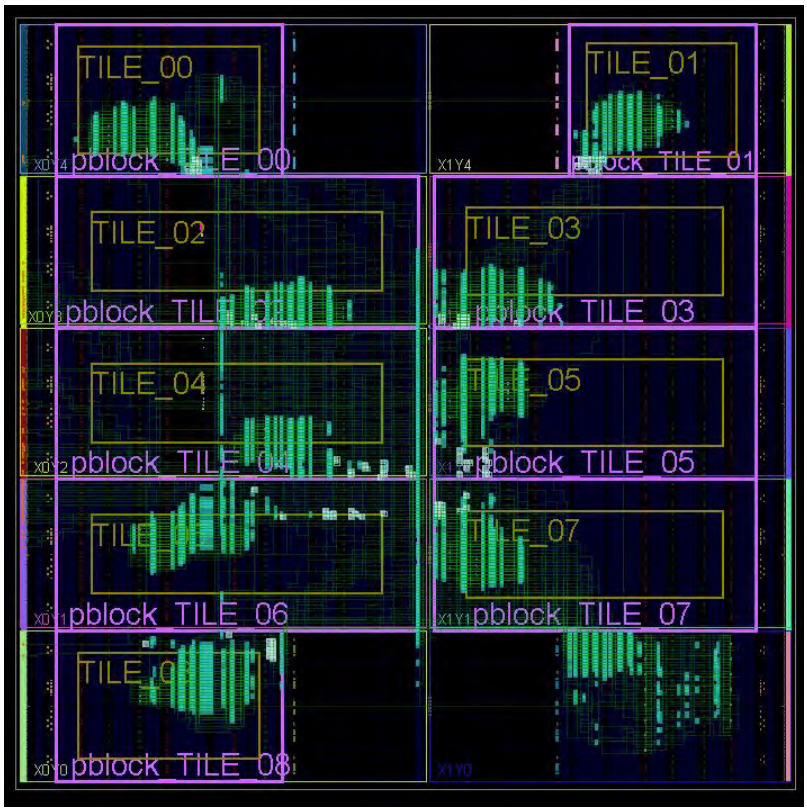

Figure 5-11. Artix-7 fully routed floorplan.

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 [36]. 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 each of the nine tiles.

Each of the nine tiles are simple VHDL components that contain a MicroBlaze soft processor (See Software). The output of each MicroBlaze tile runs to the multiplexer (MUX) component. An active tile signal from the Spartan-6 tells the MUX which three of the nine tiles are currently active. The MUX reads the active tile signal and selects only the output counts of the three active tiles. These three tile outputs are inputs to the TMR voter which actively checks the three counts for faults. The tile outputs are also routed to the Spartan-6. The TMR voter module compares outputs from the MUX 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.

Figure 5-12 shows the first part of the block diagram for the Artix-7 FPGA digital design. The left of this figure shows the inputs to the system, an all tile reset, the Artix-7 reset, the active tiles signal and the system clock. Figure 5-13 shows the 9 tiles each running a MicroBlaze, the multiplexer and the voter. Figure 5-14 once again shows the multiplexer and the voter with the outputs on the right.

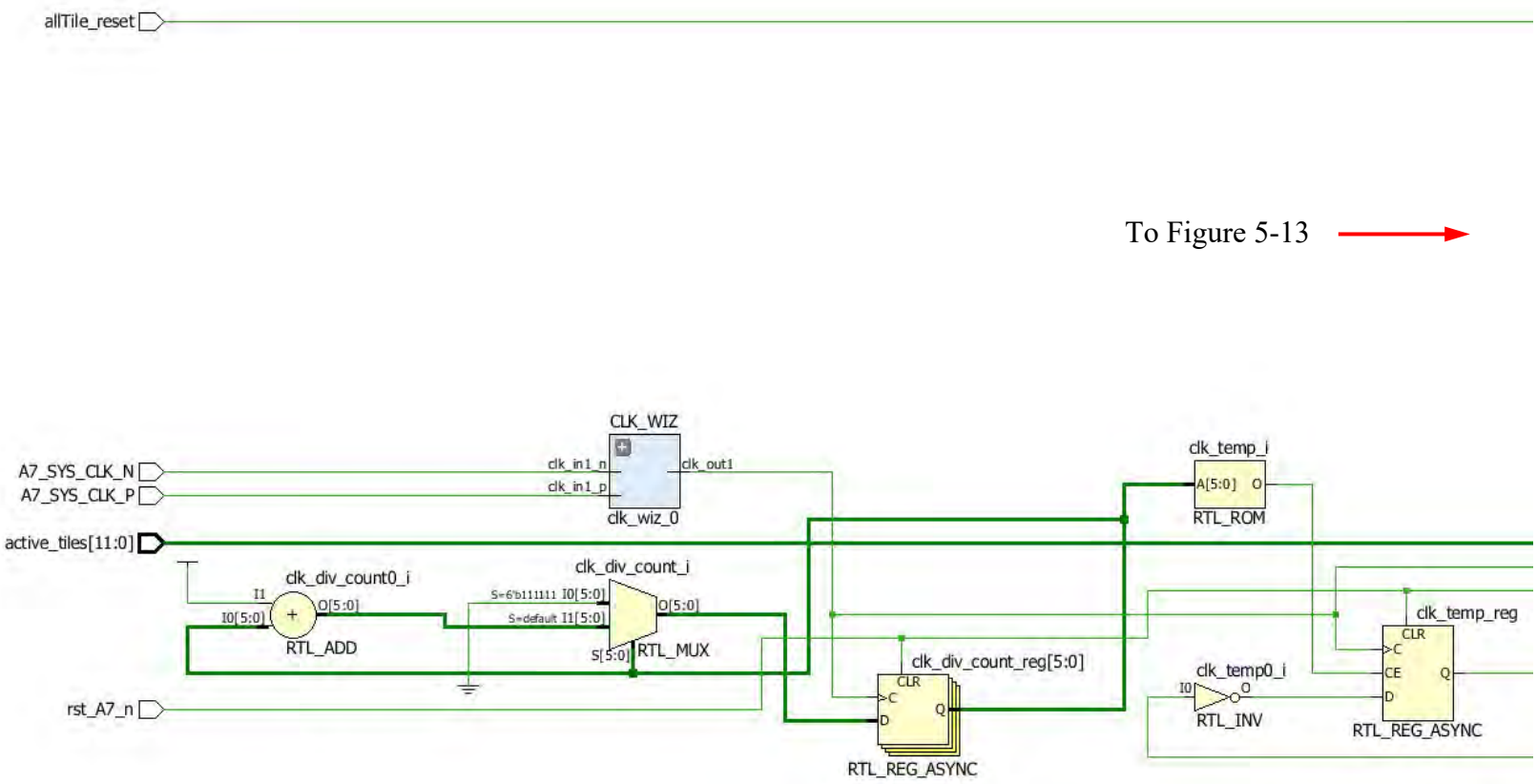

Figure 5-12. The left half of the Artix-7 block diagram showing the inputs on the left and the clock system in the middle.

To Figure 5-12

To Figure 5-14

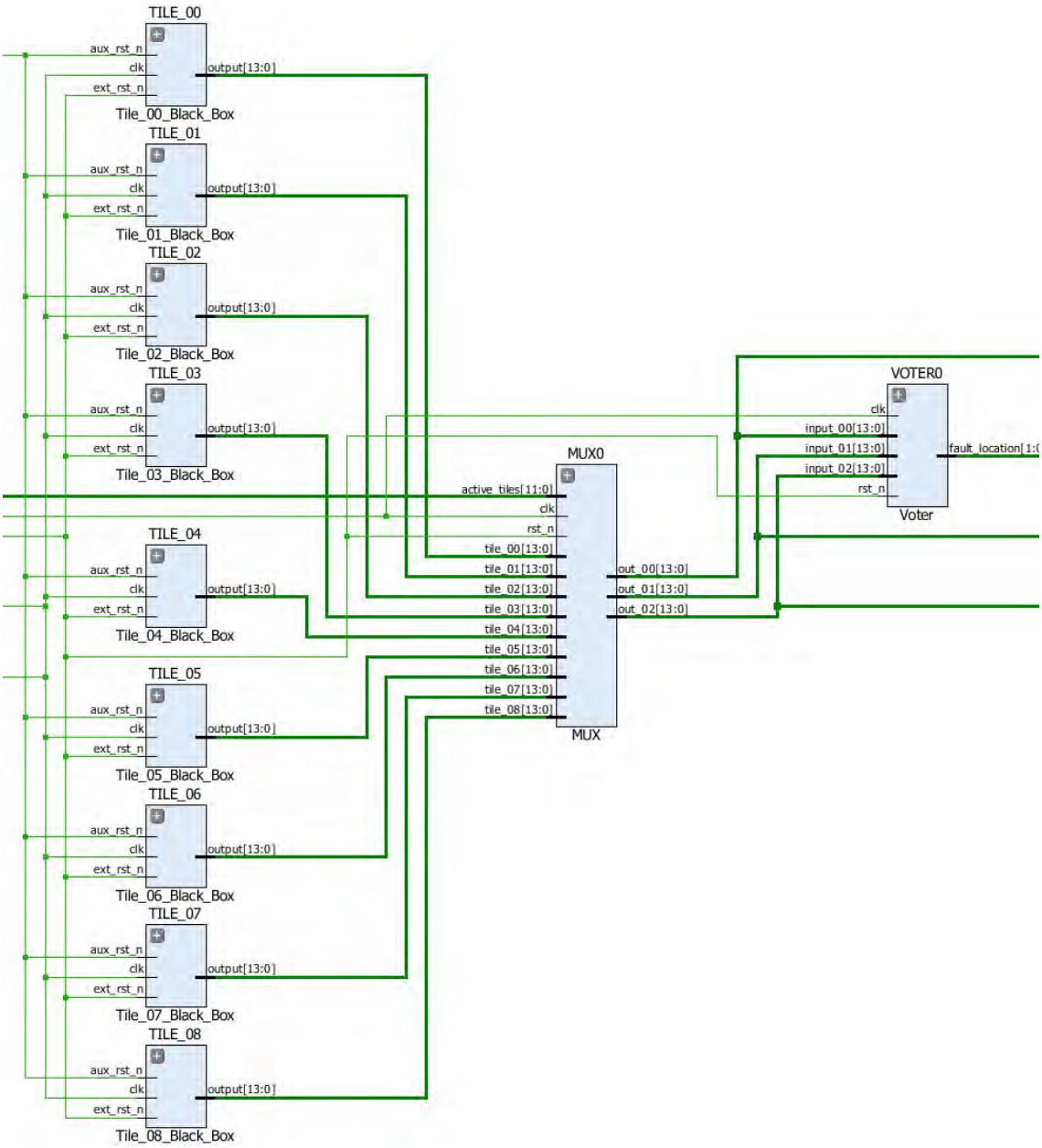

Figure 5-13. Middle portion of the Artix-7 block diagram showing the nine tiles, the MUX and the voter.

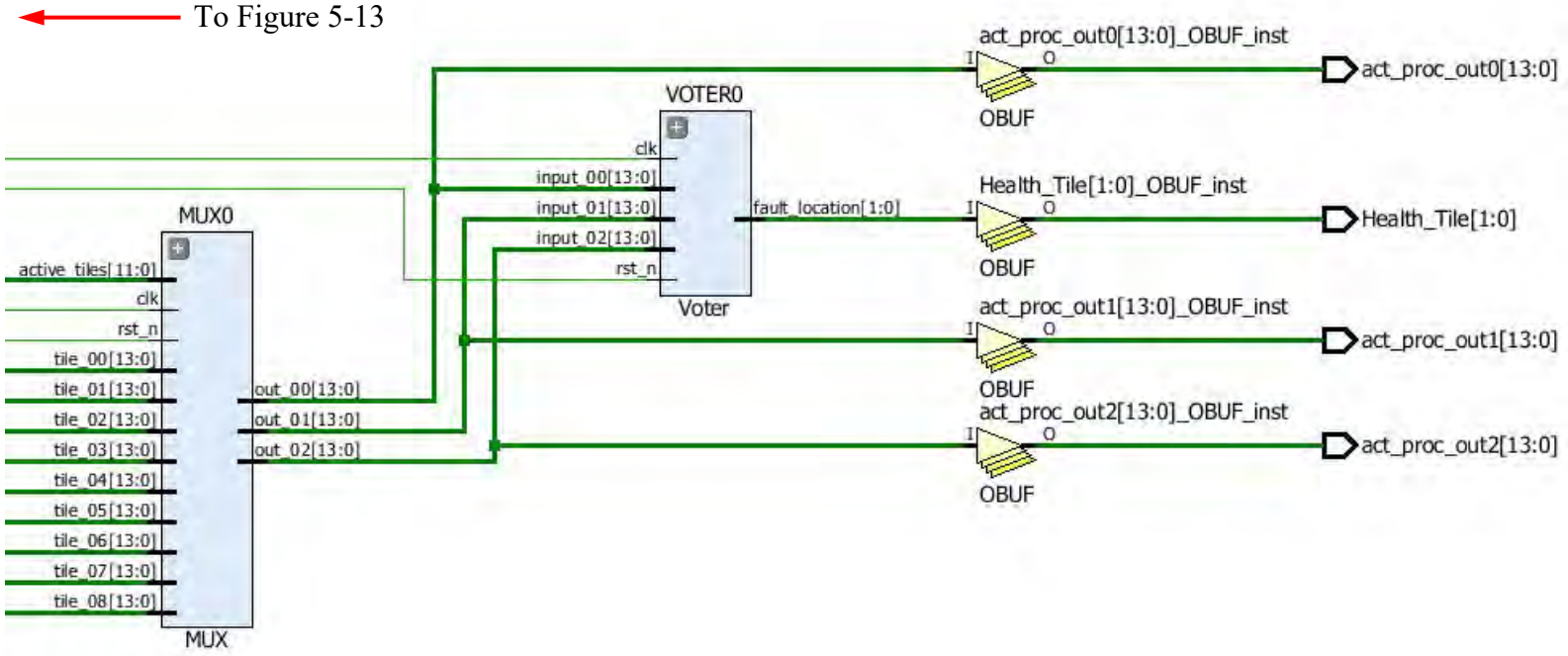

Figure 5-14. Right portion of the Artix-7 block diagram showing the MUX, voter and outputs.

New Reset System. Part of the development on the Artix-7 involved fixing a problem where the tiles would not have a synchronized count when the new tile was brought up. This would be viewed as a fault by the voter, which would mark the miscounting tile as faulty and bring on another new spare tile. The effect would cascade, eventually causing all the tiles to be faulted. The system would eventually hang because the tile repair through PR could not keep up with the tiles being marked faulty by the voter. Figure 5-15 below shows that the count values of the two existing tiles and the new tile aren't synchronized.

For the Artemis flight a new reset system was developed to synchronize the active tiles when a new tile is brought online. This allows operation to continue normally as the voter does not detect a fault. The new all tile reset is connected to each tile containing a MicroBlaze and is operated by ControlOS. When a new spare tile is needed due to a fault, all the tiles are first held in reset. Then the new tile is made active and the tiles are released from reset. This allows the active tiles to start on the same count and remain synchronized. Figure 5-16 below shows that the tiles are now synchronized after a reset occurs when a new tile is brought online. The final goal will be to initialize the new tile instead of having to reset the tiles (see Future Work).

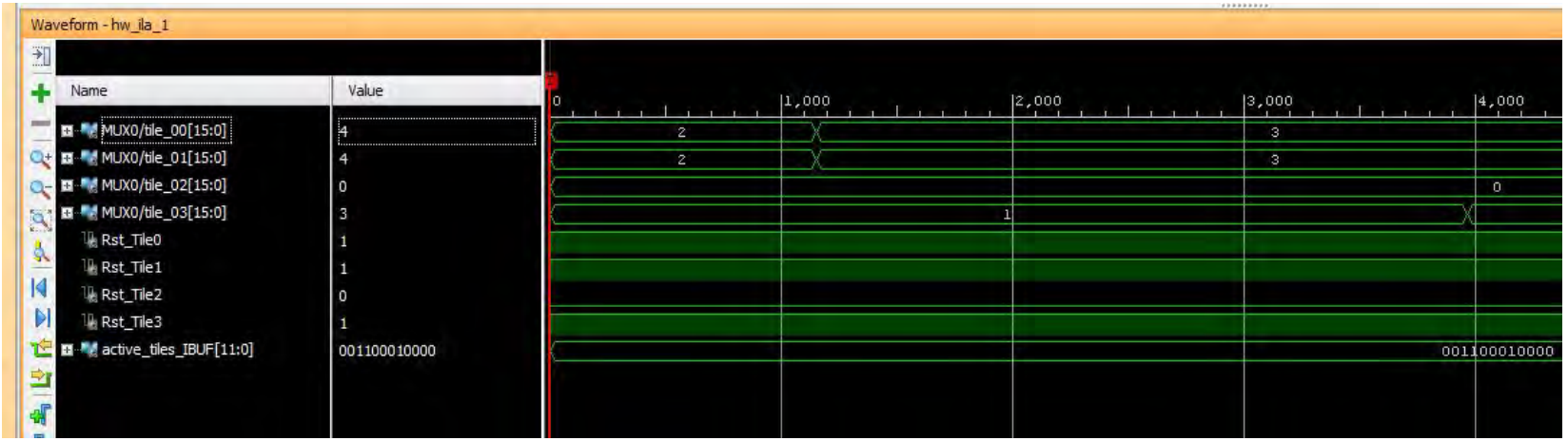

Figure 5-15. Shows the count of the new tile 3 possessing a different count than tile 0 and tile 1.

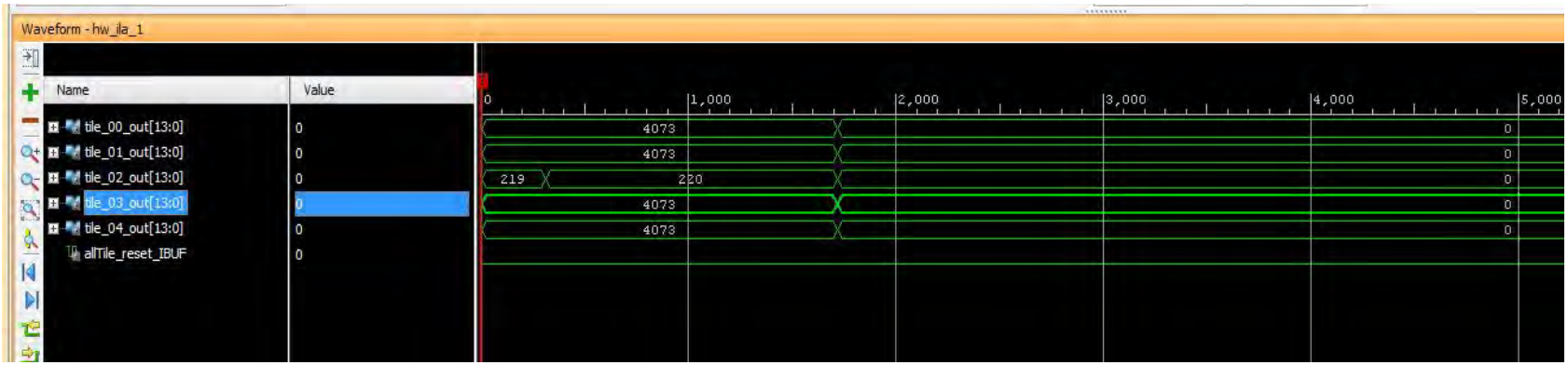

Figure 5-16. Internal Logic Analyzer signals showing the tiles synchronize after reset.

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 Spartan-6 is a single MicroBlaze soft processor dropped into the fabric that runs ControlOS, discussed in the Software section below. More about the Spartan-6 development can be found in Raymond Weber's PhD Dissertation [5] and Samuel Harkness's M.S. Thesis [3].

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, where the tools incorporate it into the FPGAs bitstream.

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 clock and two resets, an auxiliary reset and an external reset, are the only inputs. 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. Figure 5-17 shows the MicroBlaze setup with the input clock and resets on the left and the output count on the right.

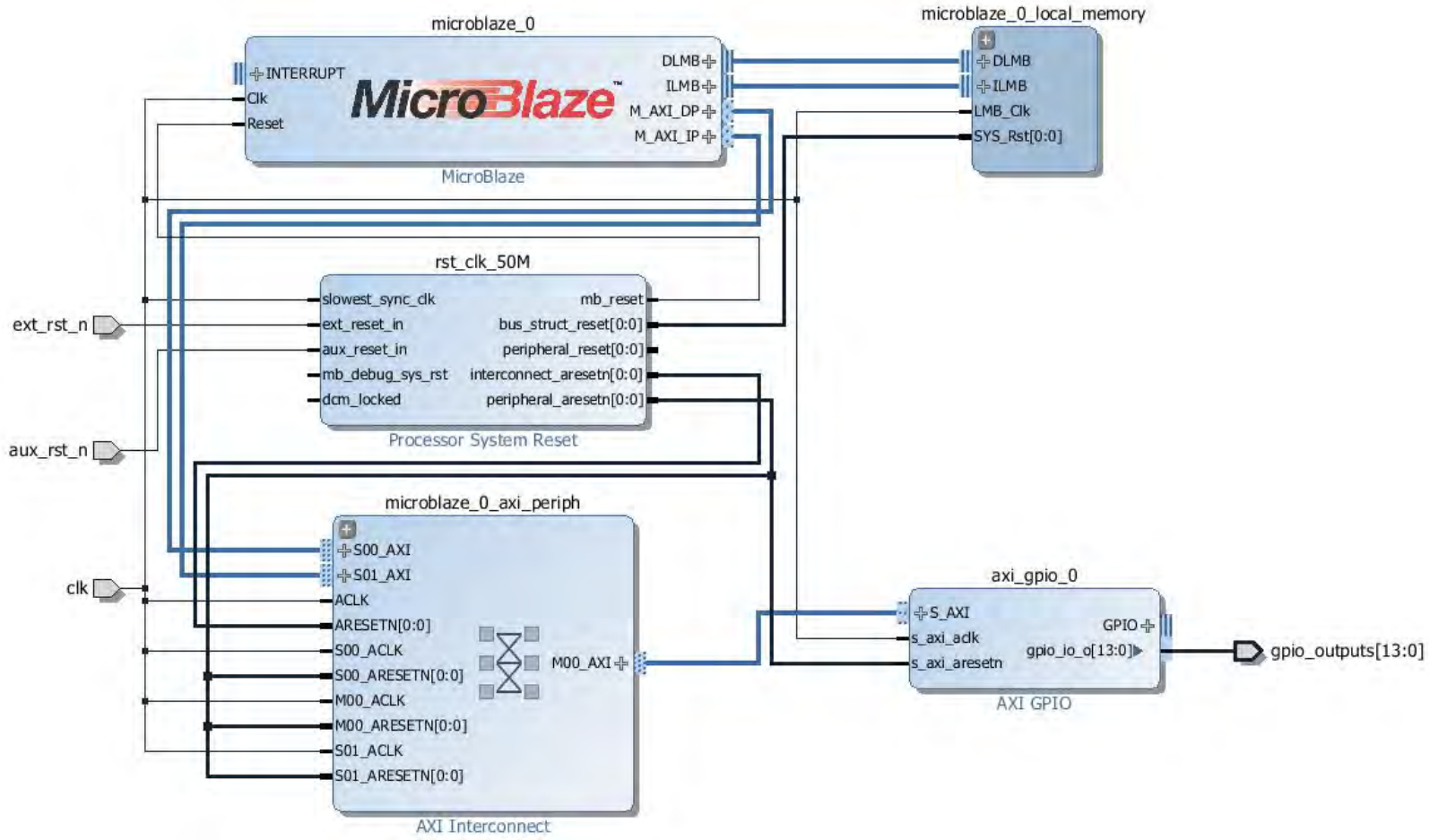

Figure 5-17. MicroBlaze design showing inputs on the left and the output on the right.

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. More about the Spartan-6 software and ControlOS can be found in Raymond Weber's PhD Dissertation [5] and Samuel Harkness's M.S. Thesis [3].

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 5-18). One of the last POST tasks is to perform the initial configuration of the Artix-7. Once the POST is passed, normal operation proceeds.

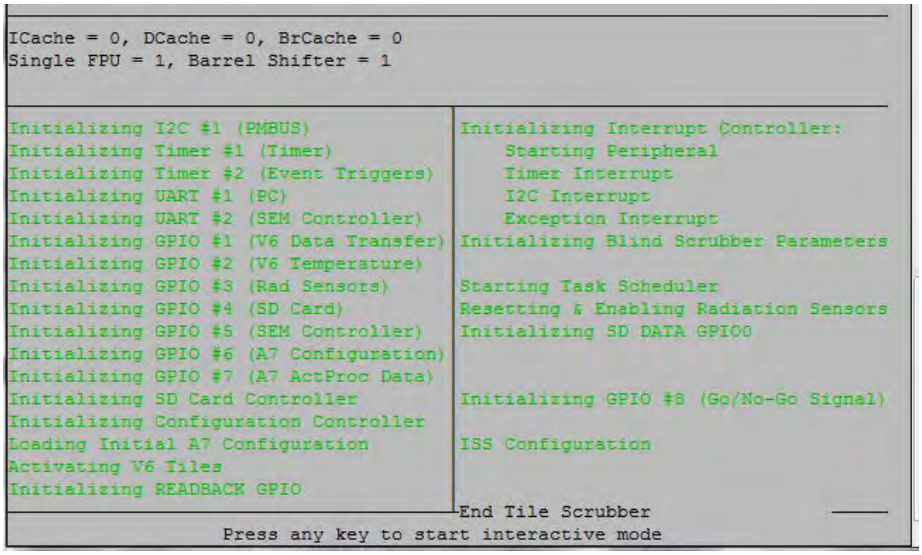

Figure 5-18. GUI POST screen.

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. Because Artemis was expected to see two to three radiation faults per day, it was determined that a data file will be written every twelve hours to capture that data. Fault injection was set to run every eleven hours so that there would be one injection per file write. The blind scrubber also runs once per file write. Move tile and repair tile occur consistently to catch and repair faulted tiles. Table 5-4 below shows each of the main ControlOS tasks and how often they are run for the Artemis mission.

Table 5-4. ControlOS tasks and how often they are performed for Artemis.

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 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 go high, lighting 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 5-19 shows the GUI screen during normal operation. In this case the first three tiles are active. Figure 5-20 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 partially reconfiguring with the same tile partial bitstream, a known corrupted bitstream is used to re-program the tile. For Artemis, bitstreams with counters that count down were used to inject faults. Upon injecting a

fault the bitstream would make the tile count backwards compared to the other active tiles. 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 5-5 shows the memory map of the SD card containing the clean initial and partial bitstreams as well as the corrupted full and partial bitstreams.

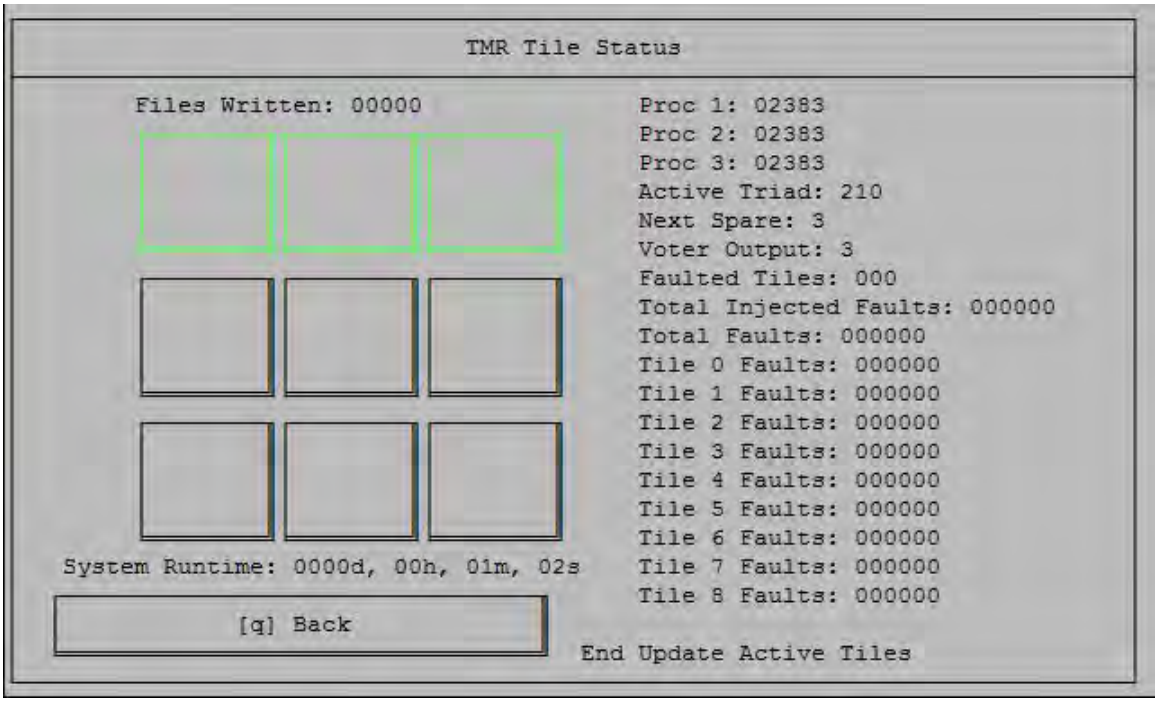

Figure 5-19. GUI tile status screen normal operation.

Figure 5-20. GUI screen with faulted tile.

Table 5-5. Bitstream start address and lengths stored on the SD card.

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

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 controls the accumulation of data, the operation of the data SD card on the data board as well as 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. After a file is written, 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 [3].

New Functionality. To stress each of the tile processors on the Artix-7, functionality was added to ControlOS to keep track of previously faulted tiles. ControlOS stores the last faulted tile and makes sure that the new spare tiles made active is the next successive tile in order. When the last three processors are the active processors, the system rolls over, making tile processor zero active during the next fault. This functionality is working as expected (See Results: Figure 7-9).

ARTEMIS ASSEMBLY AND TESTING

Flight Cleaning and Assembly

When a flight unit of Artemis was ready to be built, the PCBs had to be staked, cleaned and assembled for its mission to the ISS. Each of the PCBs were cleaned in three different baths. The first was a 3% concentration of Riptide. The second bath was in 95% ethanol. The third bath was in 95% isopropyl alcohol. In each bath, the PCBs were scrubbed with horse-hair brushes and then dried in a convection oven.

The vibration profile is insignificant for the Artemis mission because it is soft-stowed with other payloads in a pressurized cargo hold of the resupply vehicle. Despite this fact, it was deemed necessary to stake or epoxy large and major components of the system to ensure durability over the lifetime of the project. To secure each component more reliably to the PCB, a polyurethane staking compound called Arathane 5753 from Huntsman Inc. was utilized. Small amounts of this compound were applied to components that have high profiles and to components with moving parts such as connectors or SD card holders.

After component staking was complete, Artemis was assembled based on its model. Figure 6-1 below shows the cleaning, staking and assembly process. More information about this cleaning and assembly process can be found in the Artemis shop order document (ART-SO-0504) through the SSEL [37].

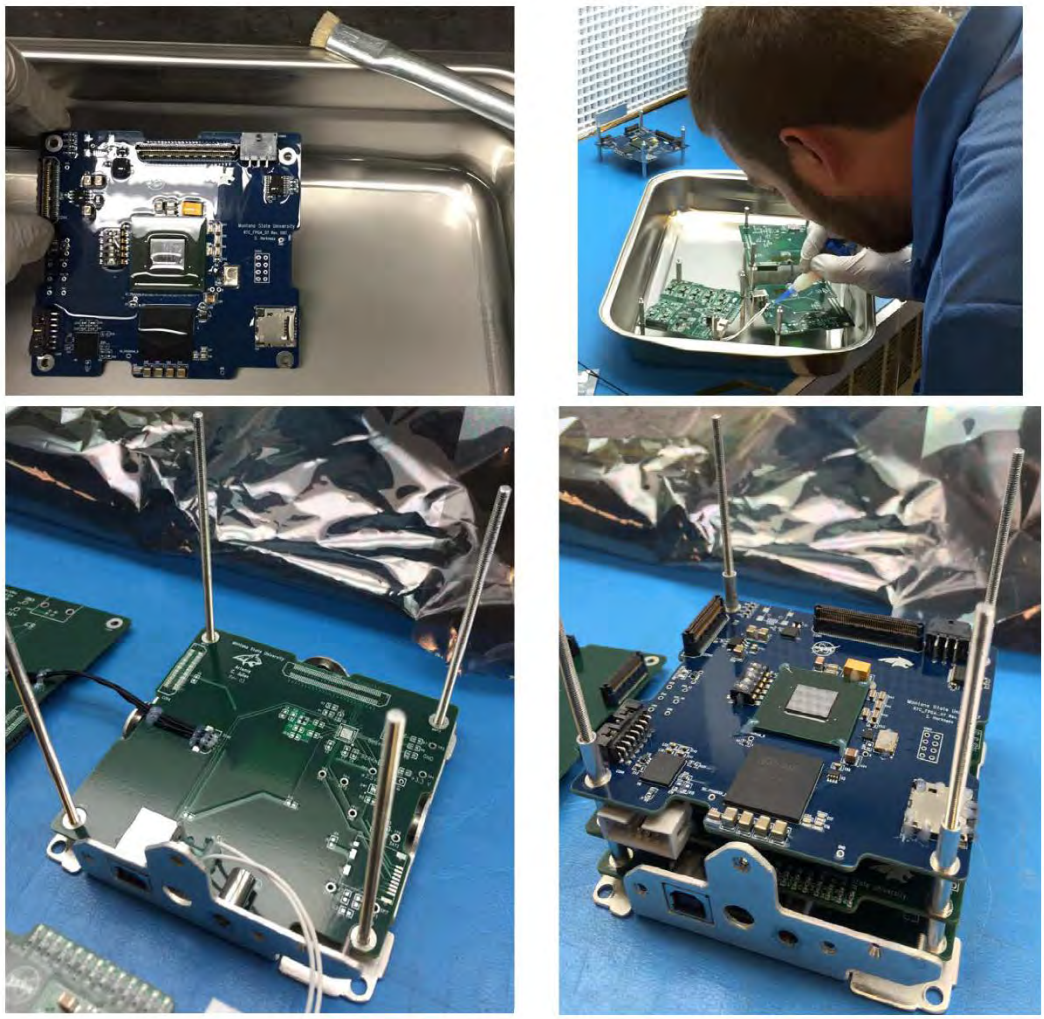

Figure 6-1. Artemis board cleaning (top-left), staking (top-right), and assembly (bottom).

Artemis Duration, Thermal and Functional Testing

Large amounts of testing were performed in the weeks leading up to the Artemis flight unit delivery. Duration, thermal and functional tests were performed. 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 transferred to the Artemis flight unit to be further tested and the cycle was repeated.

Figure 6-2 shows the typical Artemis data parser. This Visual Basic program reads the Artemis data from the text files contained in an input folder and displays selected information for the viewer. On the left, general information like the system counter, the system runtime, number of faults, active tiles, power information and the tile status can be observed. The three graphs in the middle show the system counter, the power information and the temperature information. Drop down menus and toggles can be used to select exactly what data the view wants to look at. A play/pause button allows the parser to cycle through each file, showing each data point.

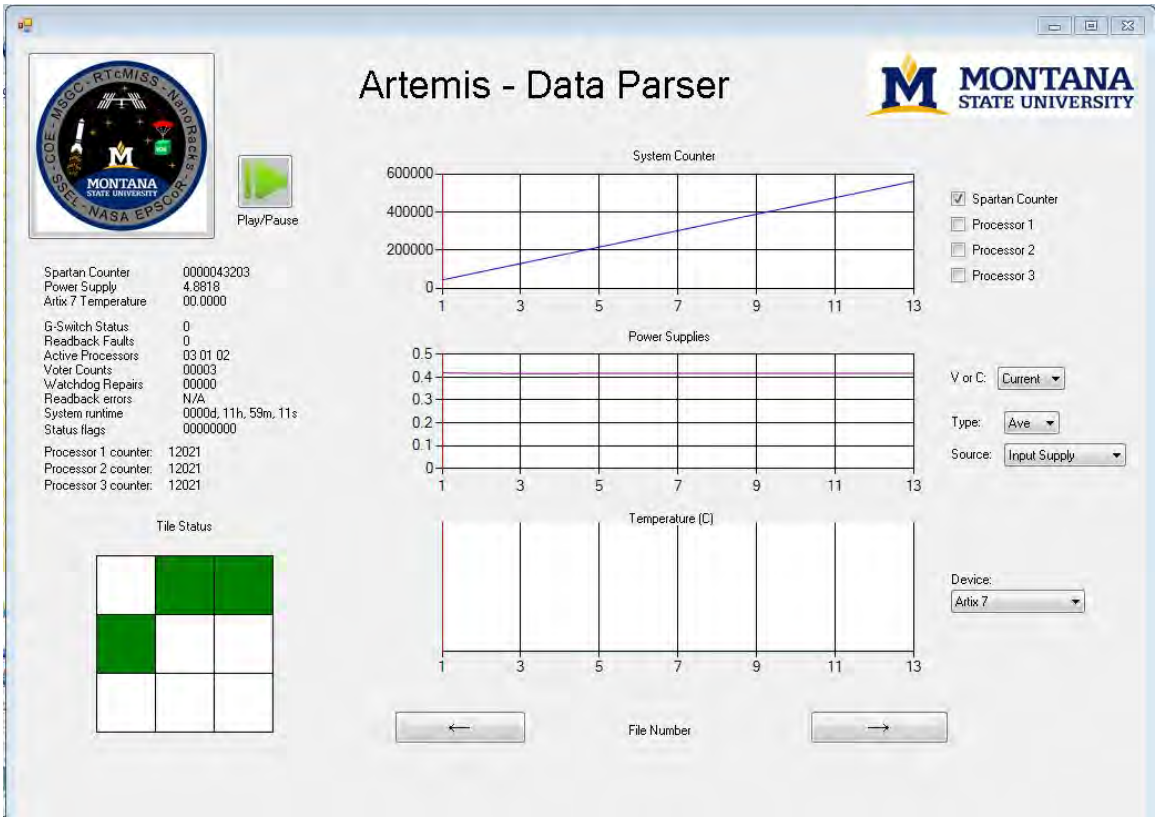

Figure 6-2. Duration test data showing A7 temperature bug and Artemis current draw.

The data parser provides a clean way to check the Artemis data and visualize its operation over the number of files written. However, Artemis data is also collected and viewed graphically using Excel. This allows for more flexibility when producing results for displays and publications.

Duration Testing. Duration testing of the Artemis system was performed in the weeks before delivering the flight unit. Figure 6-2 above shows the Spartan-6 system counter increasing linearly showing the system was running continuously throughout the test.

Thermal Testing. Thermal testing of the experimental FPGA was also performed. A bug was found to show that the Artix-7 temperature was not printing properly in the data text files (shown in Figure 6-2 above). Further investigation lead to find 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. In Figure 6-3 below, the temperature of power controller 1 is increasing throughout the thermal test.

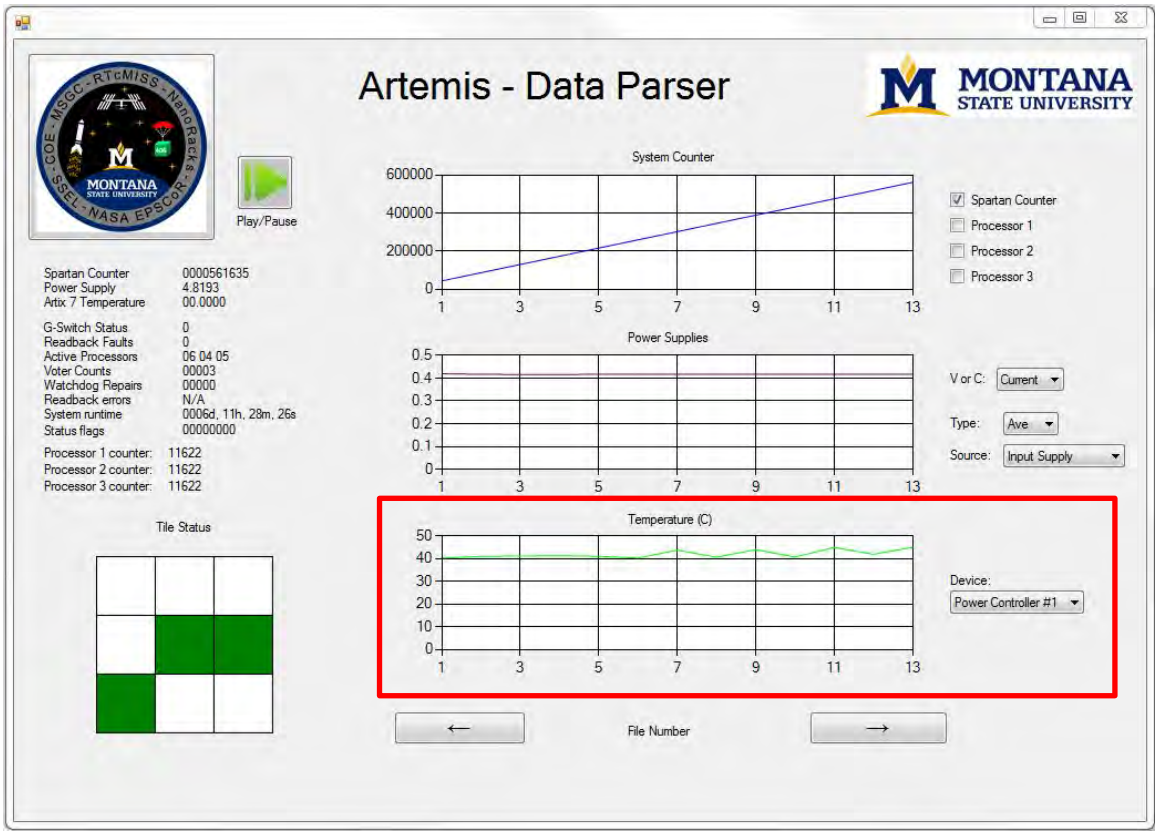

Figure 6-3. Graph of temperature of PC during thermal test.

Current Draw. 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.

Functional Test. Upon delivering the flight hardware to NanoRacks in Houston, a functional test was performed. This test involved plugging Artemis into a mockup of the rack system that they have on the ISS. This connection allowed the system to be viewed as a USB device exactly like it will on the station. Figure 6-4 shows the data from this functional test. NanoRacks left Artemis plugged in for two full days allowing the system to write four files (Figure 6-5).

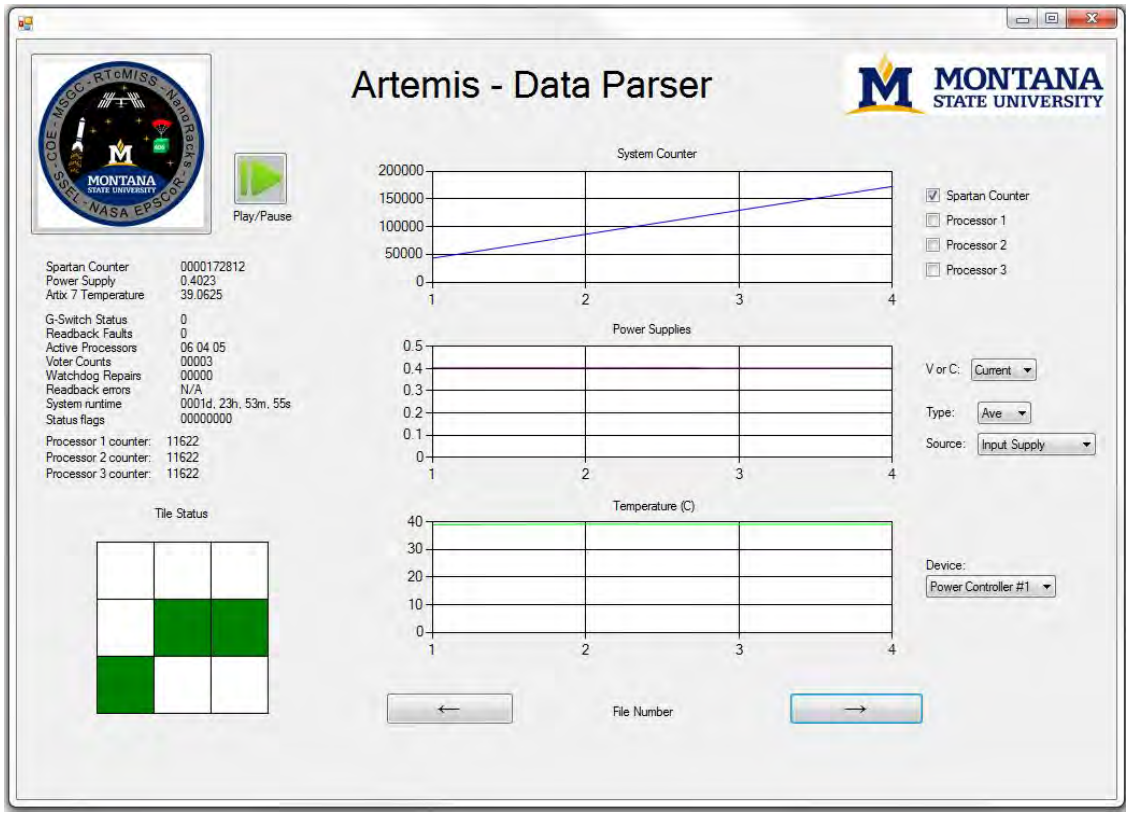

Figure 6-4. Data from the functional test at NanoRacks in Houston, Texas.

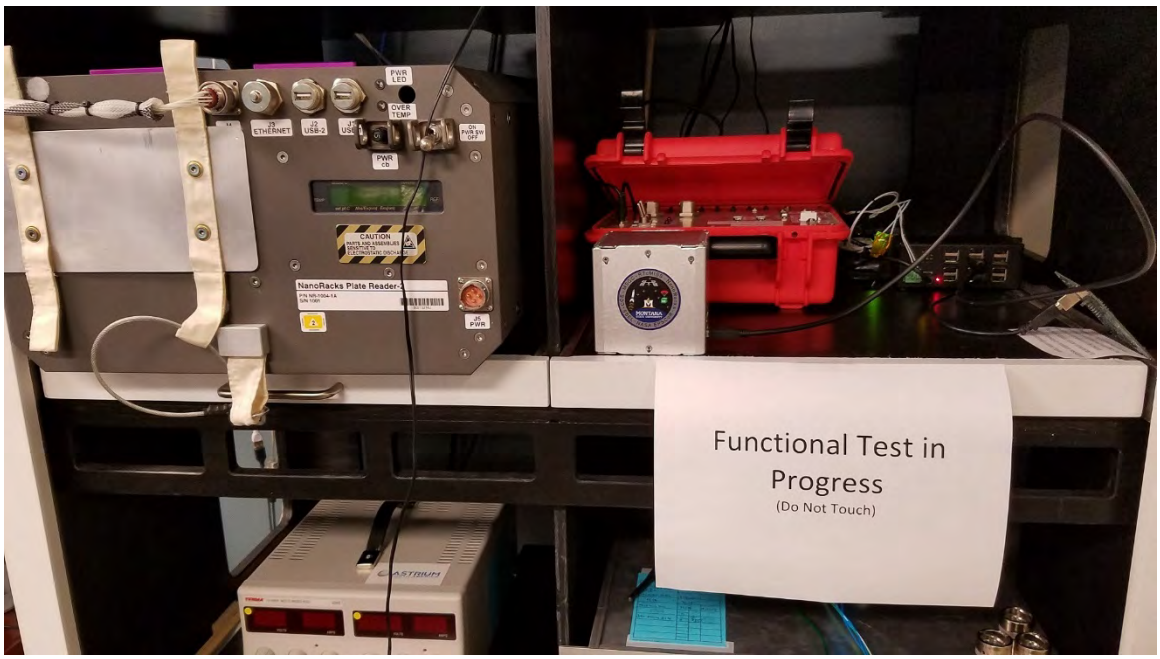

Figure 6-5. Artemis during functional test at NanoRacks in Houston, Texas.

RESULTS

Data from Artemis on the ISS is displayed in this chapter. System health data including power rail voltages and currents, as well as system temperatures are presented first. Operation and functionality data is then presented. Data for the first sixty days (1440 hours) of operation are presented in this thesis as that was all the data available at the time. At the end of the data sets a discussion section presents views on the collected data. An example data file written by Artemis can be found in Appendix A.

Voltage Data

Voltage data is collected by the T.I. power controllers on the power board. Data is streamed over PMBUS to the Spartan-6 which stores it to be printed as part of the TLM_HEALTH packet in a data file (see Packet Structure: Appendix B).

Table 7-1 shows a breakdown of each power rail name, description and the figure number the voltage data can be found in. All data is plotted against the system runtime in hours. Figure 7-1 shows the input voltage rail. The input from NanoRacks is stated to be 5 V so Artemis is receiving a slightly higher voltage than expected. Artemis can accept a range of 4.85 V to 7.28 V. Figure 7-2 shows an example of a voltage rail produced by the power board. This voltage is the core Artix-7 FPGA voltage and should be 0.95 V. The acceptable range for this rail is 0.92 V to 0.98 V as described in the Artix-7 datasheet [33]. Voltage data for the rest of the power rails can be found in Appendix D.

Table 7-1. Table containing mnemonics, descriptions and data figure numbers for the voltage of each power rail.

Mnemonic	Description	Figure Number
VOLTAGE_INS_BATT	Instantaneous input rail	Figure 7-1
VOLTAGE_AVE_BATT	Average input rail	Figure 7-1
VOLTAGE_MAX_BATT	Maximum input rail	Figure 7-1
VOLTAGE_MIN_BATT	Minimum input rail	Figure 7-1
VOLTAGE_INS_15V0A	Instantaneous 15 V analog rail	Figure 11-1
VOLTAGE_AVE_15V0A	Average 15 V analog rail	Figure 11-1
VOLTAGE_MAX_15V0A	Maximum 15 V analog rail	Figure 11-1
VOLTAGE_MIN_15V0A	Minimum 15 V analog rail	Figure 11-1
VOLTAGE_INS_N3V0A	Instantaneous -3 V analog rail	Figure 11-2
VOLTAGE_AVE_N3V0A	Average -3 V analog rail	Figure 11-2
VOLTAGE_MAX_N3V0A	Maximum -3 V analog rail	Figure 11-2
VOLTAGE_MIN_N3V0A	Minimum -3 V analog rail	Figure 11-2
VOLTAGE_INS_3V3D	Instantaneous 3.3 V digital rail	Figure 11-3
VOLTAGE_AVE_3V3D	Average 3.3 V digital rail	Figure 11-3
VOLTAGE_MAX_3V3D	Maximum 3.3 V digital rail	Figure 11-3
VOLTAGE_MIN_3V3D	Minimum 3.3 V digital rail	Figure 11-3
VOLTAGE_INS_3V0A	Instantaneous 3.0 V analog rail	Figure 11-4
VOLTAGE_AVE_3V0A	Average 3.0 V analog rail	Figure 11-4
VOLTAGE_MAX_3V0A	Maximum 3.0 V analog rail	Figure 11-4
VOLTAGE_MIN_3V0A	Minimum 3.0 V analog rail	Figure 11-4
VOLTAGE_INS_2V5D	Instantaneous 2.5 V digital rail	Figure 11-5
VOLTAGE_AVE_2V5D	Average 2.5 V digital rail	Figure 11-5
VOLTAGE_MAX_2V5D	Maximum 2.5 V digital rail	Figure 11-5
VOLTAGE_MIN_2V5D	Minimum 2.5 V digital rail	Figure 11-5
VOLTAGE_INS_2V5FA	Instantaneous 2.5 V analog rail	Figure 11-6
VOLTAGE_AVE_2V5FA	Average 2.5 V analog rail	Figure 11-6
VOLTAGE_MAX_2V5FA	Maximum 2.5 V analog rail	Figure 11-6
VOLTAGE_MIN_2V5FA	Minimum 2.5 V analog rail	Figure 11-6
VOLTAGE_INS_2V5RA	Instantaneous 2.5 V analog rail	Figure 11-7
VOLTAGE_AVE_2V5RA	Average 2.5 V analog rail	Figure 11-7
VOLTAGE_MAX_2V5RA	Maximum 2.5 V analog rail	Figure 11-7
VOLTAGE_MIN_2V5RA	Minimum 2.5 V analog rail	Figure 11-7
VOLTAGE_INS_1V8D	Instantaneous 1.8 V digital rail	Figure 11-8
VOLTAGE_AVE_1V8D	Average 1.8 V digital rail	Figure 11-8

Mnemonic	Description	Figure Number
VOLTAGE_MAX_1V8D	Maximum 1.8 V digital rail	Figure 11-8
VOLTAGE_MIN_1V8D	Minimum 1.8 V digital rail	Figure 11-8
VOLTAGE_INS_1V0SD	Instantaneous 1.0 V digital rail	Figure 11-9
VOLTAGE_AVE_1V0SD	Average 1.0 V digital rail	Figure 11-9
VOLTAGE_MAX_1V0SD	Maximum 1.0 V digital rail	Figure 11-9
VOLTAGE_MIN_1V0SD	Minimum 1.0 V digital rail	Figure 11-9
VOLTAGE_INS_0V95AD	Instantaneous 0.95 V digital rail	Figure 7-2
VOLTAGE_AVE_0V95AD	Average 0.95 V digital rail	Figure 7-2
VOLTAGE_MAX_0V95AD	Maximum 0.95 V digital rail	Figure 7-2
VOLTAGE_MIN_0V95AD	Minimum 0.95 V digital rail	Figure 7-2

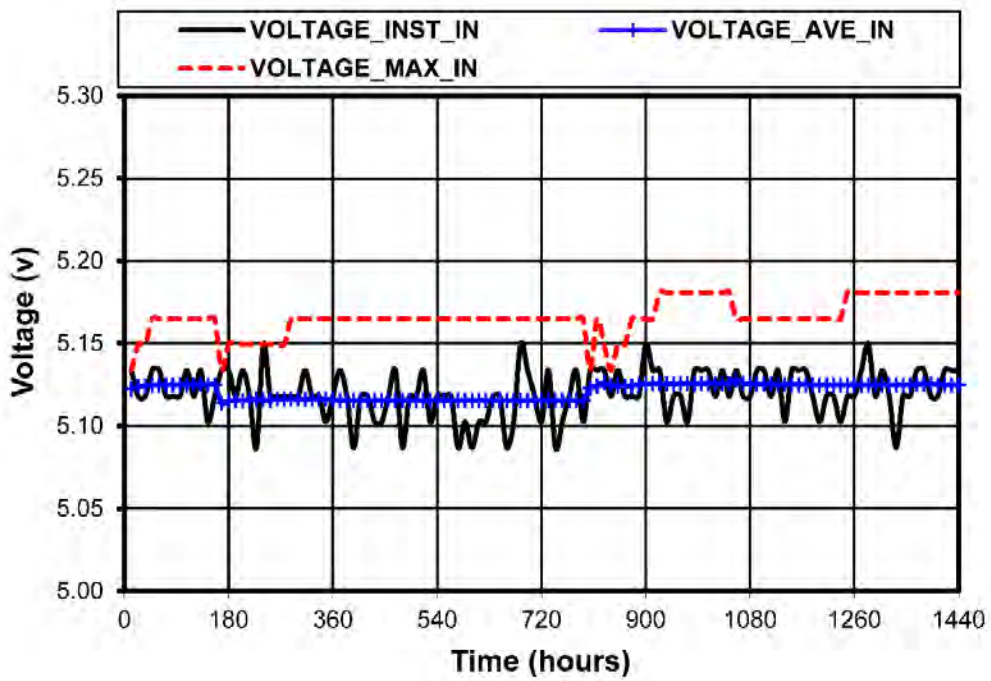

Figure 7-1. Input instantaneous, average and maximum voltage (Nominal = 5 V, Max = 7.3 V, Min = 4.85 V).

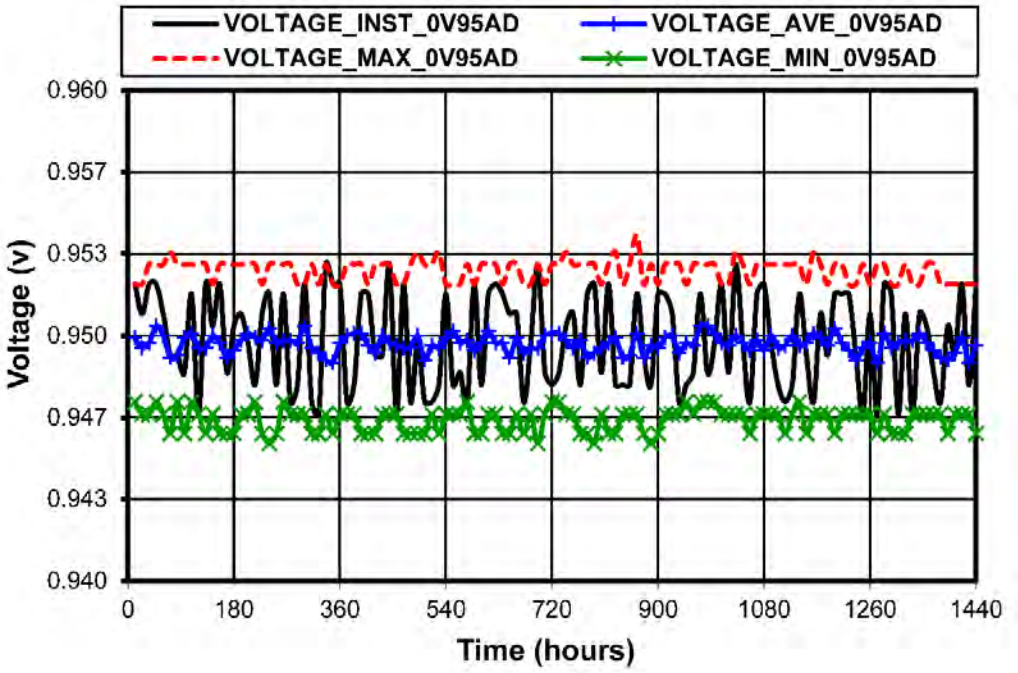

Figure 7-2. Digital 0.95 V rail instantaneous, average, maximum and minimum voltage (Nominal = 0.95 V, Max = 0.92 V, Min = 0.98 V).

Current Data

Current data is collected by the T.I. power controllers on the power board. Data is streamed over PMBUS to the Spartan-6 which stores and prints the data in a data file. The current data is contained in the TLM_HEALTH packet of the data directly after the voltage data (see Packet Structure: Appendix B). Table 7-2 shows a breakdown of each power rail name, description and the figure number the current data can be found in. Figure 7-3 shows the input current to Artemis from the NanoRacks platform. Ground testing showed a nominal current draw was around 400 mA. Figure 7-4 shows the current draw for one of the power rails (2.5 V digital). The graphs show the current data plotted versus system runtime. Current data for the rest of the power rails can be found in Appendix D.

Table 7-2. Table containing mnemonics, descriptions and data figure numbers for the current of each power rail.

Mnemonic	Description	Figure Number
CURRENT_INS_BATT	Instantaneous input rail	Figure 7-3
CURRENT_AVE_BATT	Average input rail	Figure 7-3
CURRENT_MAX_BATT	Maximum input rail	Figure 7-3
CURRENT_MIN_BATT	Minimum input rail	Figure 7-3
CURRENT_INS_15V0A	Instantaneous 15 V analog rail	Figure 11-10
CURRENT_AVE_15V0A	Average 15 V analog rail	Figure 11-10
CURRENT_MAX_15V0A	Maximum 15 V analog rail	Figure 11-10
CURRENT_MIN_15V0A	Minimum 15 V analog rail	Figure 11-10
CURRENT_INS_N3V0A	Instantaneous -3 V analog rail	Figure 11-11
CURRENT_AVE_N3V0A	Average -3 V analog rail	Figure 11-11
CURRENT_MAX_N3V0A	Maximum -3 V analog rail	Figure 11-11
CURRENT_MIN_N3V0A	Minimum -3 V analog rail	Figure 11-11
CURRENT_INS_3V3D	Instantaneous 3.3 V digital rail	Figure 11-12
CURRENT_AVE_3V3D	Average 3.3 V digital rail	Figure 11-12

Mnemonic	Description	Figure Number
CURRENT_MAX_3V3D	Maximum 3.3 V digital rail	Figure 11-12
CURRENT_MIN_3V3D	Minimum 3.3 V digital rail	Figure 11-12
CURRENT_INS_3V0A	Instantaneous 3.0 V analog rail	Figure 11-13
CURRENT_AVE_3V0A	Average 3.0 V analog rail	Figure 11-13
CURRENT_MAX_3V0A	Maximum 3.0 V analog rail	Figure 11-13
CURRENT_MIN_3V0A	Minimum 3.0 V analog rail	Figure 11-13
CURRENT_INS_2V5D	Instantaneous 2.5 V digital rail	Figure 7-4
CURRENT_AVE_2V5D	Average 2.5 V digital rail	Figure 7-4
CURRENT_MAX_2V5D	Maximum 2.5 V digital rail	Figure 7-4
CURRENT_MIN_2V5D	Minimum 2.5 V digital rail	Figure 7-4
CURRENT_INS_2V5FA	Instantaneous 2.5 V analog rail	Figure 11-14
CURRENT_AVE_2V5FA	Average 2.5 V analog rail	Figure 11-14
CURRENT_MAX_2V5FA	Maximum 2.5 V analog rail	Figure 11-14
CURRENT_MIN_2V5FA	Minimum 2.5 V analog rail	Figure 11-14
CURRENT_INS_2V5RA	Instantaneous 2.5 V analog rail	Figure 11-15
CURRENT_AVE_2V5RA	Average 2.5 V analog rail	Figure 11-15
CURRENT_MAX_2V5RA	Maximum 2.5 V analog rail	Figure 11-15
CURRENT_MIN_2V5RA	Minimum 2.5 V analog rail	Figure 11-15
CURRENT_INS_1V8D	Instantaneous 1.8 V digital rail	Figure 11-16
CURRENT_AVE_1V8D	Average 1.8 V digital rail	Figure 11-16
CURRENT_MAX_1V8D	Maximum 1.8 V digital rail	Figure 11-16
CURRENT_MIN_1V8D	Minimum 1.8 V digital rail	Figure 11-16
CURRENT_INS_1V0SD	Instantaneous 1.0 V digital rail	Figure 11-17
CURRENT_AVE_1V0SD	Average 1.0 V digital rail	Figure 11-17
CURRENT_MAX_1V0SD	Maximum 1.0 V digital rail	Figure 11-17
CURRENT_MIN_1V0SD	Minimum 1.0 V digital rail	Figure 11-17
CURRENT_INS_0V95AD	Instantaneous 0.95 V digital rail	Figure 11-18
CURRENT_AVE_0V95AD	Average 0.95 V digital rail	Figure 11-18
CURRENT_MAX_0V95AD	Maximum 0.95 V digital rail	Figure 11-18
CURRENT_MIN_0V95AD	Minimum 0.95 V digital rail	Figure 11-18

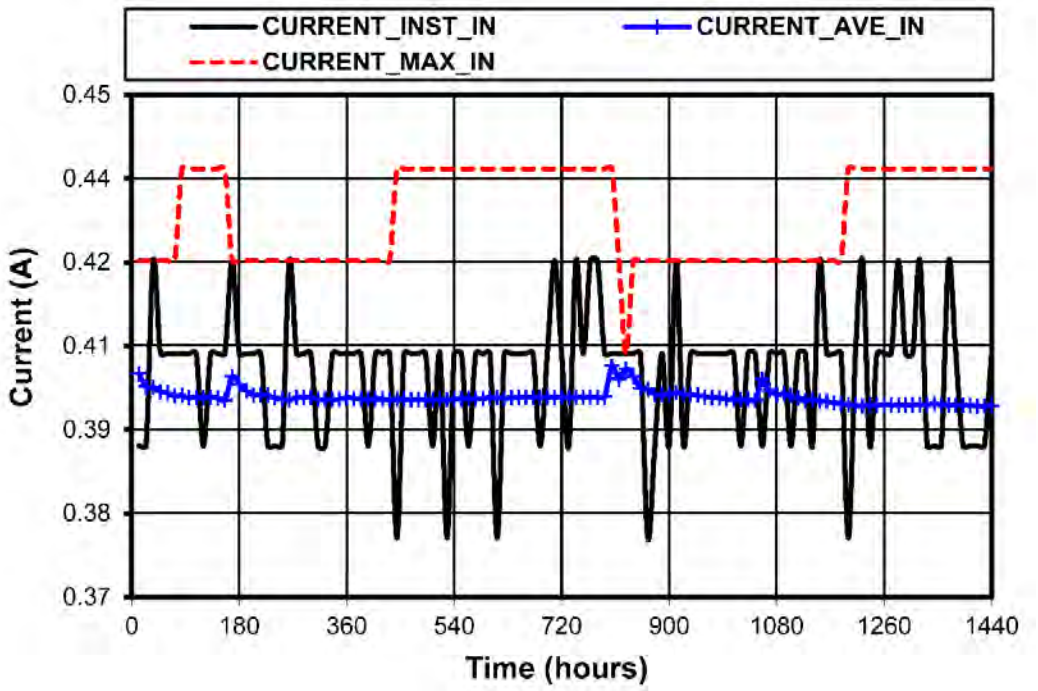

Figure 7-3. Input rail instantaneous, average and maximum current (Range: 0 – 0.4375 A).

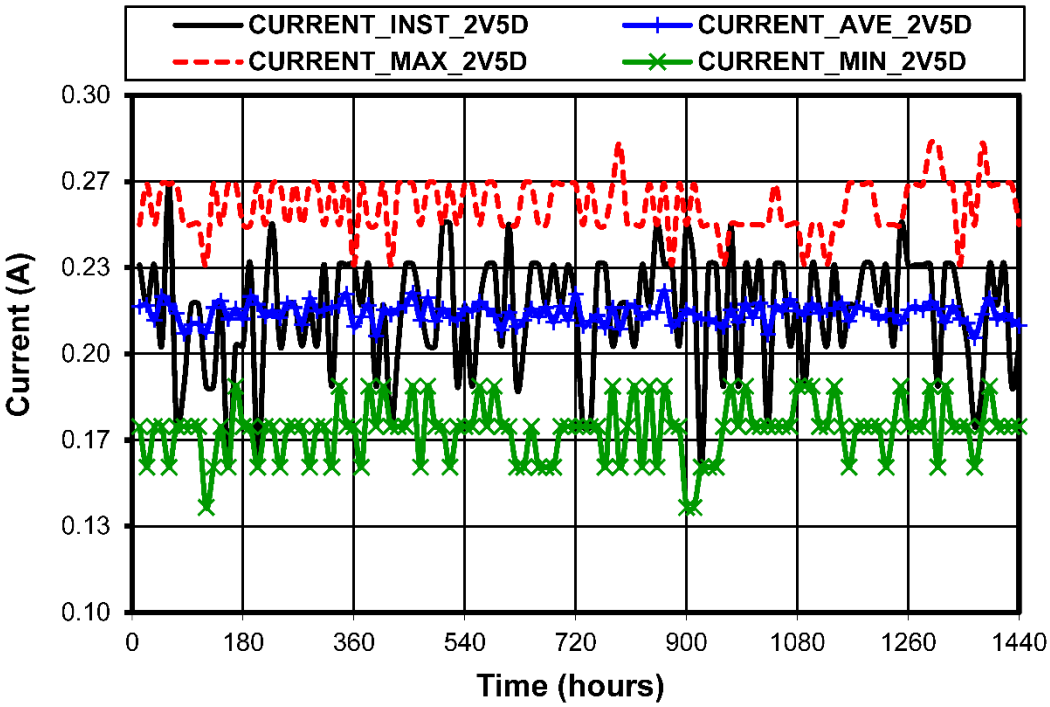

Figure 7-4. Digital 2.5 V rail instantaneous, average, maximum and minimum current (Range: 0.1406 – 0.2812).

Temperature Data

The temperature data for Artemis includes the internal temperature of the two power controllers. This data is passed to the Spartan-6 through PMBUS and printed in a data file. The Artix-7 junction temperature is also measured by a MAX6627 which is connected to the on-chip, diode-connected transistor. Although this data is collected by ControlOS, a bug is preventing the Artix-7 temperature from printing in the data files. Figure 7-5 below shows the temperature of the two power controllers.

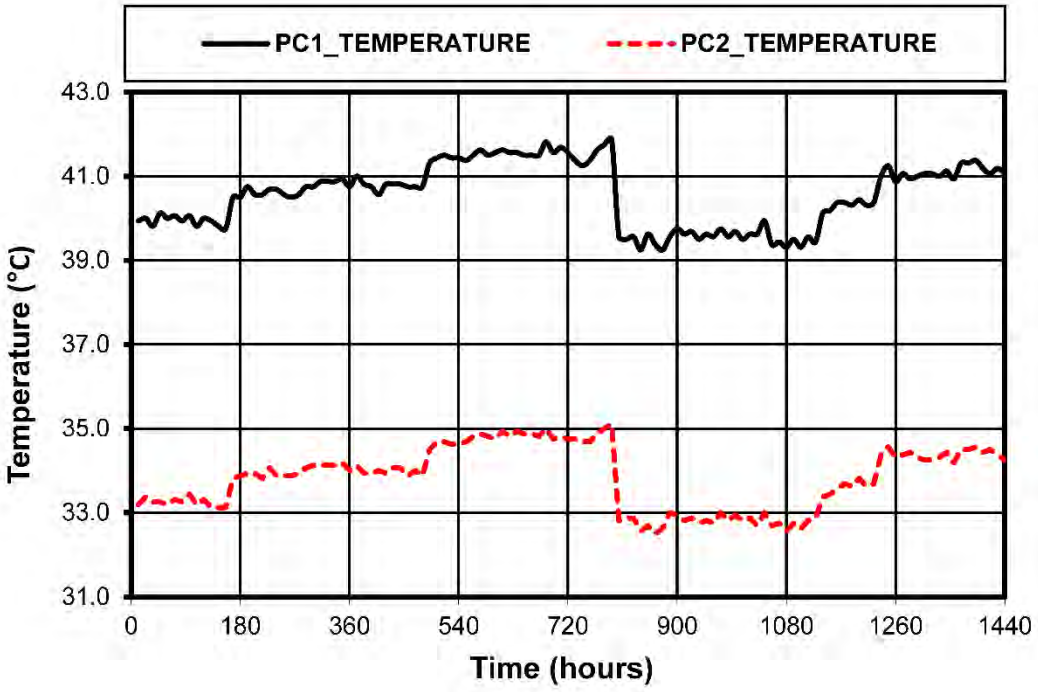

Figure 7-5. Temperature data for power controllers one (digital rails) and two (analog rails).

Operation and Functionality Data

The following data pertains to the radiation tolerant design of the Artemis system. This data is collected by the Spartan-6 and is stored in the TLM_TILE packet of the data. Values such as total number of faults, active processors and counters are displayed.

The Spartan-6 count is derived from the scheduler timer in ControlOS. This count is used to schedule tasks and is viewed as a general system counter for the operation of Artemis. In Figure 7-6, the S6 count is displayed versus system runtime in hours. The data for total faults and total injected faults is displayed in Figure 7-7. Data showing the number of faults on each individual processor tile is shown in Figure 7-8. Figure 7-9 shows which of the nine processors were active at different times during operation. The final data set shows the count of the three active processors in Figure 7-10.

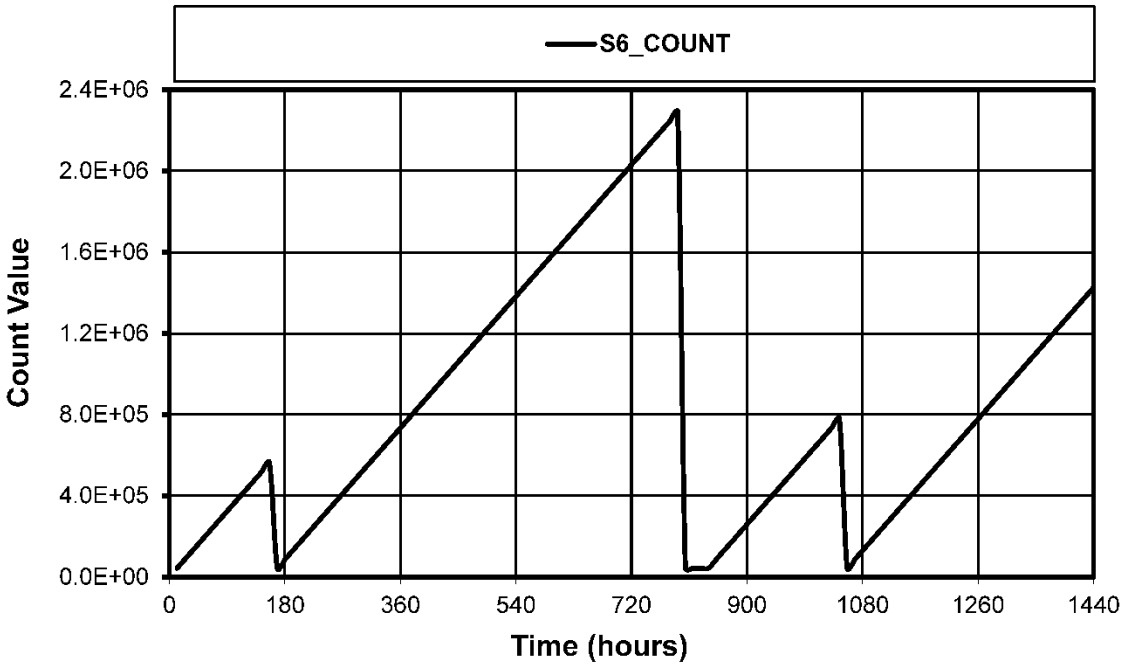

Figure 7-6. Spartan-6 system counter.

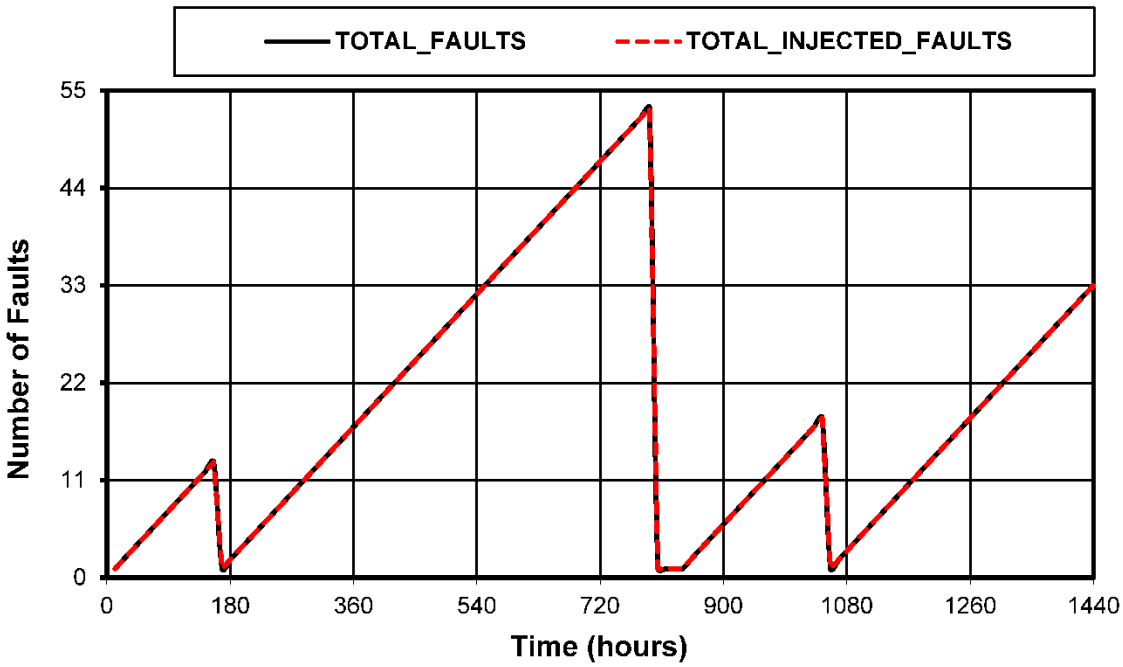

Figure 7-7. Number of total faults and total injected faults

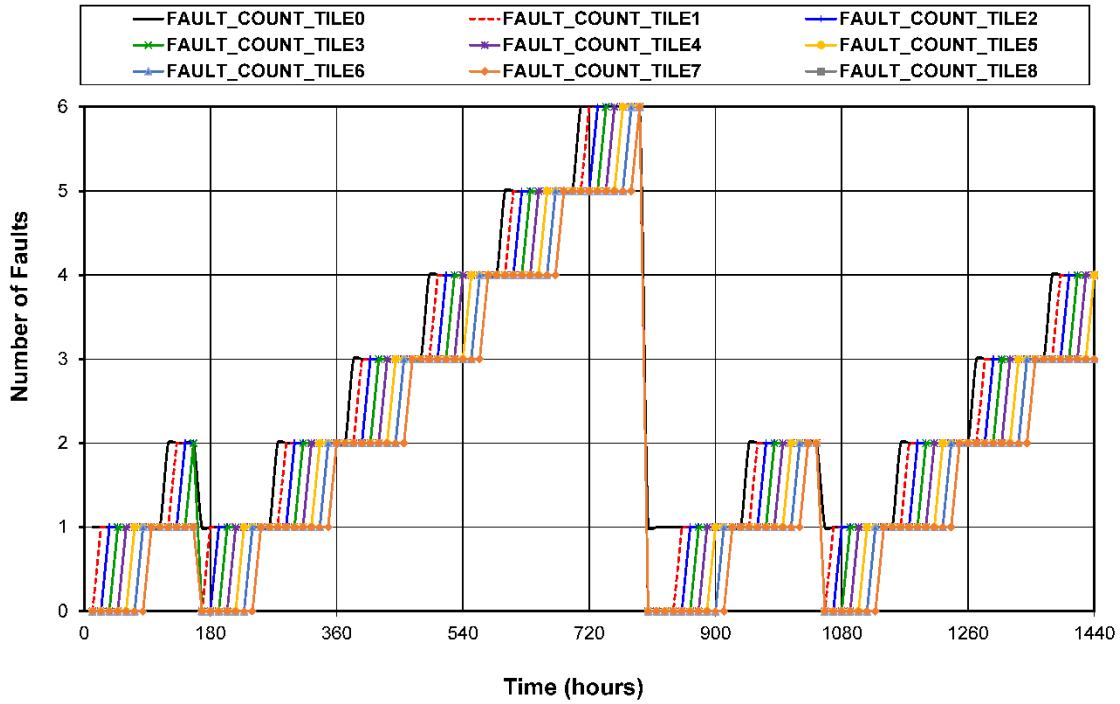

Figure 7-8. Number of faults on each individual tile.

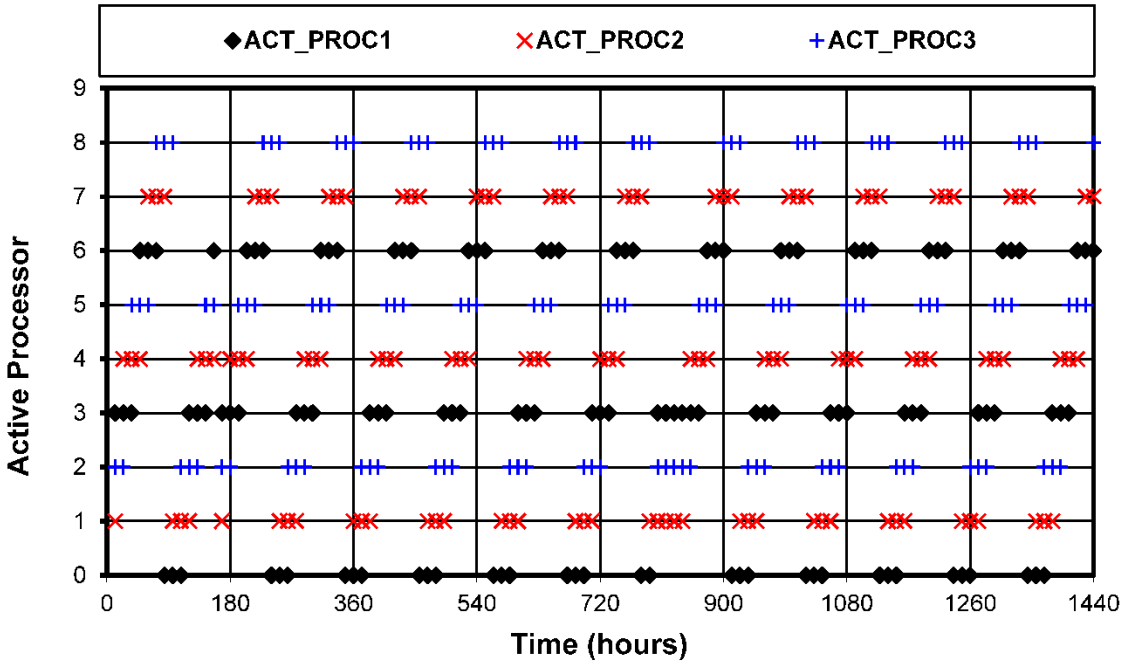

Figure 7-9. Active processor during data file write.

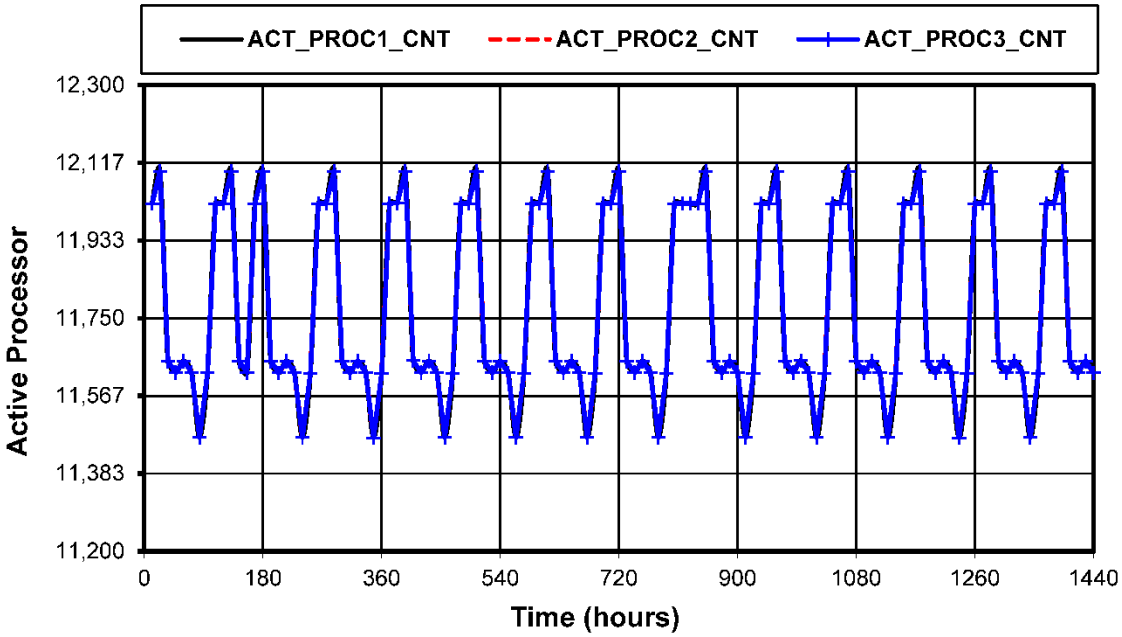

Figure 7-10. Count value of the active processors.

Discussion

This section provides a discussion on the data presented throughout this chapter. Voltage, current and temperature data are discussed first followed by a discussion of the operational data from Artemis.

The voltage data for Artemis is nominal when compared to ground testing. Artemis is receiving a slightly higher voltage than expected at around 5.1 V. All the digital voltage rails are operating within close margins of their desired voltages. The -3 V rail is not operating correctly but this was a known issue and that power rail is not currently in use for the Artemis mission.

The currents data for Artemis is consistent with ground testing. The Artemis system is drawing 400 mA from the power supply which is nominal. All the current data for the digital voltage rails is nominal. The current data for the analog rails is nominal as

well. Since none of the analog rails are in use they have either no current or a small residual current draw.

Power controller one, which controls the digital power rails, is typically about seven degrees Celsius hotter than power controller two which controls the analog rails. The difference may be due to a higher temperature on the board in that area or to a calibration issue between the two power controllers. This temperature difference is consistent with ground testing however the overall temperature of each controller are one to two degrees Celsius hotter on the ISS than they were during pre-flight testing.

The Spartan-6 count (Figure 7-6) shows that the system has been active as it counts every second during operation. Each time the count drops to zero shows when the Artemis unit was power cycled by the NanoRacks platform on the ISS. The linear increase in the Spartan-6 count shows that the FPGA and ControlOS are operating properly.

The data shown in Figure 7-7, showing total faults and total injected faults is different than expected. If Artemis was receiving faults from radiation strikes, the two data sets should deviate from each other showing that there have been more total faults (some natural faults and some injected faults) than just injected faults. This observation implies several findings. Artemis may not have received any natural faults. This may be due to the station shielding the Artemis unit. It is also possible that the Artix-7 is being struck by radiation but no active logic elements are being hit. Figure 7-11 shows that the device utilization of the Artix-7 is low, producing a lower probability that an active region of the FPGA is struck with radiation.

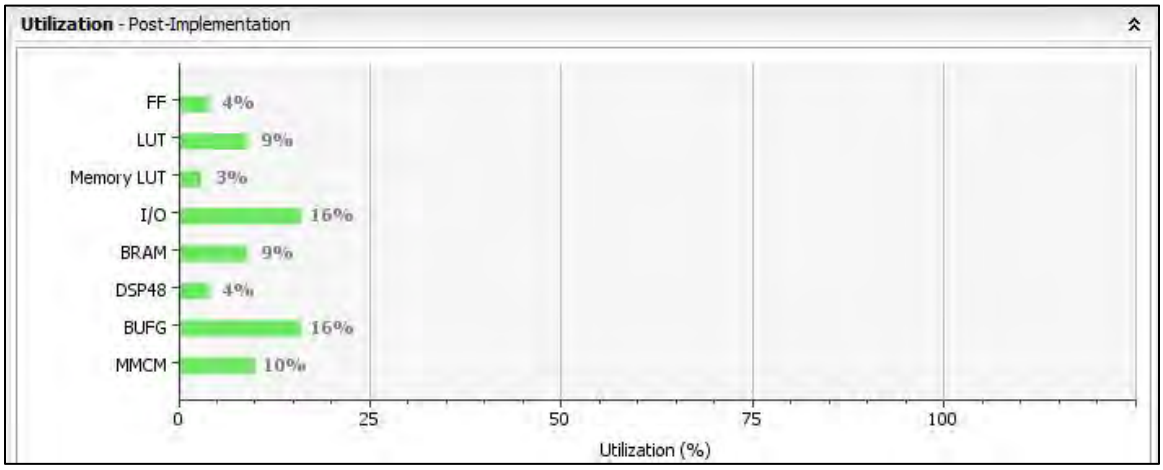

Figure 7-11. Artix-7 utilization report.

The tile fault data in Figure 7-8 is consistent with the previous graph showing total faults. The tile faults all increment by one until the system is power cycled. If a tile had received a natural fault it would show in this figure but as with the total fault data, this graph only shows injected faults.

The active processor data in Figure 7-9 shows that each time a fault occurs the next available spare processor is brought into operation in progressive order. This way each tile processor could be stressed over the duration of the mission. This data would show the active processor moving more often if the Artix-7 was receiving natural faults.

The final data presented for Artemis in Figure 7-10 shows the counter value for the three active processors over time. This count value is output from the active processor, selected by the MUX and voted on by the voter. When one processor has a different count then the other two a fault has occurred. This figure shows that at the time the data files are written, no fault has occurred. If this figure showed a processor count value being different than the other two while no fault is registered in ControlOS then the

system may not be functioning properly. Therefore, in this case the processor count data shows nominal operation.

FUTURE WORK

Artemis Future Development

Tile Initialization

Synchronizing the tiles for Artemis using resets is only a couple steps towards a final design. Ideally an Enable will be used for each tile. When one active tile is faulted the two other active tiles are disabled, effectively pausing them. The new spare tile to be brought online is initialized with the data of the two paused tiles. Once this step has been confirmed, the new tile and the two active tiles are all enabled together, ensuring that they are synchronized.

Readback Scrubbing

Incorporating a working readback scrubber task would add another level of strong fault mitigation. A readback scrubber reads the configuration memory of the FPGA and compares it to a golden copy. If an error is found, the scrubber reconfigures the FPGA. If no error if found, no time is taken to reconfigure the FPGA. This is advantageous over just a blind scrubber which reconfigures the FPGA no matter what, taking away from processing time.

Post Flight Testing

The Artemis mission provides a rare opportunity to perform post flight testing because the flight unit will be returned to MSU after its mission and landing in the return vehicle. An Artemis engineering unit has been running at MSU in close unison with the

Artemis flight unit on the ISS. This will allow for comparison in data between the ground unit and the ISS unit.

Natural Radiation Faults

It may be necessary to re-evaluate and re-simulate the radiation environment using CREME96, an SEE rate prediction tool, to find an explanation on why no natural radiation faults have occurred. This analysis should be performed with the values for the Artix-7 process node, the Artix-7 architecture and updated timing for PR and tile switching.

Another way to potentially produce more noticeable faults would be to utilize more of the FPGA resources. Utilizing more BRAM by pipelining the data into as much memory as possible would create a higher probability of a radiation strike causing a fault in an active region of the FPGA.

RadSat

While data is being received from Artemis on the ISS, the RTCS is being developed and integrated for a satellite mission. Two versions of this mission are being developed, dubbed RadSat-u and RadSat-g. RadSat will be a 3U CubeSat that will most likely 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 RTCS discussed in this thesis. Commands and data will occur between the MFIB and the payload while power is provided by the EPS board.

The power board for Artemis has already undergone a revision for the RadSat mission. This revision involved removing all the analog rail components and the analog power controller, shown in Figure 8-1 below. Without a radiation sensor requiring the five analog voltages produced by the power board, they can be eliminated to reduce the overall power draw of Artemis. With these power rails removed, Artemis draws 310 mA compared to 400 mA with all rails being used. This power saving is crucial for RadSat which has a strict power budget.

Depending on RadSat's orbit, the satellite should be in a harsh radiation environment for 6-12 months. This mission will finally give the RTCS enough time in a radiation environment to advance to TRL-8 and TRL-9. Figure 8-2 shows an exploded CAD model view of RadSat while Figure 8-3 shows a rendering of RadSat in orbit.

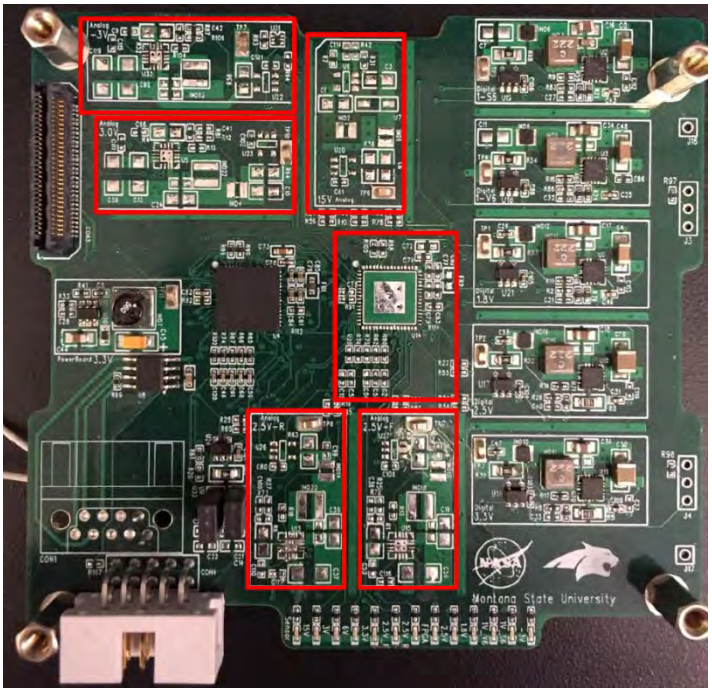

Figure 8-1. Power board with analog power controller and rail components removed (red outline).

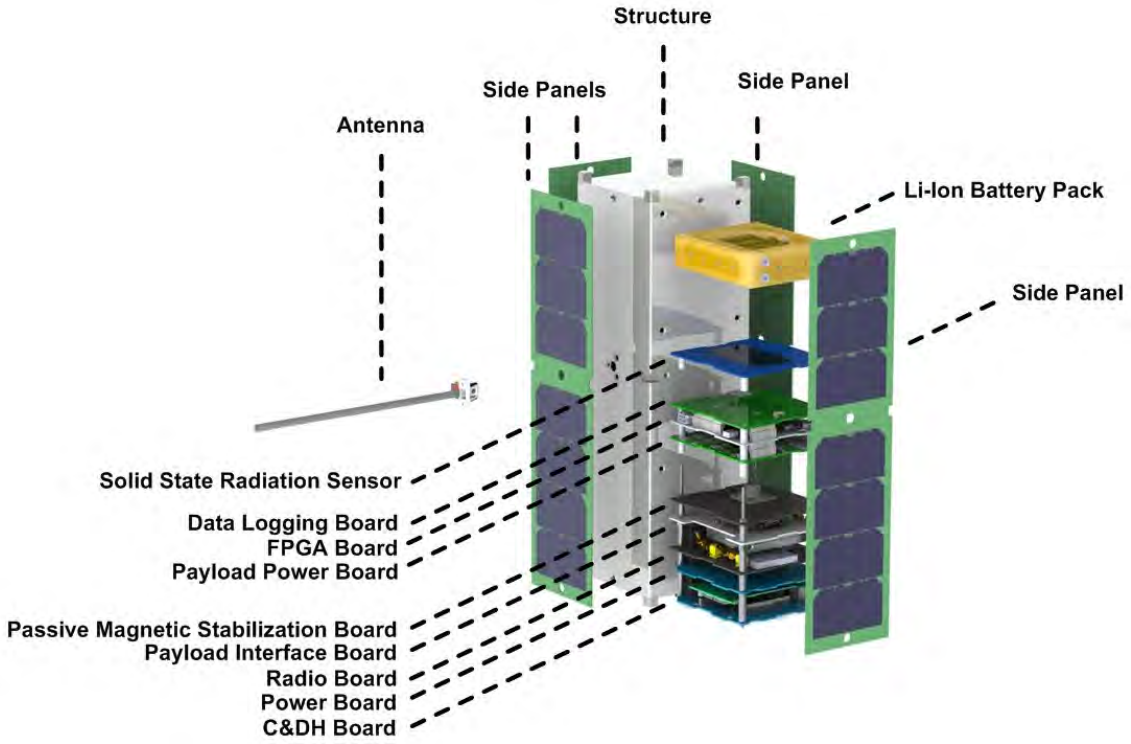

Figure 8-2. Exploded CAD Model of RadSat.

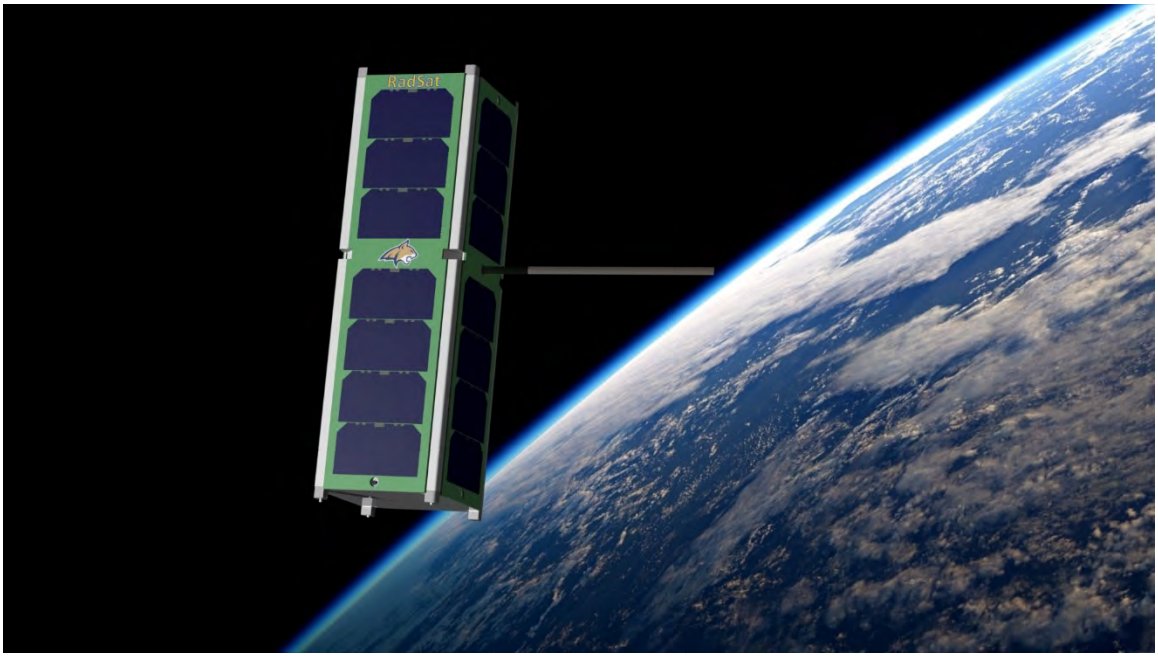

Figure 8-3. Rendering of RadSat in orbit.

CONCLUSION

The research presented in this thesis was 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 more data from Artemis as well as the launch of RadSat into low Earth orbit.

.

REFERENCES CITED

[1] NASA, "NASA Space Technology Roadmaps - Modeling Simulation, Information Technology and Processing Roadmap, Technology Area 11," 2010.

[2] B. J. Lameres, "RadSat - Radiation Tolerant Small Sat Computer System," in *Proceedings of the AIAA/USU Confrence on Small Satellites*, 2015.

[3] S. A. Harkness, "Experiment Platform to Facilitate Flight Testing of Fault Tolerant Reconfigurable Computer Systems," 2015.

[4] H. Barnaby, "Total-Ionizing-Dose Effects in Mordern CMOS Technologies," *IEEE Transactions,* vol. 53, no. Nuclear Science, pp. 3103-3121, 2006.

[5] R. J. Weber, "Reconfigurable Hardware Accelerators for High Performance Radiation Tolerant Computers," 2014.

[6] A. Holmes-Siedle and L. Adams, Handbook of Radiation Effects, 2 ed., New York: Oxford University Press, 2002.

[7] W. Schimmerling, "The Space Radiation Environment: An Introduction," 2011.

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

[9] J. R. Schwank, "Space and Military Radiation Effects in Silicon-on-Insulator Devices," Sandia National Labs, Albuquerque, N.M., 1996.

[10] J. H. Adams, R. Silberberg and C. H. Tsao, "Cosmic Ray Effects on Microelectronics, Part I: The Near-Earth Particle Environment," 1981.

[11] NASA, "NASA Mission Pages," 2013. [Online]. Available: https://www.nasa.gov/mission_pages/sunearth/news/gallery/20130228-radiationbelts.html. [Accessed March 2017].

[12] B. LaMeres, S. Harkness, M. Handley, P. Moholt, C. Julien, T. Kaiser, D. Klumpar, K. Mashburn, L. Springer and G. Crum, "RadSat - Radiation Tolerant SmallSat Computer System," in *SmallSat Conference Proceedings*, Logan, UT, 2015.

[13] B. J. LaMeres, "FPGA-Based Radiation Tolerant Computing," in *Embry-Riddle Aeronautical University Research Colloquium*, Daytona Beach, FL, 2012.

[14] M. Fras, H. Kroha, J. v. Loeben, O. Reimann, R. Richter and B. Weber, "Use of Triple Modular Redundancy (TMR) technology in FPGAs for the reduction of faults due to radiation in the readout of the ATLAS Monitored Drift Tube (MDT) chambers," in *Topical Workshop on Electronics for Particle Physics*, Aachen, Germany, 2010.

[15] NASA, "New Horizons: Spacecraft Systems and Components," 2016.

[16] A. S. Keys, M. D. Watson, D. O. Frazier, J. H. Adams, M. A. Johnson and E. A. Kolawa, "High Performance, Radiation-Hardened Electronics for Space Applications," in *International Planetary Probes Workshop - 5*, Bordeaux, France, 2007.

[17] Xilinx, "Soft Error Mitigation Controller v4.1," Xilinx, 2015.

[18] Xilinx, "Vivado Design Suite User Guide: Partial Reconfiguration," 2016.

[19] J. Fabula, "The Total Ionizing Dose Performance of Deep Submicron CMOS Processes," in *Military and Aerospace Programmable Logic Devices Conference*, Annapolis, 2008.

[20] H. Quinn, "High-Performance Computing for Airborn Applications," in *Nuclear and Space Radiation Effects Conference*, Denver, 2010.

[21] "Technology Readiness Level," NASA, 2012. [Online]. Available: https://www.nasa.gov/directorates/heo/scan/engineering/technology/txt_accordion1.html. [Accessed 2017].

[22] T. M. Buerkle, "Ionizing Radiation Detector For the Environmental Awareness IN FPGA-Based Flight Computers," Bozeman, MT, 2012.

[23] J. S. Hane, "A Fault-Tolerant Computer Architecture for Space Vehicle Applications," Bozeman, MT, 2012.

[24] J. A. Hogan, "Reliability Analysis of Radiation Induced Fault Mitigation Strategies in Field Programmable Gate Arrays," Bozeman, MT, 2014.

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

[26] B. J. LaMeres, "Space Flight Demonstration of a Radiation Tolerant, FPGA-based Computer System on the International Space Station," 2014.

[27] Z. Jacobs and M. Bezold, "Interface Control Document Between CubeLab Modules and the NanoRacks Platform," 2011.

[28] T.I., "12-Rail Sequencer and System Health Monitor with Fan Control," Texas Instruments, 2010.

[29] T.I., "42V, 0.3A, Step-Down Voltage Regulator in SOT-23," Texas Instruments, 2013.

[30] T.I., "20V, 1.4A Step-Up Voltage Regulator in SOT-23," Texas Instruments, 2013.

[31] T.I., "Using the TPS6215X In An Inverting Buck-Boost Topology," Texas Instruments, 2013.

[32] T.I., "PMBus Specification Ver 1.1," 2007.

[33] Xilinx, "Artix-7 FPGAs Datasheet: DC and AC Switching Characteristics," Xilinx Inc., 2015.

[34] "7 Series Configuration Guide," Xilinx, 2016.

[35] "MAX6627/MAX6628 Datasheet," Maxim Integrated, 2016.

[36] Xilinx, "Clocking Wizard v5.1," Xilinx Inc, 2015.

[37] C. Julien and K. Mashburn, "Shop Order: Artemis Program," Space Science and Engineering Laboratory, Bozeman, MT, 2016.

APPENDICES

APPENDIX A

Artemis Data File Example

```
000    SYNC FA
001    HEADER      TLM_TILE
002    S6_COUNT    0000043203
006    ACT_TILES   0213
008    FAULTED_TILES    000
00A    FAULT_COUNT_TILE0     00000001
00C    FAULT_COUNT_TILE1     00000000
00E    FAULT_COUNT_TILE2     00000000
010    FAULT_COUNT_TILE3     00000000
012    FAULT_COUNT_TILE4     00000000
014    FAULT_COUNT_TILE5     00000000
016    FAULT_COUNT_TILE6     00000000
018    FAULT_COUNT_TILE7     00000000
01A    FAULT_COUNT_TILE8     00000000
01C    TOTAL_INJECTED_FAULTS       00000001
01E    TOTAL_FAULTS     00000001
020    NEXT_SPARE       4
022    READBACK_FAULTS       0
024    GSWITCH     0
025    WATCHDOG 00000
027    ACT_PROC1  03
028    ACT_PROC2  01
029    ACT_PROC3  02
02A    ACT_PROC1_CNT    12021
02C    ACT_PROC2_CNT    12021
02E    ACT_PROC3_CNT    12021
030    VOTER       00003
031    CRC    7D
032    CRC    64
000    SYNC FA
001    HEADER      TLM_HEALTH
002    VOLTAGE_INST_BATT     5.1338
006    VOLTAGE_AVE_BATT      5.1225
00A    VOLTAGE_MAX_BATT      5.1338
00E    VOLTAGE_MIN_BATT      0.0000
012    VOLTAGE_INST_15V0A    14.9941
016    VOLTAGE_AVE_15V0A     14.9979
01A    VOLTAGE_MAX_15V0A     15.0312
01E    VOLTAGE_MIN_15V0A     14.9727
022    VOLTAGE_INST_-3V0A    0.3003
026    VOLTAGE_AVE_-3V0A     0.2987
02A    VOLTAGE_MAX_-3V0A     0.3003
02E    VOLTAGE_MIN_-3V0A     0.2969
032    VOLTAGE_INST_3V3D     3.3010
```

036	VOLTAGE_AVE_3V3D	3.2969
03A	VOLTAGE_MAX_3V3D	3.3030
03E	VOLTAGE_MIN_3V3D	3.2915
042	VOLTAGE_INST_3V0A	3.0061
046	VOLTAGE_AVE_3V0A	3.0019
04A	VOLTAGE_MAX_3V0A	3.0137
04E	VOLTAGE_MIN_3V0A	2.9890
052	VOLTAGE_INST_2V5D	2.5048
056	VOLTAGE_AVE_2V5D	2.5054
05A	VOLTAGE_MAX_2V5D	2.5238
05E	VOLTAGE_MIN_2V5D	2.4828
062	VOLTAGE_INST_2V5FA	2.5139
066	VOLTAGE_AVE_2V5FA	2.5078
06A	VOLTAGE_MAX_2V5FA	2.5231
06E	VOLTAGE_MIN_2V5FA	2.4901
072	VOLTAGE_INST_2V5RA	2.5018
076	VOLTAGE_AVE_2V5RA	2.4995
07A	VOLTAGE_MAX_2V5RA	2.5066
07E	VOLTAGE_MIN_2V5RA	2.4872
082	VOLTAGE_INST_1V8D	1.8025
086	VOLTAGE_AVE_1V8D	1.8001
08A	VOLTAGE_MAX_1V8D	1.8037
08E	VOLTAGE_MIN_1V8D	1.7896
092	VOLTAGE_INST_1V0SD	1.0071
096	VOLTAGE_AVE_1V0SD	1.0033
09A	VOLTAGE_MAX_1V0SD	1.0083
09E	VOLTAGE_MIN_1V0SD	0.9919
0A2	VOLTAGE_INST_1V0VD	0.9521
0A6	VOLTAGE_AVE_1V0VD	0.9499
0AA	VOLTAGE_MAX_1V0VD	0.9521
0AE	VOLTAGE_MIN_1V0VD	0.9473
0B2	CURRENT_INST_BATT	0.3906
0B6	CURRENT_AVE_BATT	0.4028
0BA	CURRENT_MAX_BATT	0.4219
0BE	CURRENT_MIN_BATT	0.0000
0C2	CURRENT_INST_15V0A	0.0000
0C6	CURRENT_AVE_15V0A	0.0000
0CA	CURRENT_MAX_15V0A	0.0000
0CE	CURRENT_MIN_15V0A	0.0000
0D2	CURRENT_INST_-3V0A	0.0000
0D6	CURRENT_AVE_-3V0A	0.0000
0DA	CURRENT_MAX_-3V0A	0.0000
0DE	CURRENT_MIN_-3V0A	0.0000
0E2	CURRENT_INST_3V3D	0.0781

0E6	CURRENT_AVE_3V3D	0.0781
0EA	CURRENT_MAX_3V3D	0.0781
0EE	CURRENT_MIN_3V3D	0.0781
0F2	CURRENT_INST_3V0A	0.0312
0F6	CURRENT_AVE_3V0A	0.0312
0FA	CURRENT_MAX_3V0A	0.0312
0FE	CURRENT_MIN_3V0A	0.0312
102	CURRENT_INST_2V5D	0.2344
106	CURRENT_AVE_2V5D	0.2183
10A	CURRENT_MAX_2V5D	0.2500
10E	CURRENT_MIN_2V5D	0.1719
112	CURRENT_INST_2V5FA	0.0312
116	CURRENT_AVE_2V5FA	0.0312
11A	CURRENT_MAX_2V5FA	0.0312
11E	CURRENT_MIN_2V5FA	0.0312
122	CURRENT_INST_2V5RA	0.0156
126	CURRENT_AVE_2V5RA	0.0156
12A	CURRENT_MAX_2V5RA	0.0156
12E	CURRENT_MIN_2V5RA	0.0156
132	CURRENT_INST_1V8D	0.0938
136	CURRENT_AVE_1V8D	0.0951
13A	CURRENT_MAX_1V8D	0.1094
13E	CURRENT_MIN_1V8D	0.0938
142	CURRENT_INST_1V0SD	0.1562
146	CURRENT_AVE_1V0SD	0.1576
14A	CURRENT_MAX_1V0SD	0.1719
14E	CURRENT_MIN_1V0SD	0.1562
152	CURRENT_INST_1V0VD	0.0781
156	CURRENT_AVE_1V0VD	0.0781
15A	CURRENT_MAX_1V0VD	0.0781
15E	CURRENT_MIN_1V0VD	0.0781
162	V6_TEMPERATURE	00.0000
164	PC1_TEMPERATURE	39.9375
166	PC1_EXT_TEMPERATURE	00.0000
168	PC2_TEMPERATURE	33.1875
16A	PC2_EXT_TEMPERATURE	00.0000
16C	SYSTEM_RUNTIME	0000d, 11h, 58m, 17s
174	SYSTEM_STATUS_FLAG	00000000
175	CRC	72
176	CRC	EA

APPENDIX B

Artemis Packet Structure

Table B-1. TLM_TILE packet of Artemis data.

Mnemonic	Size	Offset	Type	Units	Description
SYNC	1	0	INT8U	none	0xC0 Sync byte
PKT_TYPE	1	1	INT8U	none	0x88 TLM_TILE packet type identifier
S6_COUNT	4	2	INT32U	Counts	Count value
ACT_TILES	2	6	INT16U	none	Active processors
FAULTED_TILES	2	8	INT16U	none	Faulted processors
FAULT_COUNT_TILE0	2	10	INT8U	Counts	Tile0 fault counter
FAULT_COUNT_TILE1	2	12	INT8U	Counts	Tile1 fault counter
FAULT_COUNT_TILE2	2	14	INT8U	Counts	Tile2 fault counter
FAULT_COUNT_TILE3	2	16	INT8U	Counts	Tile3 fault counter
FAULT_COUNT_TILE4	2	18	INT8U	Counts	Tile4 fault counter
FAULT_COUNT_TILE5	2	20	INT8U	Counts	Tile5 fault counter
FAULT_COUNT_TILE6	2	22	INT8U	Counts	Tile6 fault counter
FAULT_COUNT_TILE7	2	24	INT8U	Counts	Tile7 fault counter
FAULT_COUNT_TILE8	2	26	INT8U	Counts	Tile8 fault counter
READBACK_FAULTS	2	28	INT16U	none	Number of readback faults that have occurred
WATCHDOG	2	30	INT8U	none	How many watchdog repairs have occurred
ACT_PROC1	1	32	INT8U	none	First active tile with processor
ACT_PROC2	1	34	INT8U	none	Second active tile with processor
ACT_PROC3	1	36	INT8U	none	Third active with processor
ACT_PROC1_CNT	2	38	INT16U	Counts	First active processor counter
ACT_PROC2_CNT	2	40	INT16U	Counts	Second active processor counter
ACT_PROC3_CNT	2	42	INT16U	Counts	Third active processor counter
VOTER_CNTS	2	44	INT16U	none	Voter value
CRC	2	46	INT16U	None	CRC
SYNC	1	47	INT8U	none	0xC0 Sync byte

Table B-2. TLM_HEALTH packet of Artemis data.

Mnemonic	Size	Offset	Type	Units	Description
SYNC	1	0	INT8U	none	0xC0 Sync byte
PKT_TYPE	1	1	INT8U	none	**0x33** TLM_HEALTH packet type identifier
VOLTAGE_INS_BATT	2	2	INT16S	mV	Analog Tlm
VOLTAGE_AVE_BATT	2	4	INT16S	mV	Analog Tlm
VOLTAGE_MAX_BATT	2	6	INT16S	mV	Analog Tlm
VOLTAGE_MIN_BATT	2	8	INT16S	mV	Analog Tlm
VOLTAGE_INS_15V0A	2	10	INT16S	mV	Analog Tlm
VOLTAGE_AVE_15V0A	2	12	INT16S	mV	Analog Tlm
VOLTAGE_MAX_15V0A	2	14	INT16S	mV	Analog Tlm
VOLTAGE_MIN_15V0A	2	16	INT16S	mV	Analog Tlm
VOLTAGE_INS_N3V0A	2	18	INT16S	mV	Analog Tlm
VOLTAGE_AVE_N3V0A	2	20	INT16S	mV	Analog Tlm
VOLTAGE_MAX_N3V0A	2	22	INT16S	mV	Analog Tlm
VOLTAGE_MIN_N3V0A	2	24	INT16S	mV	Analog Tlm
VOLTAGE_INS_3V3D	2	26	INT16S	mV	Analog Tlm
VOLTAGE_AVE_3V3D	2	28	INT16S	mV	Analog Tlm
VOLTAGE_MAX_3V3D	2	30	INT16S	mV	Analog Tlm
VOLTAGE_MIN_3V3D	2	32	INT16S	mV	Analog Tlm
VOLTAGE_INS_3V0A	2	34	INT16S	mV	Analog Tlm
VOLTAGE_AVE_3V0A	2	36	INT16S	mV	Analog Tlm
VOLTAGE_MAX_3V0A	2	38	INT16S	mV	Analog Tlm
VOLTAGE_MIN_3V0A	2	40	INT16S	mV	Analog Tlm
VOLTAGE_INS_2V5D	2	42	INT16S	mV	Analog Tlm
VOLTAGE_AVE_2V5D	2	44	INT16S	mV	Analog Tlm
VOLTAGE_MAX_2V5D	2	46	INT16S	mV	Analog Tlm
VOLTAGE_MIN_2V5D	2	48	INT16S	mV	Analog Tlm
VOLTAGE_INS_2V5FA	2	50	INT16S	mV	Analog Tlm
VOLTAGE_AVE_2V5FA	2	52	INT16S	mV	Analog Tlm
VOLTAGE_MAX_2V5FA	2	54	INT16S	mV	Analog Tlm
VOLTAGE_MIN_2V5FA	2	56	INT16S	mV	Analog Tlm
VOLTAGE_INS_2V5RA	2	58	INT16S	mV	Analog Tlm
VOLTAGE_AVE_2V5RA	2	60	INT16S	mV	Analog Tlm
VOLTAGE_MAX_2V5RA	2	62	INT16S	mV	Analog Tlm
VOLTAGE_MIN_2V5RA	2	64	INT16S	mV	Analog Tlm
VOLTAGE_INS_1V8D	2	66	INT16S	mV	Analog Tlm
VOLTAGE_AVE_1V8D	2	68	INT16S	mV	Analog Tlm
VOLTAGE_MAX_1V8D	2	70	INT16S	mV	Analog Tlm
VOLTAGE_MIN_1V8D	2	72	INT16S	mV	Analog Tlm

Mnemonic	Size	Offset	Type	Units	Description
VOLTAGE_INS_1V0SD	2	74	INT16S	mV	Analog Tlm
VOLTAGE_AVE_1V0SD	2	76	INT16S	mV	Analog Tlm
VOLTAGE_MAX_1V0SD	2	78	INT16S	mV	Analog Tlm
VOLTAGE_MIN_1V0SD	2	80	INT16S	mV	Analog Tlm
VOLTAGE_INS_1V0VD	2	82	INT16S	mV	Analog Tlm
VOLTAGE_AVE_1V0VD	2	84	INT16S	mV	Analog Tlm
VOLTAGE_MAX_1V0VD	2	86	INT16S	mV	Analog Tlm
VOLTAGE_MIN_1V0VD	2	88	INT16S	mV	Analog Tlm
CURRENT_INS_BATT	2	90	INT16S	mA	Analog Tlm
CURRENT_AVE_BATT	2	92	INT16S	mA	Analog Tlm
CURRENT_MAX_BATT	2	94	INT16S	mA	Analog Tlm
CURRENT_MIN_BATT	2	96	INT16S	mA	Analog Tlm
CURRENT_INS_15V0A	2	98	INT16S	mA	Analog Tlm
CURRENT_AVE_15V0A	2	100	INT16S	mA	Analog Tlm
CURRENT_MAX_15V0A	2	102	INT16S	mA	Analog Tlm
CURRENT_MIN_15V0A	2	104	INT16S	mA	Analog Tlm
CURRENT_INS_N3V0A	2	106	INT16S	mA	Analog Tlm
CURRENT_AVE_N3V0A	2	108	INT16S	mA	Analog Tlm
CURRENT_MAX_N3V0A	2	110	INT16S	mA	Analog Tlm
CURRENT_MIN_N3V0A	2	112	INT16S	mA	Analog Tlm
CURRENT_INS_3V3D	2	114	INT16S	mA	Analog Tlm
CURRENT_AVE_3V3D	2	116	INT16S	mA	Analog Tlm
CURRENT_MAX_3V3D	2	118	INT16S	mA	Analog Tlm
CURRENT_MIN_3V3D	2	120	INT16S	mA	Analog Tlm
CURRENT_INS_3V0A	2	122	INT16S	mA	Analog Tlm
CURRENT_AVE_3V0A	2	124	INT16S	mA	Analog Tlm
CURRENT_MAX_3V0A	2	126	INT16S	mA	Analog Tlm
CURRENT_MIN_3V0A	2	128	INT16S	mA	Analog Tlm
CURRENT_INS_2V5D	2	130	INT16S	mA	Analog Tlm
CURRENT_AVE_2V5D	2	132	INT16S	mA	Analog Tlm
CURRENT_MAX_2V5D	2	134	INT16S	mA	Analog Tlm
CURRENT_MIN_2V5D	2	136	INT16S	mA	Analog Tlm
CURRENT_INS_2V5FA	2	138	INT16S	mA	Analog Tlm
CURRENT_AVE_2V5FA	2	140	INT16S	mA	Analog Tlm
CURRENT_MAX_2V5FA	2	142	INT16S	mA	Analog Tlm
CURRENT_MIN_2V5FA	2	144	INT16S	mA	Analog Tlm
CURRENT_INS_2V5RA	2	146	INT16S	mA	Analog Tlm
CURRENT_AVE_2V5RA	2	148	INT16S	mA	Analog Tlm
CURRENT_MAX_2V5RA	2	150	INT16S	mA	Analog Tlm
CURRENT_MIN_2V5RA	2	152	INT16S	mA	Analog Tlm
CURRENT_INS_1V8D	2	154	INT16S	mA	Analog Tlm
CURRENT_AVE_1V8D	2	156	INT16S	mA	Analog Tlm

Mnemonic	Size	Offset	Type	Units	Description
CURRENT_MAX_1V8D	2	158	INT16S	mA	Analog Tlm
CURRENT_MIN_1V8D	2	160	INT16S	mA	Analog Tlm
CURRENT_INS_1V0SD	2	162	INT16S	mA	Analog Tlm
CURRENT_AVE_1V0SD	2	164	INT16S	mA	Analog Tlm
CURRENT_MAX_1V0SD	2	166	INT16S	mA	Analog Tlm
CURRENT_MIN_1V0SD	2	168	INT16S	mA	Analog Tlm
CURRENT_INS_1V0VD	2	170	INT16S	mA	Analog Tlm
CURRENT_AVE_1V0VD	2	172	INT16S	mA	Analog Tlm
CURRENT_MAX_1V0VD	2	174	INT16S	mA	Analog Tlm
CURRENT_MIN_1V0VD	2	176	INT16S	mA	Analog Tlm
V6_TEMPERATURE	1	177	INT8S	°C	Virtex-6 junction temperature
PC1_TEMPERATURE	1	178	INT8S	°C	Power controller #1 internal temp sensor
PC1_EXT_TEMPERATURE	1	179	INT8S	°C	Power controller #1 external temp sensor
PC2_TEMPERATURE	1	180	INT8S	°C	Power controller #2 internal temp sensor measurement
PC2_EXT_TEMPERATURE	1	181	INT8S	°C	Power controller #2 external temp sensor
SYSTEM_RUNTIME_DAYS	2	183	INT16U	days	System runtime in days
SYSTEM_RUNTIME_MS	2	185	INT16U	ms	System runtime in milliseconds
SYSTEM_STATUS_FLAG	2	187	INT16U	none	TBD
CRC	2	189	INT16U	None	CRC
SYNC	1	190	INT8U	none	0xC0 Sync byte

APPENDIX C

Artemis Cleaning, Staking and Assembly

The following table shows the steps taken to clean, stake and assemble the Artemis flight unit. Refer to the Artemis Shop Order documentation for images and more detail [37].

Table C-1. Steps and operations to clean, stake and assemble the Artemis flight unit.

Step Number	Operation
1	Gather all parts and materials as listed in the table on page 2 of [37]
2	Clean top and bottom interface, power and FPGA boards using the following cleaning procedure: Bath: - Prepare bath of 3% concentration of Riptide degreaser - Vigorously scrub both sides of the PCB at least twice using a horse-hair brush - Rinse very well with Riptide degreaser Drying: - Dry in convection oven at 60degC for 3-5 hours Bath: - Prepare bath of 95% ETOH - Vigorously scrub both sides of the PCB at least twice using a horse-hair brush - Rinse very well with 95% ETOH Drying: - Dry in convection oven at 60degC for 3-5 hours Bath: - Prepare bath of 95% IPA - Vigorously scrub both sides of the PCB at least twice using a horse-hair brush - Rinse very well with 95% IPA Drying: - Dry in convection oven at 60degC for 3-5 hours
3	Perform final visual inspection of all PCBs under microscope with at least 5x power. Repeat Bath if needed to remove visual contamination.
4	Store PCB in clean ESD-safe container until needed for further processing
5	Prepare Arathane 5753 using a 20:100:7 by weight mix ratio as shown below: Huntsman, Inc.

Step Number	Operation
	Arathane 5753 A Mass: 2.44 grams Actual Mass: Arathane 5753 B Mass: 12.2 grams Actual Mass: Cab-O-Sil M-6 Mass: 0.854 grams Actual Mass: Mix Date/Time: ___9/12/2016 at 9:00am Degas epoxy for 5-10mins or until the bubbles collapse
6	Stake the following items on the PCBs within 1hour of mixing: Large ICs, inductors, capacitors, oscillators, FPGA select switch, programming jumper, harness connectors/jumpers and other piece parts, as required
7	Cure staking compound for at least 8-10 hours at 60degC Full room temp cure will complete in 7 days.
8	Perform final visual inspection of all PCBs under microscope with at least 5x power. Verify that no staking material has seeped between or under pins of the integrated circuits or connector housing
9	Orient bottom plate of flight chassis plate as shown and install Bottom Interface board. Install four threaded rods through the interface board and into the baseplate.
10	Install four M3 x 0.472” aluminum spacers onto the threaded rods above the interface board.
11	Install Power board above the interface board on the 0.472” aluminum spacers. Install four M3 x 0.472” aluminum spacers onto the threaded rods above the Power board.
12	Install FPGA board above the Power board on the 0.472” aluminum spacers. Carefully engage the inter-board connectors and ensue a solid connection. Install four M3 x 0.472” aluminum spacers onto the threaded rods above the FPGA board.
13	Install the Top Interface board above the Detector boards on the 0.472” aluminum spacers. Carefully engage the inter-board connectors and ensure a solid connection.
14	Install four 0.948” aluminum spacers onto the threaded above the top interface board.
15	Install four 0.472” aluminum spacers onto the threaded rods above the 0.948” spacers as shown. Install four mid-plane standoffs (with their included spacers) onto the threaded rods as shown. Note the location of each standoff per its A, B, C, D markings (second image). Verify that the threaded rods are just flush under the bottom plate. Secure using four M3 SS small pattern nuts. Perform a fit-check with the chassis walls to ensure the mounting holes of the mid-plane standoffs line up with the appropriate holes in

Step Number	Operation
	the chassis frame. If needed, add a shim(s) under each of the mid-plane standoffs to match the height of the chassis mounting holes.
16	Perform Artemis functional test for the ISS mission.
17	Install chassis frame and top plate onto the baseplate and secure with fourteen M3 x 5mm FHS.
18	Prepare 3M Scotch Weld 2216 using a 7:5 weight mix ratio as shown below: Huntsman, Inc. 3M Scotch Weld 2216 Part A Mass: 8.05 grams Actual Mass: 3M Scotch Weld 2216 Part B Mass: 5.75 grams Actual Mass: Degas epoxy for 5-10mins or until the bubbles collapse
19	Stake the following items within 45 minutes of mixing: The nut, spacer and standoff of each threaded rod. Each M3 FHS on outside of chassis.
20	Cure staking compound for at least 3 hours at 50degC
21	Perform mission simulation testing to ensure end to end performance of the Artemis payload.
22	Package in clean ESD-safe wrapping for transport to NanoRacks LLC

APPENDIX D

Additional Artemis Data

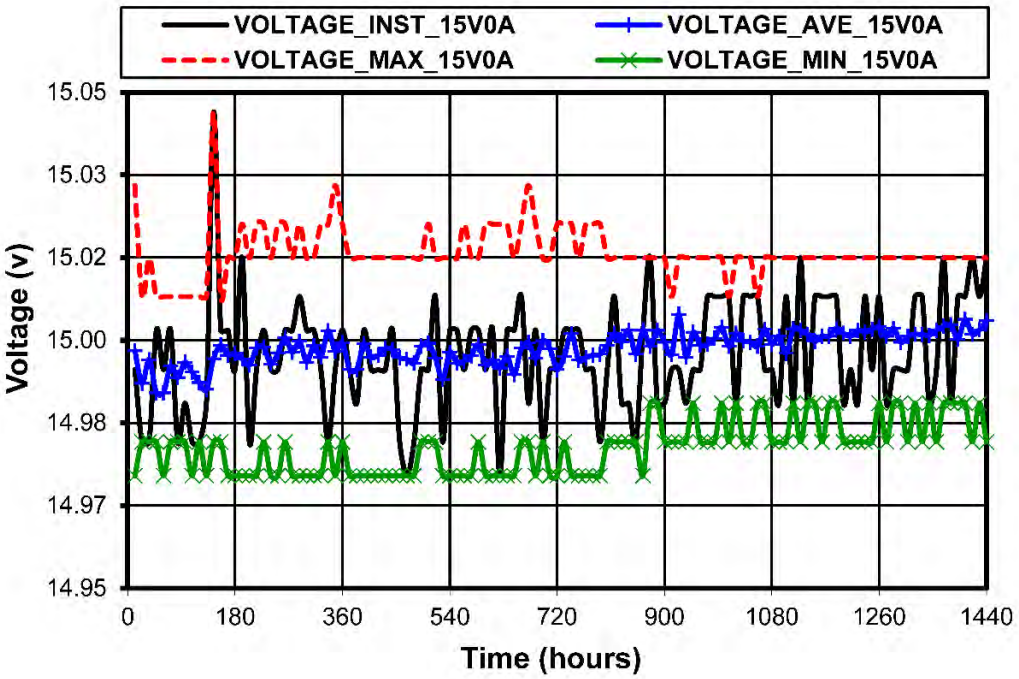

Figure 11-1. Analog 15 V rail instantaneous, average, maximum and minimum voltage (Nominal = 15.0 V, Max = 15.75 V, Min = 14.25 V).

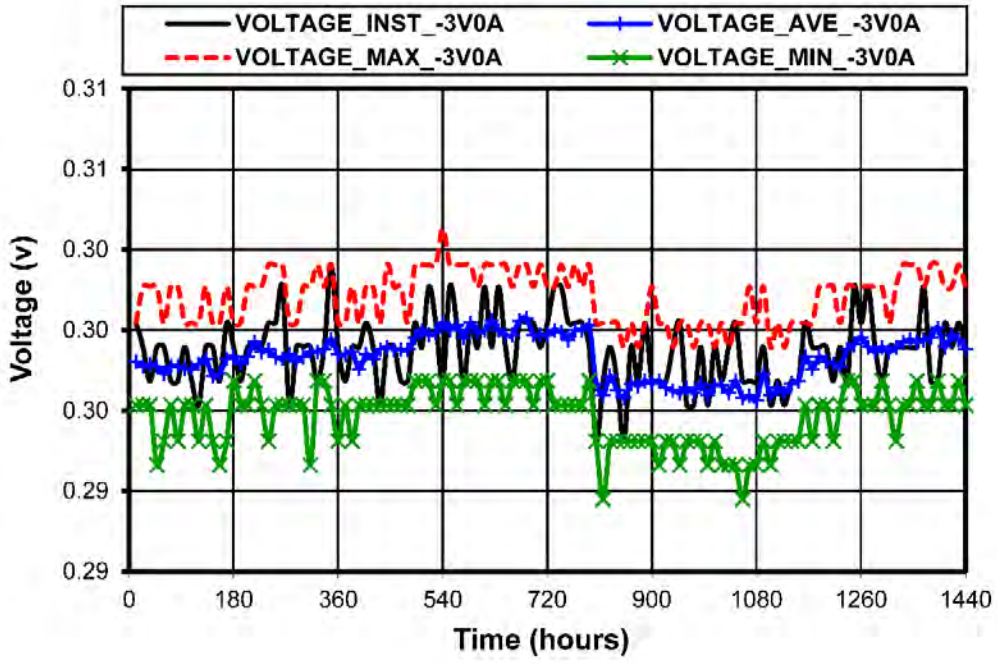

Figure 11-2. Analog -3.0 V rail instantaneous, average, maximum and minimum voltage (Nominal = -3.0 V, Max = -3.15 V, Min = -2.85 V).

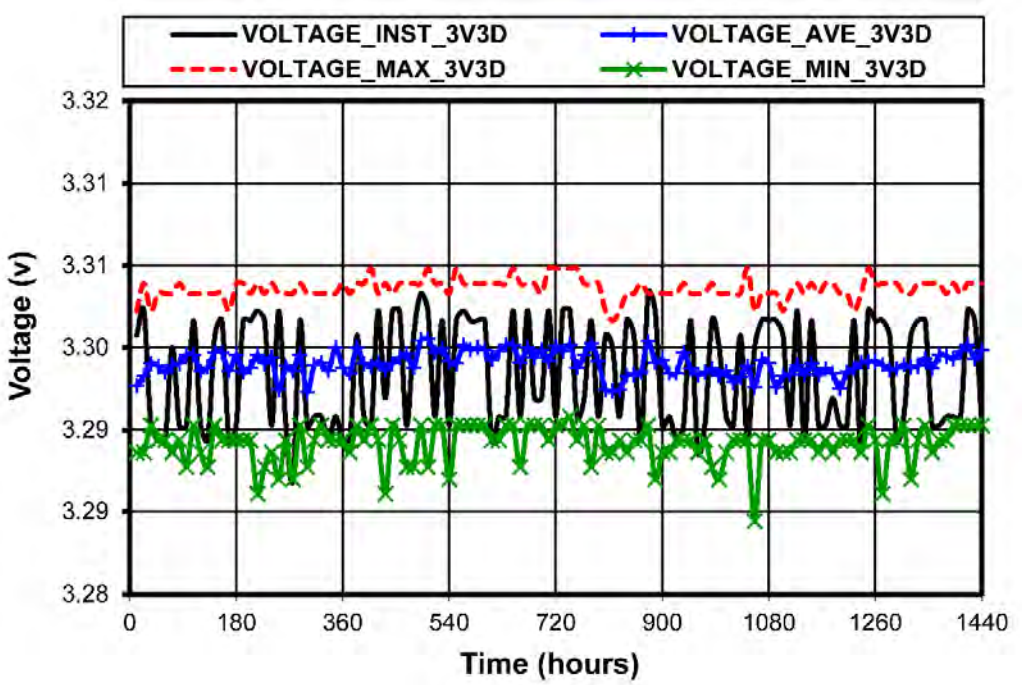

Figure 11-3. Digital 3.3 V rail instantaneous, average, maximum and minimum voltage (Nominal = 3.3 V, Max = 3.465 V, Min = 3.135 V).

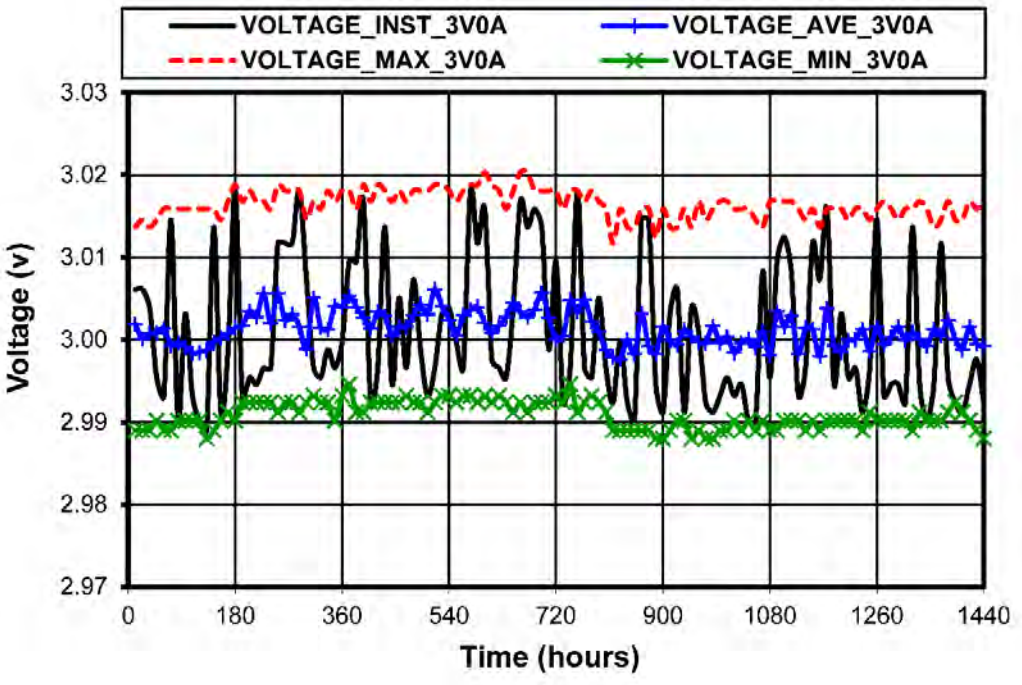

Figure 11-4. Analog 3.0 V rail instantaneous, average, maximum and minimum voltage (Nominal = 3.0 V, Max = 3.15 V, Min = 2.85 V).

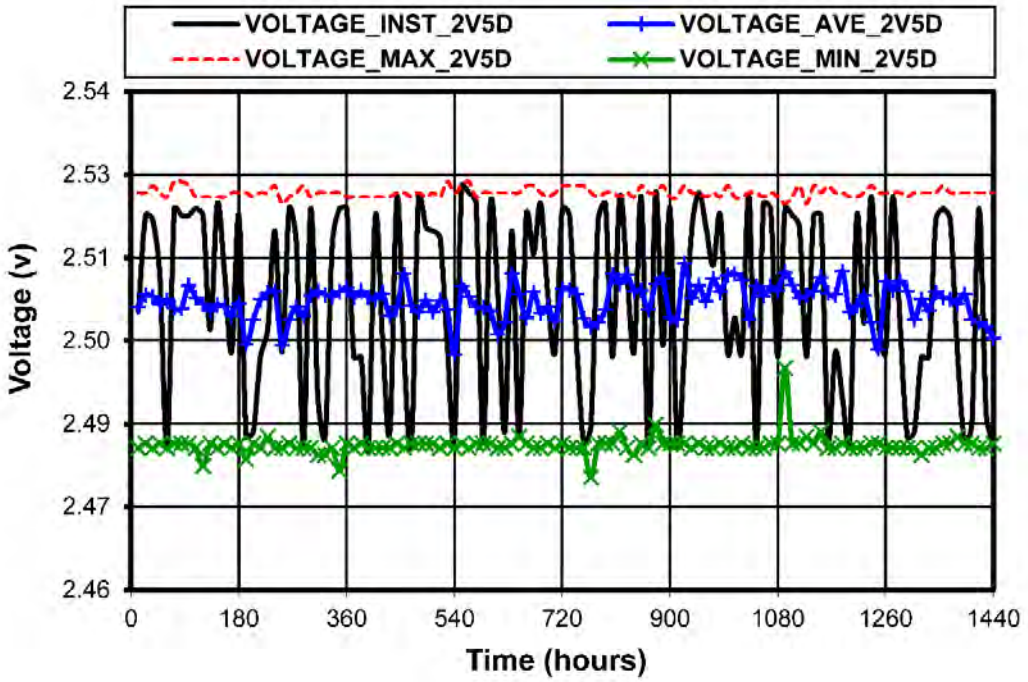

Figure 11-5. Digital 2.5 V rail instantaneous, average, maximum and minimum voltage (Nominal = 2.5 V, Max = 2.625 V, Min = 2.375 V).

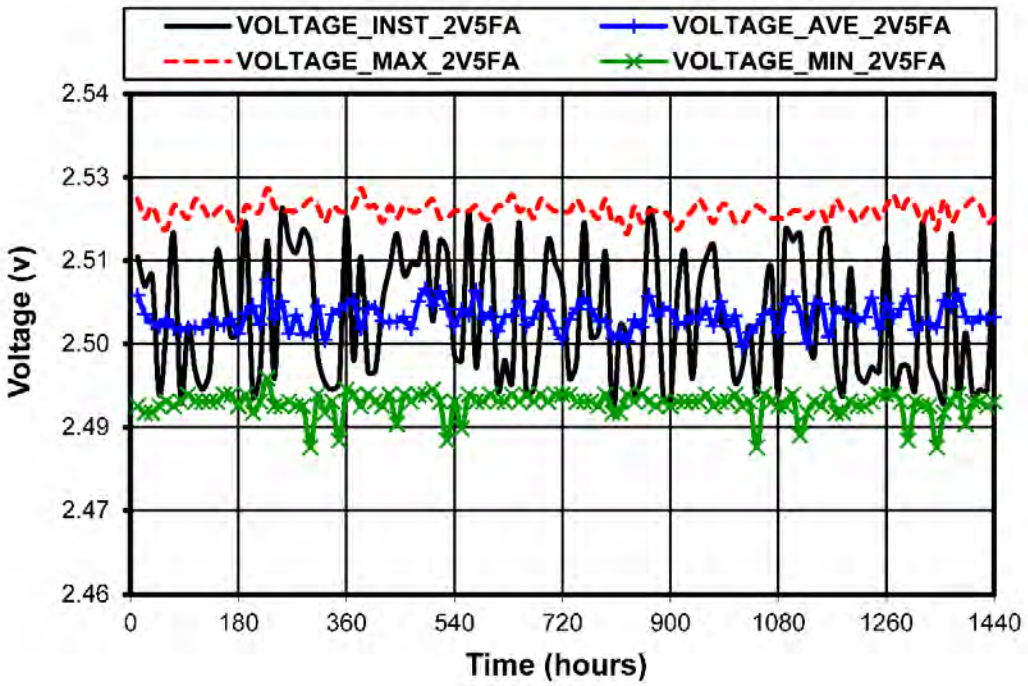

Figure 11-6. Analog 2.5 V rail instantaneous, average, maximum and minimum voltage (Nominal = 2.5 V, Max = 2.625 V, Min = 2.375 V).

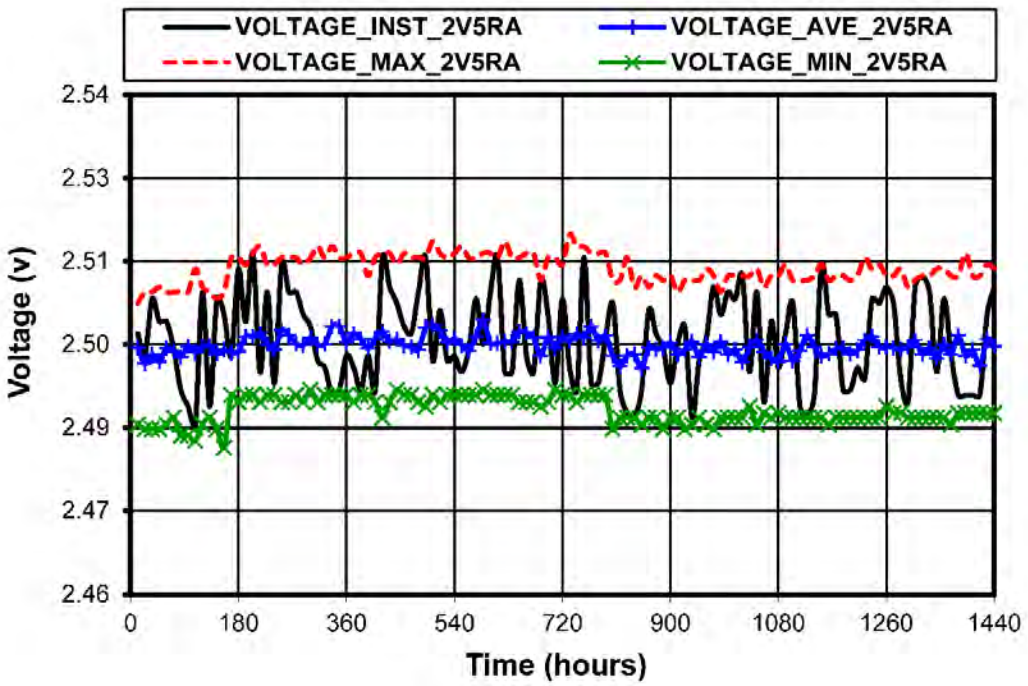

Figure 11-7. Analog 2.5 V rail instantaneous, average, maximum and minimum voltage (Nominal = 2.5 V, Max = 2.625 V, Min = 2.375 V).

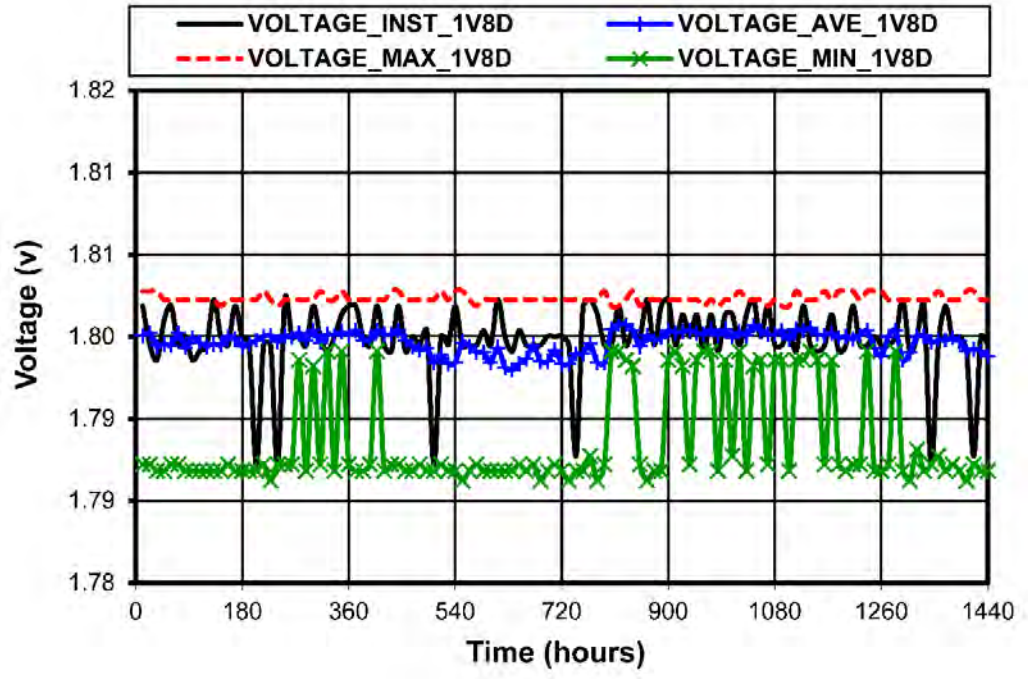

Figure 11-8. Digital 1.8 V rail instantaneous, average, maximum and minimum voltage (Nominal = 1.8 V, Max = 1.89 V, Min = 1.71 V).

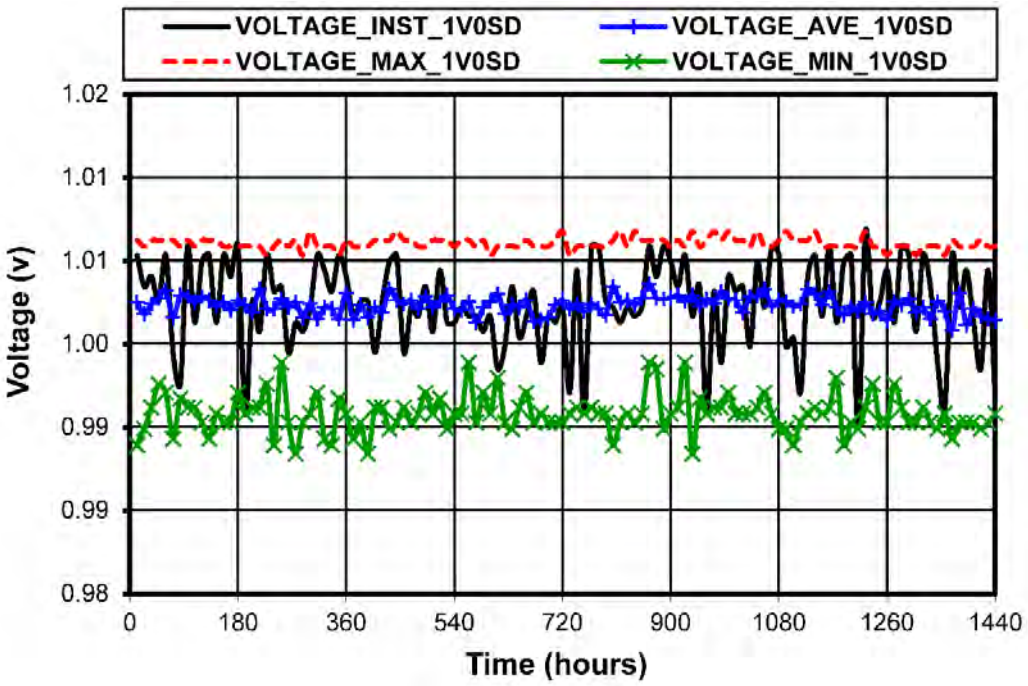

Figure 11-9. Digital 1.0 V rail instantaneous, average, maximum and minimum voltage (Nominal = 1.0 V, Max = 1.05 V, Min = 0.95 V).

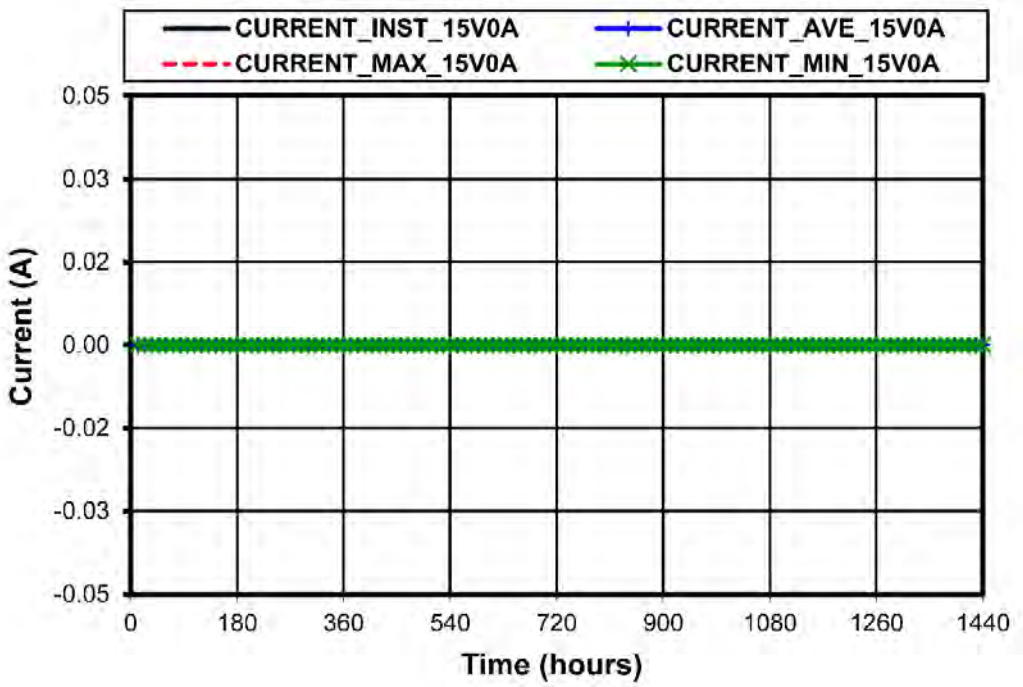

Figure 11-10. Analog 15 V rail instantaneous, average, maximum and minimum current (not used on Artemis).

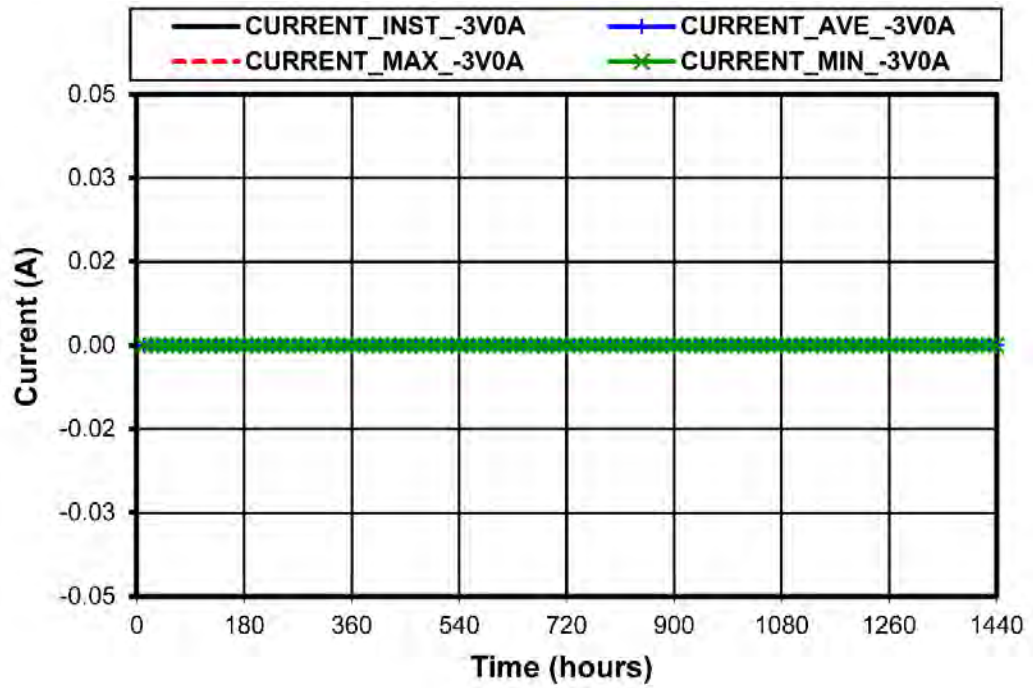

Figure 11-11. Analog -3.0 V rail instantaneous, average, maximum and minimum current (not used on Artemis).

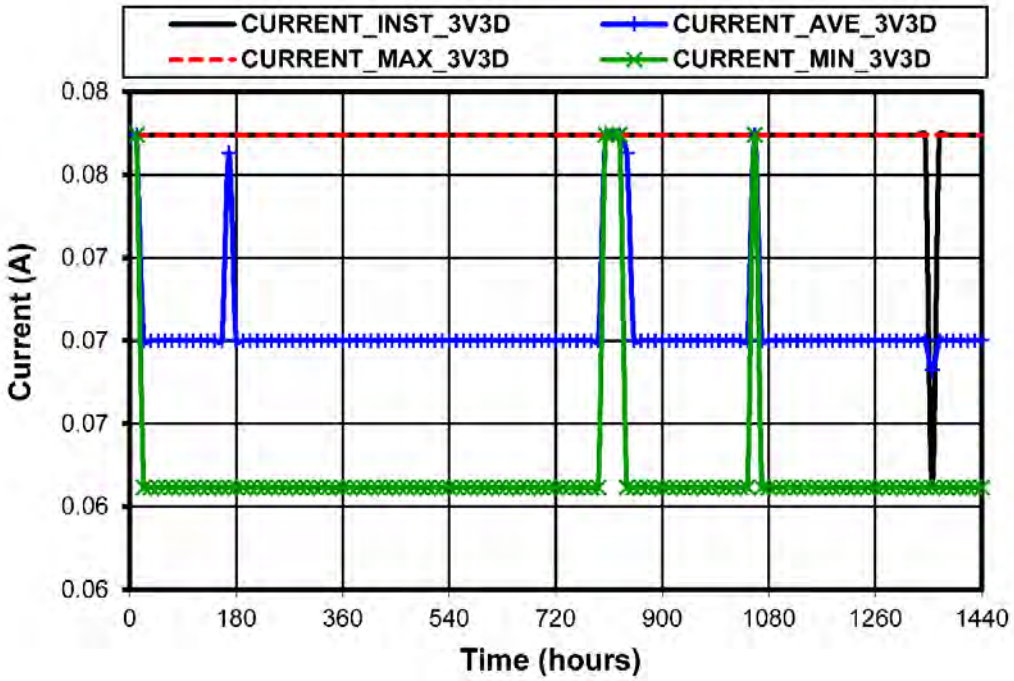

Figure 11-12. Digital 3.3 V rail instantaneous, average, maximum and minimum current (Range: 0.0625 – 0.0781 A).

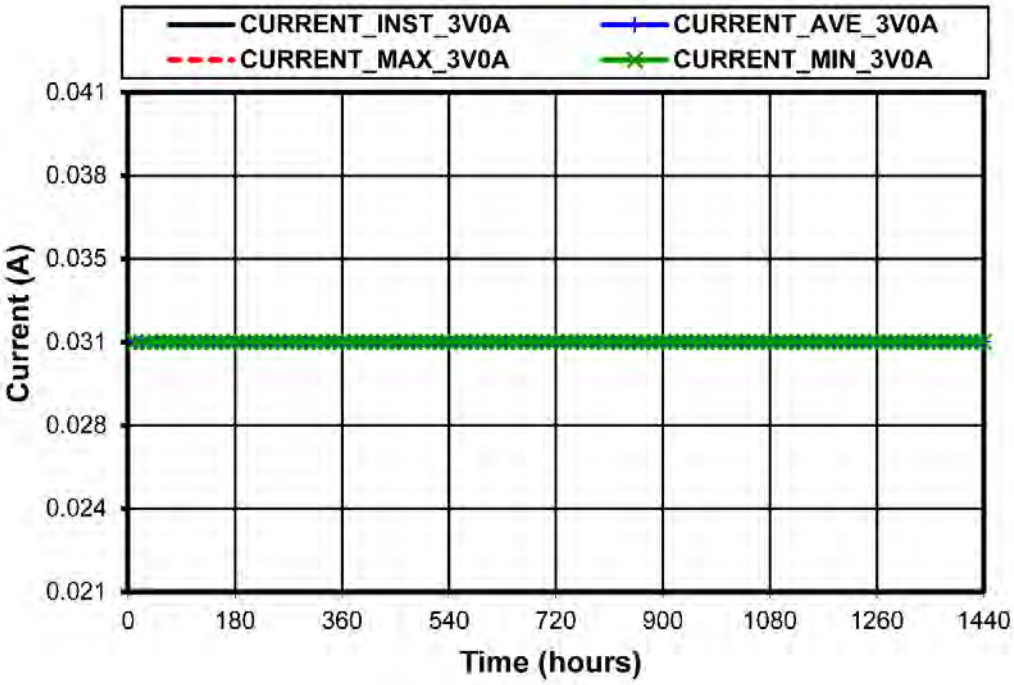

Figure 11-13. Analog 3.0 V rail instantaneous, average, maximum and minimum current (not used on Artemis).

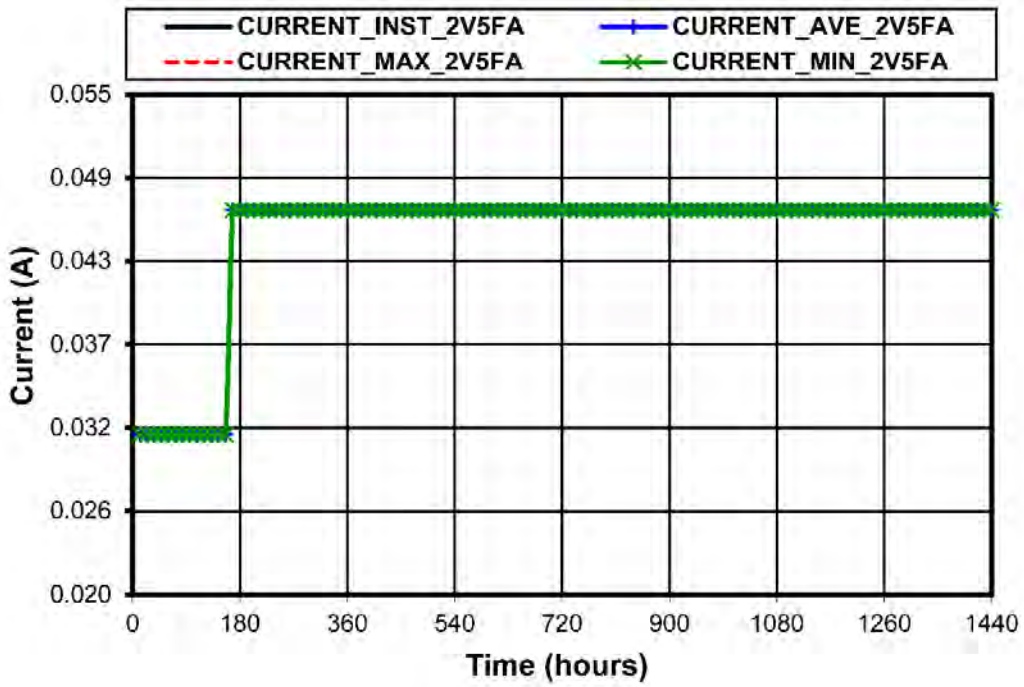

Figure 11-14. Analog 2.5 V rail instantaneous, average, maximum and minimum current (not used on Artemis).

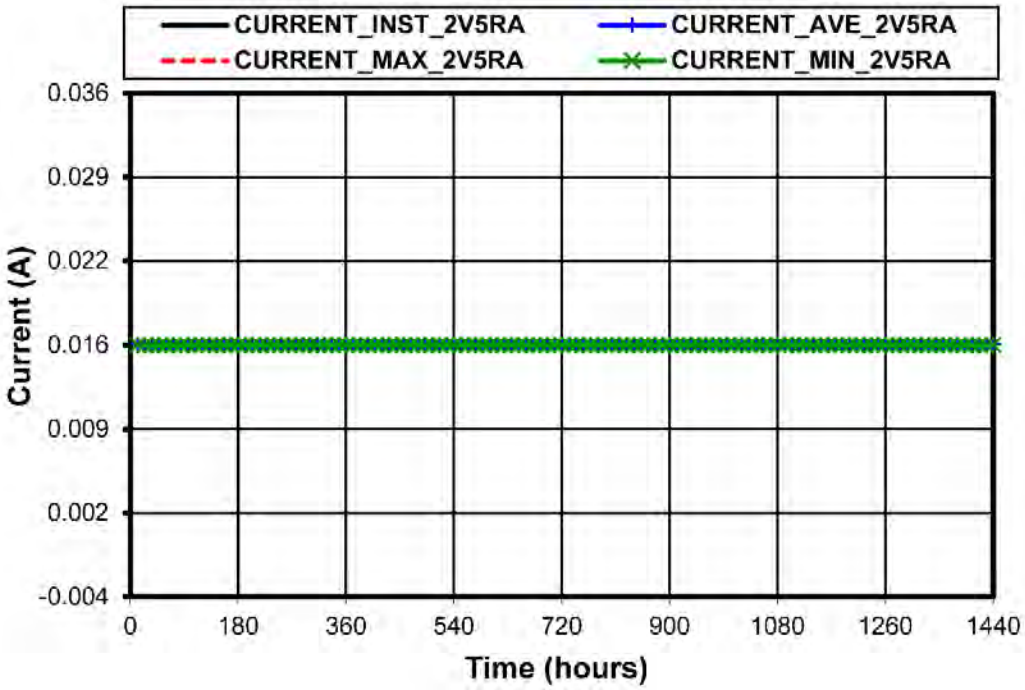

Figure 11-15. Analog 2.5 V rail instantaneous, average, maximum and minimum current (not used on Artemis).

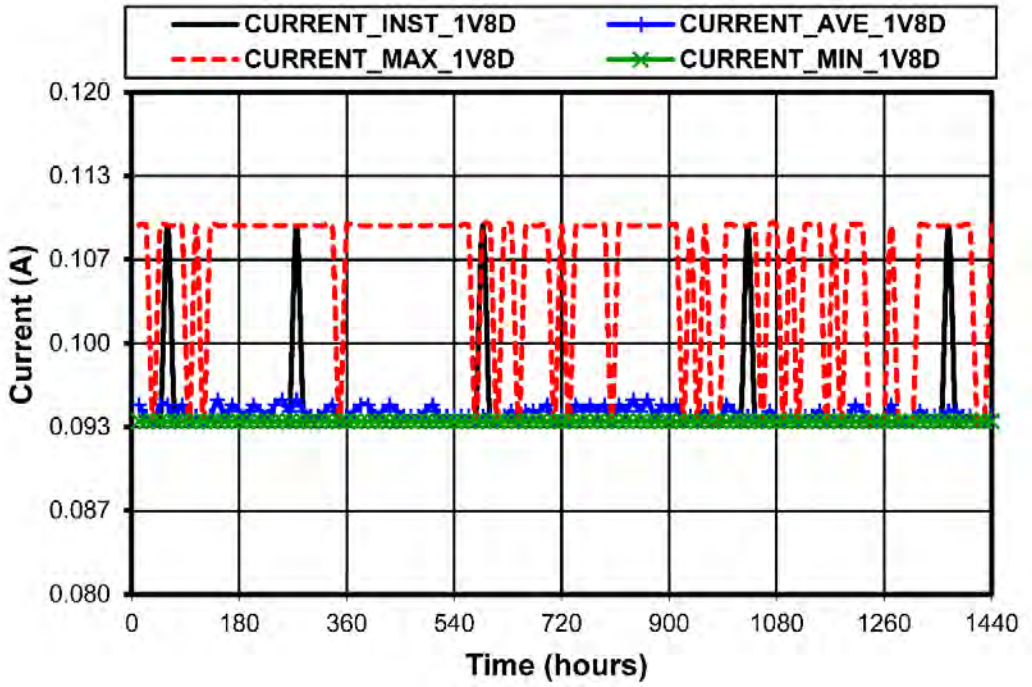

Figure 11-16. Digital 1.8 V rail instantaneous, average, maximum and minimum current (Range: 0.0938 – 0.1094 A).

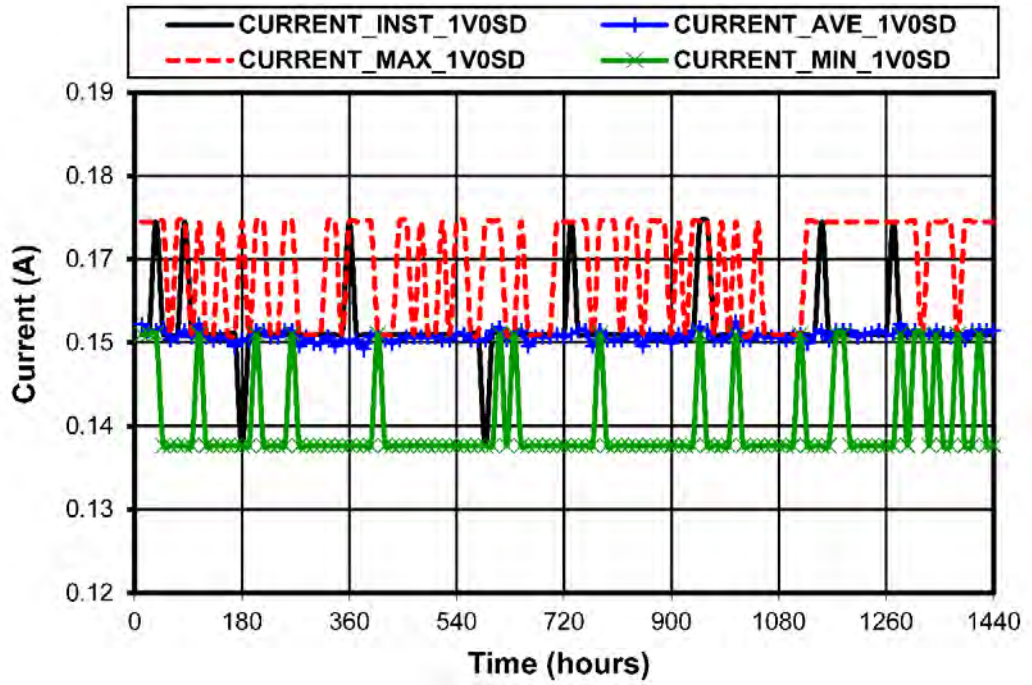

Figure 11-17. Digital 1.0 V rail instantaneous, average, maximum and minimum current (Range: 0.1406 – 0.1719 A).

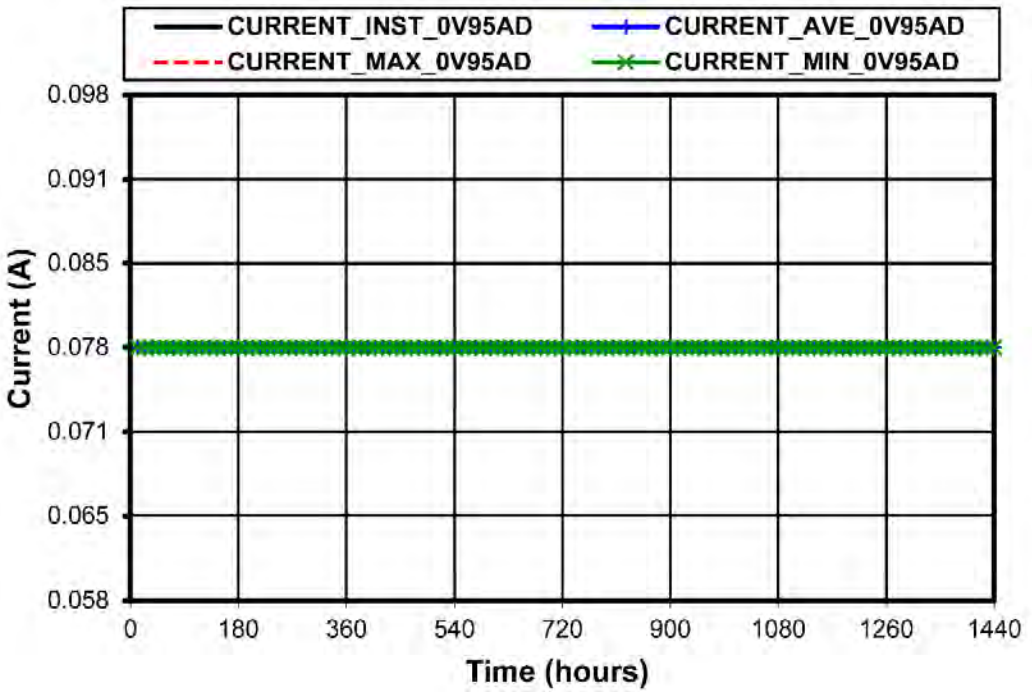

Figure 11-18. Digital 0.95 V rail instantaneous, average, maximum and minimum current (Range: 0.078 – 0.078 A).