EXPERIMENT PLATFORM TO FACILITATE FLIGHT TESTING OF FAULT TOLERANT RECONFIGURABLE COMPUTER SYSTEMS

by

Samuel Andrew Harkness

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, 2015

TABLE OF CONTENTS

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

ABSTRACT

Computers play an important role in spaceflight and with ever more complex mission goals and sensors, current devices are not sufficient to meet the computational requirements of future missions. These challenges are complicated by memory corruption caused by high energy radiation inherent in the space environment. MSU has developed a novel space computing system based on commercial FPGAs to improve performance and reduce cost. This system employs TMR with spares, memory scrubbing, and partial reconfiguration to achieve a radiation hardened, high performance system. This strategy is leveraged on modern fabrication process nodes largely eliminating long term effects of radiation on silicon devices and shifting the focus strictly on memory corruption errors. This thesis improves on the usability of Montana State University's (MSU) existing CubeSat computing research platform through the addition of a robust data-logging system.

1

BACKGROUND

Future Space Computing Requirements

One of NASA's current research interests is in improving the processing power of space computers while simultaneously reducing cost. Data intensive applications generate massive amounts of data, but currently require human analysis and must be transmitted back to Earth. However, the radio downlink is bandwidth limited and not sufficient to handle the volume of data generated. As a result, preprocessing of data aboard the spacecraft is desired. Commercial computer systems have long been capable of meeting the on-board computing performance requirements, but experience system failures due to the harsh radiation environment of space. [18–21]

The Space Radiation Environment

The space radiation environment contains a spectrum of particles possessing energies from 10s of keV up into the TeV range. Specific examples of ionizing particles are high-energy electrons, protons, and cosmic rays. Cosmic rays are heavy ions accelerated to relativistic speeds, ranging across the periodic table. Whether a particles poses a threat depends on the particle energy, particle species, and the material of the device under bombardment, most commonly silicon. [1–3]

Table 1.1: Energy of various particles [1–3]

Particle Type	Energy Range
Trapped protons/electrons	$\leq$ 100 MeV
Alpha particles	5 MeV
Solar protons	$\leq$ 1 GeV
Cosmic rays	TeV

The Earth's atmosphere and magnetosphere provide a significant shield against ionizing radiation. Radiation particles reaching Earth with relatively low levels of energy can be deflected by the magnetosphere. Particles that do penetrate the magnetosphere then encounter the atmosphere, where collisions with gas molecules rob kinetic energy from the ionizing radiation. [2] The result is a gradiant of particles with decreasing energy as they decrease in altitude. As altitude increases, the radiation environment quickly becomes inhospitable to electronic devices, as shown in Figure 1.1.

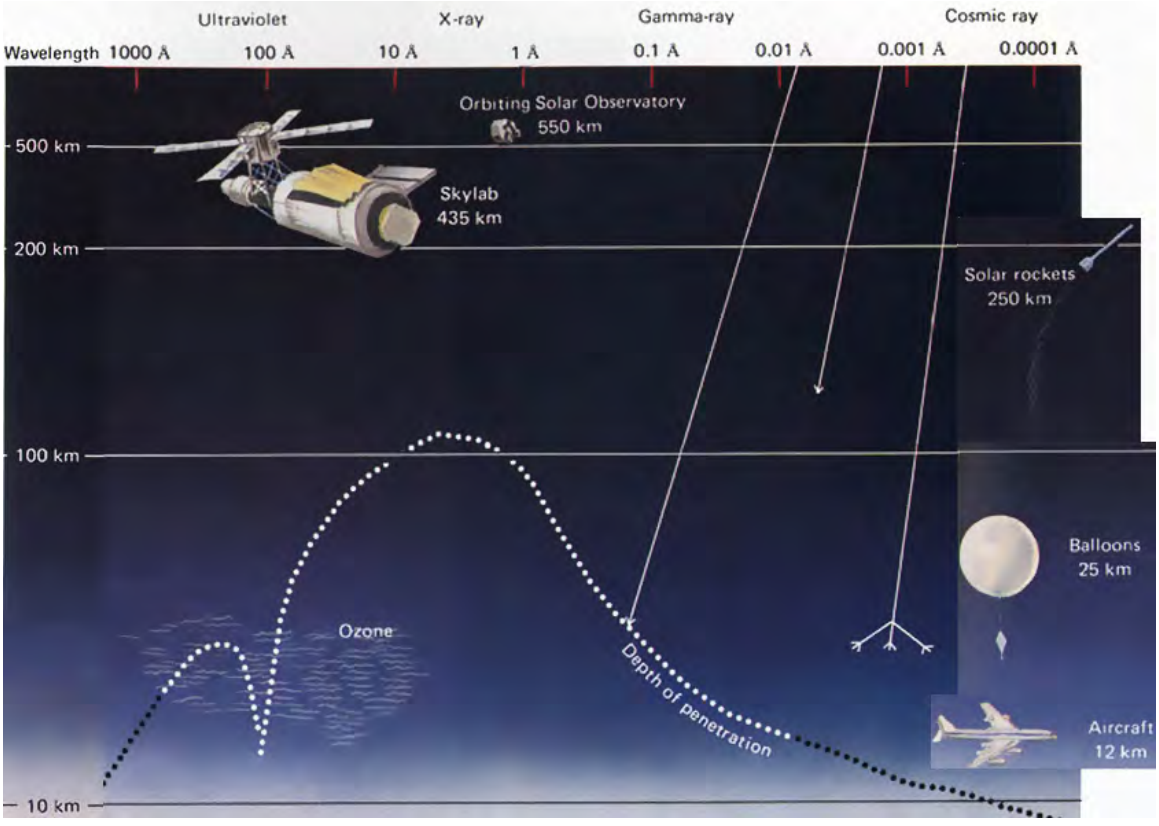

Figure 1.1: Spectrum of solar radiation and the altitude it penetrates into the atmosphere [7]

Of particular interest when placing computers in space/orbit are the Van Allen Belts. The Van Allen Belts are two layers of trapped radiation held in place by the

magnetosphere. The inner belt consists primarily of more massive particles such as protons, while the outer belt holds less massive particles such as electrons unable to 'punch through' the magnetic force to reach the inner belt. Either one or both of the belts must usually be transited for space applications.

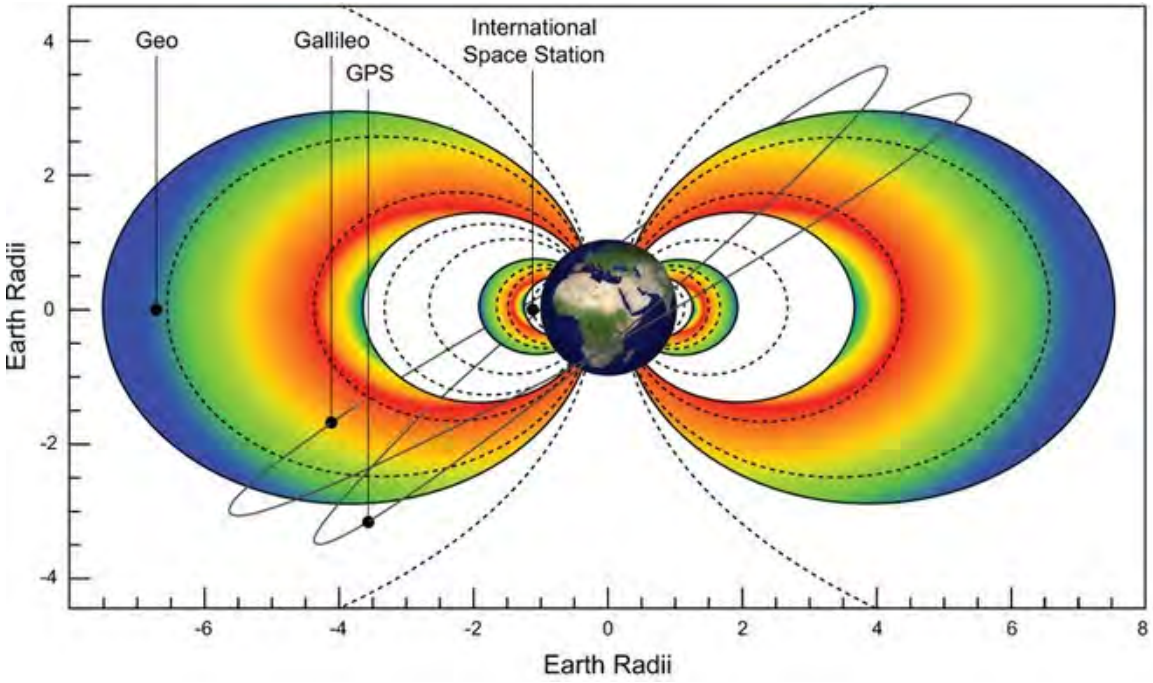

Figure 1.2: Approximate locations of the Van Allen Belts compared to common satellite orbits measured in Earth radii [8]

The Van Allen Belts are symmetric about the Earths magnetic axis, which is tilted from the Earths rotational axis by $\approx 11^{\circ}$. The intersection between these two axes is also 500 km north of the Earths center. This asymmetry leads to the belts not being symmetrical around the Earth. As a result, the Van Allen Belts are further away from the surface near Singapore, and closer to the surface over South America and the south Atlantic. This area is referred to as the South Atlantic Anomaly (SAA), and is a region of greatly increased radiation. [22]

Radiation Effects

Ionization through particle interaction is the process through which electrical charge is deposited in a material as a high-energy particle passes through it. Generally characterized in terms of their kinetic energy, the particles responsible for this ionization are often called energetic or high-energy, indication that said particles have been accelerated to relativistic speeds and possess incredible amounts of kinetic energy. As these particles pass through a material, they collide with the individual atoms of the material and impart their kinetic energy to the material. The amount of energy transferred is dependent on particle energy, the material being transited, and the particle range in the material. [23]

In a semiconductor, the most common substreate used in electronic devices, the amount of energy required to excite a valence-band electron into the conduction band is defined as the band gap energy. [24] Exciting a valence electron to the conduction band is synonymous with generating an electron-hole pair. This newly energized electron is free to move about the material and will do so according to the present electric fields. These electron hole pairs are the source of most radiation-induced effects in modern electronics. The effects of ionizing radiation on silicon devices can be broadly divided into two categories: cumulative effects and transient effects.

Total Ionizing Dose

The rate at which cumulative effects build up is a function of the severity of the radiation environment and the duration of exposure. Total Ionizing Dose (TID) is a measurement of the energy deposited in a material per unit mass, represented by unit rad. The survivable TID of a device is often specified in $kilo - rad$ or $krad$. [1]

TID is a permanent degradation of a semiconductor device operating in a radiation rich environment. TID is raised when a low-energy particle becomes trapped inside a semiconductor structure, modifying the material properties. The primary cause of TID-related failure is trapped charge in insulator layers. [25] Trapped charge in a metal-oxide-semiconductor field-effect transistor (MOSFET), the most common transistor used in modern computers, can skew threshold voltages, resulting in timing failures. Should sufficient charge accumulate in the gate oxide of a MOSFET device, a conduction channel will be induced as if a control voltage was applied to the gate terminal, and the MOSFET is effectively permanently active. An N-channel MOSFET in this state is always on, while a P-channel MOSFET would be always off.

As charge accumulates in field oxides, leakage currents can begin to flow between transistors. [26] Though this doesn't necessarily impact the functionality of the device, it will negatively impact the efficiency of the system. Increased leakage currents increase thermal output and power draw, leading to a decrease in system runtime if powered by a finite storage element, such as batteries. TID effects are non-repairable, though failure states can be delayed by reducing timing constraints.

Single Event Effects

Transient effects concern the initial interaction of the ionizing particle with the device under bombardment, and the immediate aftereffects. Single Event Effects (SEEs) are the aftereffects of the interaction of a high-energy particle and a semiconductor device. A SEE is the result of a single strike on a device in a vulnerable region, creating a path of electron-hole pairs. Electron-hole pairs generated in this manner will quickly dissipate through the process of recombination, manifesting as a

short-lived current or voltage within the circuit. Depending on the time and place a SEE manifests, it can be further categorized.

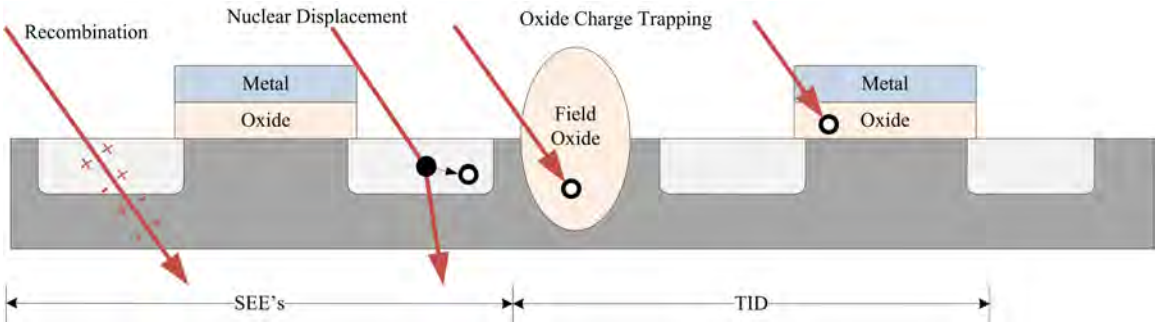

Figure 1.3: Cross section of a device showing the spatial differences between SEEs and TID [9]

The most basic SEE is a Single Event Transient (SET). A SET is the previously described transient current or voltage inherent to all SEEs. A SET can be as short-lived as a few hundred picoseconds. [27] A SET occurring at the input to a synchronous, bistable circuit element (such as a flip-flop) within the setup and hold window has the potential to be latched into the system, creating an erroneous reading. Such an occurrence is defined as a Single Event Upset (SEU). An SEU can also occur if a radiation strike occurs at a specific place and causes a memory value to change, regardless of timing. Corruption of memory elements resulting in undesirable behavior or performance degradation is called a Single Event Functional Interrupt (SEFI).

Parasitic bipolar junction Transistors (BJT) are created between adjacent NMOS and PMOS devices as an unavoidable side-effect of CMOS fabrication processes. The activation of these BJTs can create a low-impedance path between V_{CC} and V_{DD}. An SEE which activates one of these parasitic BJTs is defined as a Single Event Latchup (SEL). The best-case SEL scenario is an observed increase in power consumption. Should the parasitic transistor enter a thermal runaway condition, it is possible

to exceed current/temperature thresholds and permanently damage the device. [28] Single Event Burnout (SEB), single event gate rupture, and single event snapback are other destructive SEEs more common to vertical semiconductor processes. [28]

Current Solutions to Mitigating TID

The most obvious way to mitigate radiation effects is to use shielding. Unfortunately, shielding is inappropriate for space applications because of its additional weight, and the cascading reactions within the shield due to high energy particles. Figure 1.4 illustrates the diminishing returns for reduced dosage vs thickness of aluminum shielding. Shielding inefficiency becomes even more pronounced for higher energy radiation (100 MeV - 1 GeV). Some cosmic rays even have sufficiently high energy to penetrate any amount of shielding that could be reasonably flown on a spacecraft. [2] In fact, shielding may actually worsen the radiation environment, as a radiation particle may generate a secondary stream of radiation as a result of collisions with the shielding material, as shown in Figure 1.5. Instead of shielding, radiation hardened computers have attempted to fix the problem at the physical layer.

Radiation-Hardened-By-Design (RHBD) uses non-standard layout techniques to minimize the probability of charge being trapped in insulating structures. Figure 1.6 is an example of using guard rings to minimize the area of susceptible insulating regions and provide alternate conduction paths for errant charge carriers to flow. [12] SELs can be mitigated by using isolating trenches.

Another historical technique to mitigate radiation effects is the use of non-standard materials or semiconductor processes. Radiation-Hardened-By-Process (RHBP) fabricates devices with materials that are less susceptible to charge trapping and/or produce fewer charge carriers in the event of a radiation strike. Silicon-On-

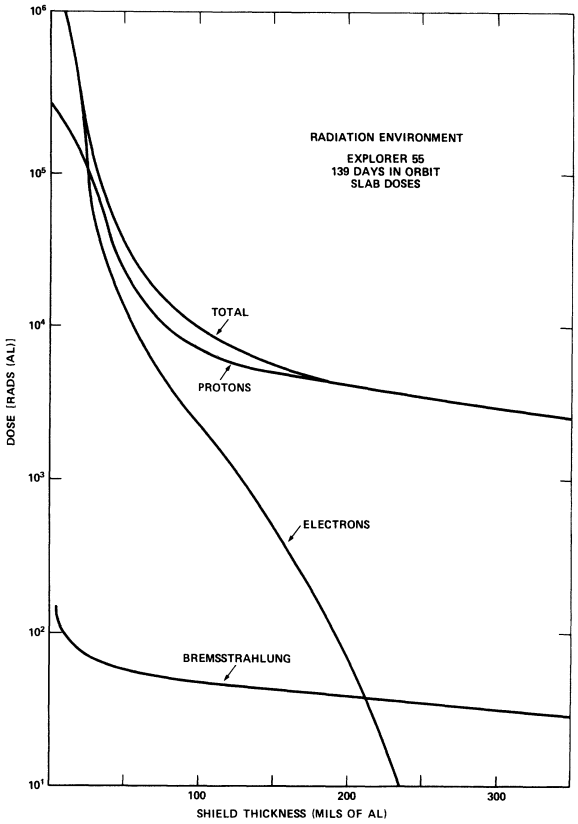

Figure 1.4: Dosage from protons, electrons, and Bremsstrahlung measured through Aluminum shielding [10]

Insulator (SOI) is a technique where the transistor structure is built on top of an insulating layer, reducing the total path length of a penetrating particle, with the intended result of reducing the number of generated electron-hole pairs. [29, 30]

Unfortunately, RHBD and RHBP have multiple drawbacks. First, additional circuit area is required, lowering the performance and power efficiency compared to standard designs and processes. Second, non-standard approaches and low-volume manufacturing are more expensive. For example, the BAE RAD750 used on the Mars Curiosity Rover costs $200,000. [31]

Radiation hardened processor performance lags behind commercial equivalents by approximately 10 years, as shown in Figure 1.8. This performance gap, in addition

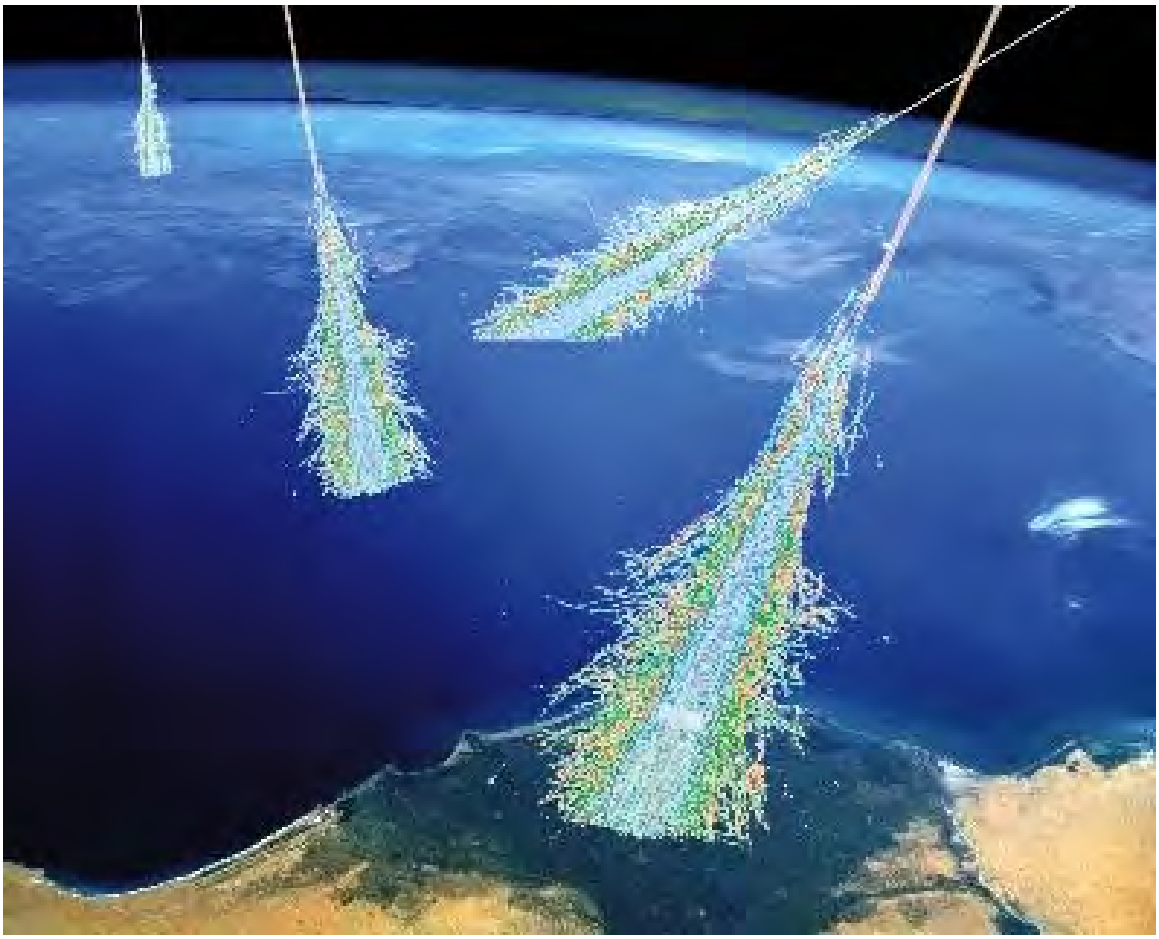

Figure 1.5: Scattershot effect due to interaction of cosmic rays with a shielding material e.g. atmosphere [11]

to the high cost of custom products, has created a demand for different approaches to radiation tolerance.

Current Soltuions to Mitigating SEE

Fortunately, modern semiconductor processing has reduced failures due to TID as a side-effect of shrinking feature size. Both the amount of charge buildup in an oxide and the resultant threshold voltage shift are largely proportional to oxide thickness

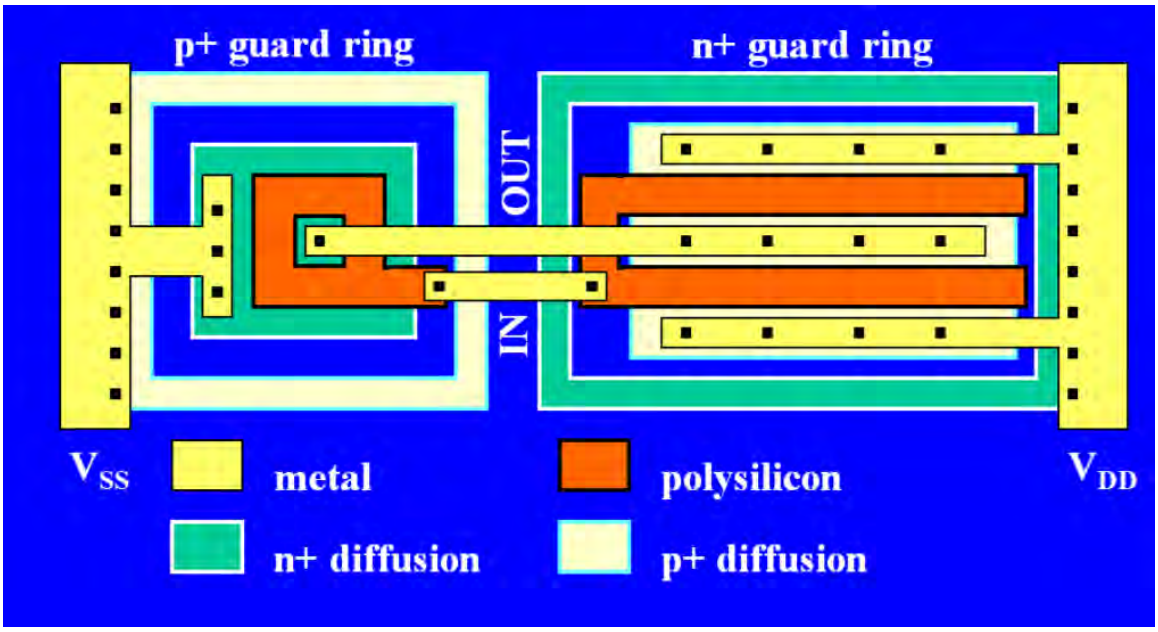

Figure 1.6: RHBD inverter using isolated transistors with guards rings [12]

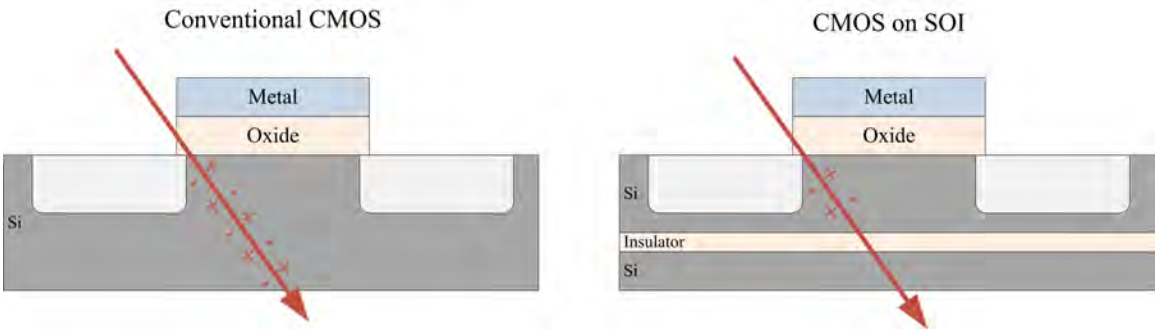

Figure 1.7: Traditional CMOS transistor & RHBP CMOS transistor using SOI [12]

due to the lowering of the statistical probability of charge being trapped in a reduced volume. [26] For example, the Xilinx Virtex-6 implemented in 40 nm technology has achieved 380 $krad$ inherently and 2 $Mrad$ with reduced timing. [18] This amount of TID immunity is suitable for the majority of orbital missions and historically has been achieved through expensive, custom processing of the semiconductor substrate. Therefore, the primary area of study today is SEE mitigation.

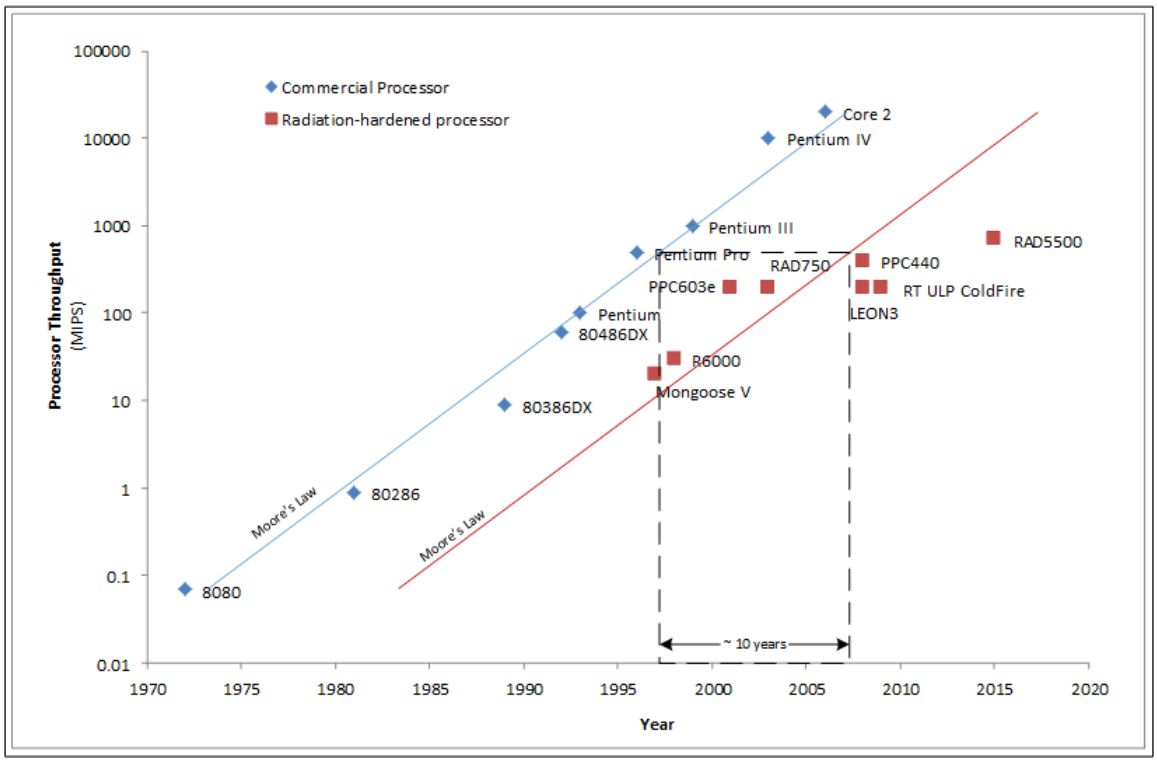

Figure 1.8: Performance of commercial and radiation hardened processors [13]

As SEE mitigation techniques at the manufacturing level lead to slow and expensive computers, system level approaches are more appealing. A common system level technique is Triple Modulo Redundancy (TMR). [32–35] TMR systems run three identical copies of a component in parallel. The majority result of the three systems is taken as the true output. In the event of an SEU, a recovery procedure is typically required to recover the faulted circuit to an operational state. TMR can be implemented at nearly any level, ranging from bit level triplication of circuits to system level triplication of major electronic components. [36] The beginnings of TMR are rooted in early theoretical work concerning how reliable computing machines could be created from inherently unreliable parts. [37]

Another technique for radiation hardening a system is to ensure the integrity of memory contents through the use of memory scrubbing. In this process, the contents of memory are periodically checked against known good data (aka a golden copy), located in a more radiation-safe memory such as PROM or EEPROM. This prevents the accumulation of errors in memory, reducing the likelihood of using corrupted values in computations. There are two common types of scrubbing. Blind scrubbing rewrites the memory location with the golden copy, regardless of its current value. Readback scrubbing first checks the memory to see if a write operation is necessary. Readback scrubbing is more complex, but can save time statistically if a write operation takes more cycles than a read operation.

FPGA Basics

Field Programmable Gate Arrays (FPGAs) are semiconductor devices on which the function can be defined after manufacturing. An FPGA enables programming of product features and functions and reconfiguration of hardware for specific applications even after the final product has been installed. FPGAs allow flexibility in designs without introducing a large amount of delay into the design schedule. [38] This is achieved by having programmable interconnect and logic blocks, which allow hardware changes with software-like design times. [39]

Radiation Effects Mitigation in Commercial FPGAs

In a traditional microprocessor, a memory upset may result in an incorrect instruction execution or a corrupted computation. The same can happen in a FPGA, but errors occurring in configuration memory manifest as changes in the physical circuitry implemented on the device. Due to this transistor level weakness, SEEs must be mitigated at the configuration layer and application layer. The configuration layer

contains hard logic cores and administrative circuits used to define the behavior of the device. The application layer contains user-defined circuitry, application memory, and the interconnect between the two.

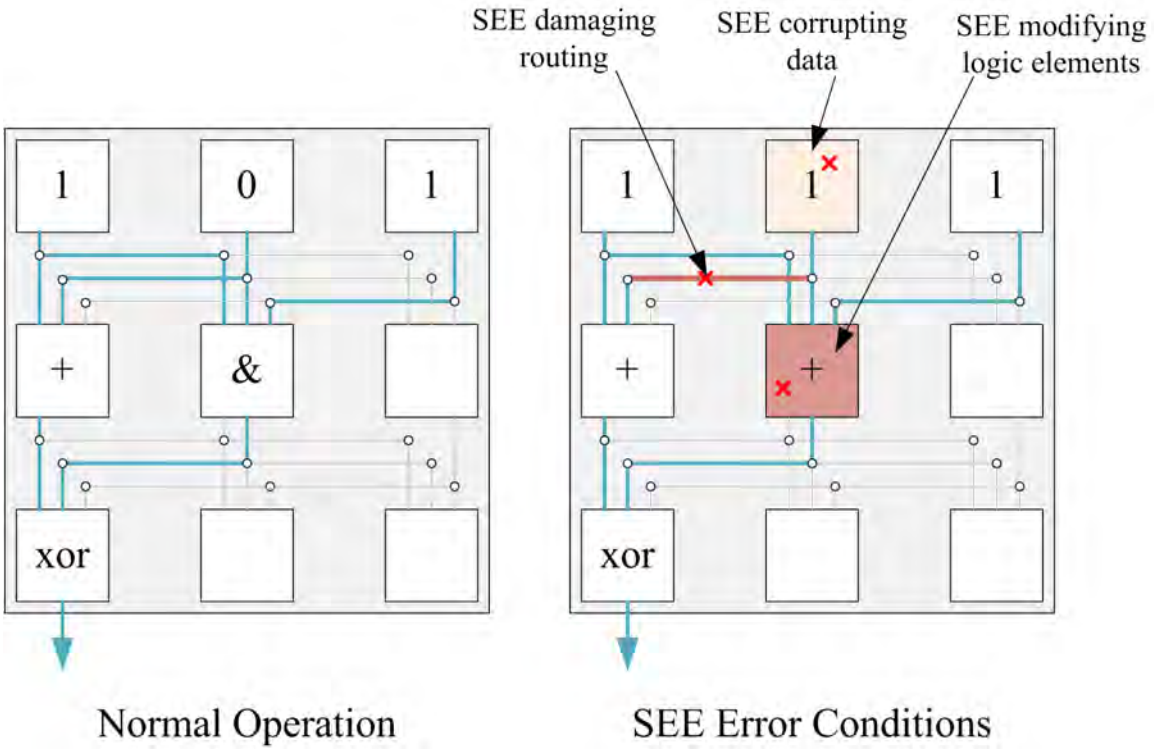

Figure 1.9: Simplified FPGA logic blocks and faulted conditions [9]

As stated earlier, programmable logic devices are defined by data contained in configuration memory. An SEU occurring in configuration memory alters the corresponding logic element. When such a change occurs in an occupied area of the design, the system will cease proper operation. For example, a SEU in a logic element could change an AND gate to an OR gate. In system interconnect, connections could be destroyed or added. Either is likely to adversely affect system operation. This logic element will continue improper operation until the corrupted configuration memory is

overwritten with correct data, either through a full device configuration, or through a partial reconfiguration. [40]

SEUs may also occur in user memory. Such an error may be as limited as incrementing an integer by one, or as catastrophic as sending a hardware state machine into an unrecoverable state. The consequences of memory upsets are very diverse and system dependent. As there is no golden copy for user memory, scrubbing is not an option to mitigate SEUs. Instead, scrubbing is used to maintain the reliability of configuration memory in FPGA systems. [41]

The most widely adopted technique for FPGA SEU mitigation is a combination of TMR, fault detection and the prevention of errors propagating through a system, and configuration memory scrubbing, preventing faults from accumulating in the TMR system. The combination of these techniques is commonly referred to as TMR+scrubbing. In addition, FPGA manufacturers have included system architecture to assist in soft error mitigation. [42]

MOTIVATION

MSU's Contribution to Space Computing

The focus of this research has been to build upon the traditional fault mitigation techniques in an effort to increase the performance and reliability of FPGAs for aerospace applications.

Partial Reconfiguration as SEE mitigation

Partial reconfiguration (PR) is an advanced FPGA technique where a portion of the FPGA is reconfigured while the rest of the fabric continues to run unaffected. In terrestrial applications, partial reconfiguration allows for more precise control of an FPGA. For example, power draw can be reduced during periods where more advanced functions (such as floating point operations) are unneeded. Or, limited fabric resources can be reused by different functions during different modes of operation.

In this architecture, an FPGA is partitioned into discrete, identical partial reconfigurable processing resources. These are referred to as 'tiles', and they represent the granularity of the TMR implementation. The control system selects 3 of the tiles to run in TMR. In the case of the TMR voter detecting an error, the system can be halted and the corrupted tile repaired via PR. On completion of PR, the system resumes operation.

If enough resources exist, extra tiles can be defined to act as spares. In the event of a fault, the corrupted tile is taken out of the active TMR triad and a spare brought online in its place, significantly reducing the amount of time the system is halted. Then the corrupted tile can be repaired in the background using PR and reintroduced to the system as a healthy spare. This system is referred to as

TMR+spares. Markov analysis has shown that one or two spares are sufficient for radiation hardening purposes in a practical radiation environment. [15]

Radiation Sensor

In addition to the TMR+Scrubbing+Spares architecture, Montana State University developed a two-dimensional silicon based radiation sensor to provide environmental feedback for the system. Using the spatial location of a potential radiation strike, a SEU can potentially be preemptively repaired using partial reconfiguration or memory scrubbing. Long-term sensor data can also be used to provide knowledge of radiation conditions.

The sensor is created with P-type regions on the upper surface of the silicon and N-type regions on the bottom of the sensor, resulting in a PN junction. When a radiation strike occurs, it leaves a path of electron-hole pairs. If not allowed to recombine in place, these charge carriers are attracted to opposite sides of the silicon and result in a transient current (Figure 2.1). [43]

To increase the sensitivity of the sensor the PN junction is reverse biased, widening the depletion region. When fully depleted, the bias voltage is sufficient to increase the depletion region to its maximum width, minimizing recombination and maximizing sensor sensitivity. [14, 43]

To create a two dimensional sensor and assuming 'ideal' radiation strikes (i.e. the particle trajectory is roughly perpendicular to the surface of the sensor), the doped regions are created in strips on the top and bottom surfaces of the sensor (Figure 2.3). A strike that has sufficient energy to pass entirely through the sensor produces a current on the top of the sensor as an X coordinate, and a current on the bottom of the sensor as a Y coordinate. [14, 43]

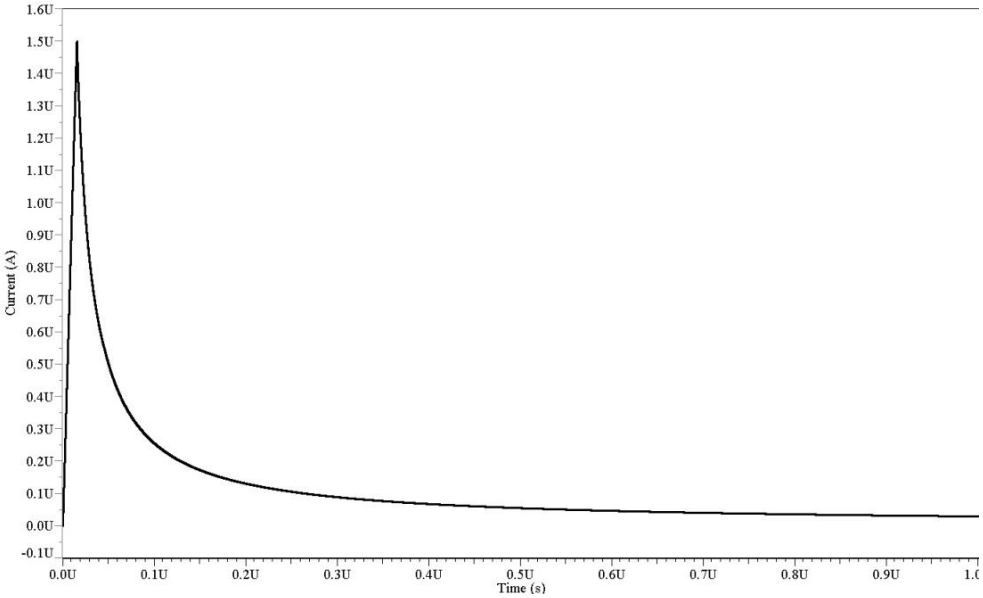

Figure 2.1: Back side (Electron) current produced out of the sensor for a 21.8 MeV Kr strike [14]

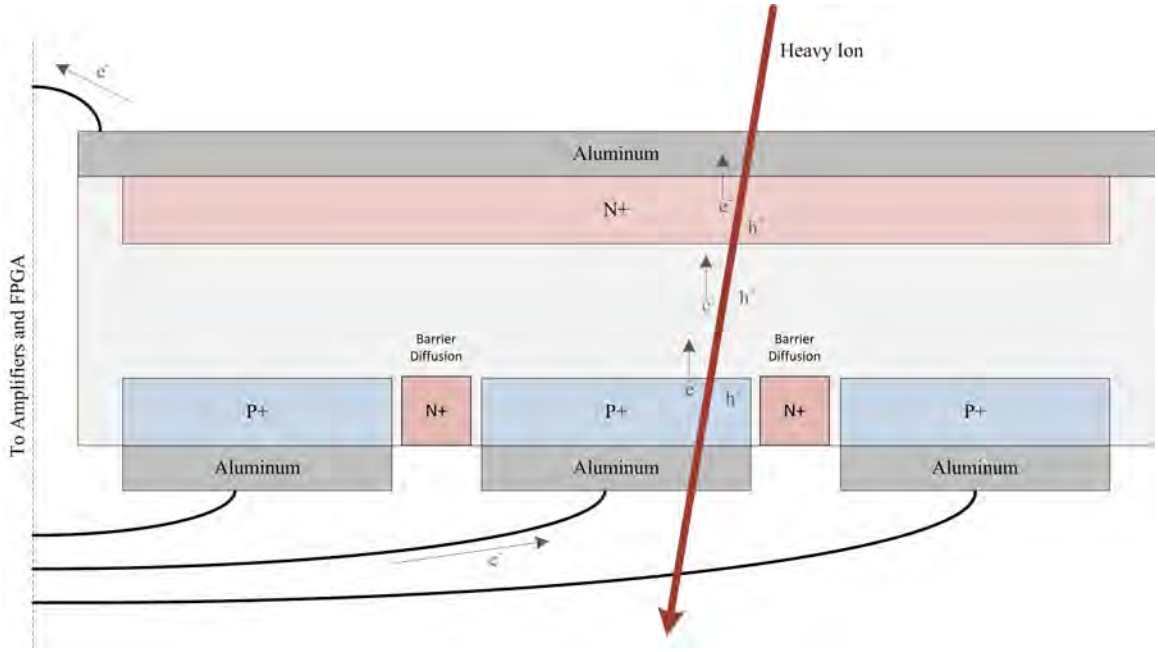

Figure 2.2: Cross section of a radiation strike in the radiation sensor [9]

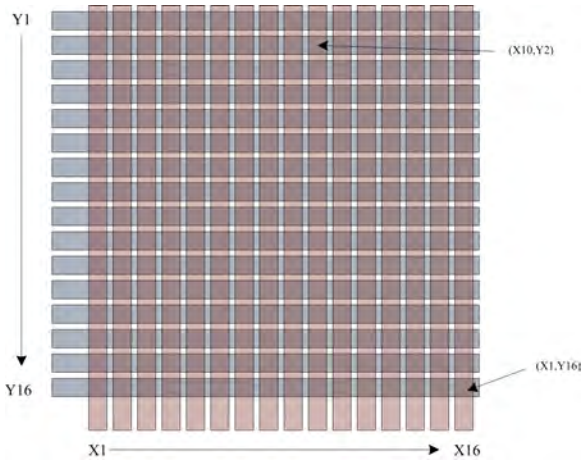

Figure 2.3: Two-Dimensional Spatially Aware Radiation Sensor Layout [9]

The Need for High Altitude Testing

In previous revisions of this architecture, benchtop and laboratory testing was performed to simulate a radiation strike. Cyclotron testing of the system was performed at the Texas Cyclotron Facility using 25 MeV Krypton and Argon beams. This testing utilized a custom aperture that allowed the beam to be isolated to a specific XY coordinate. This setup tested the sensor's ability to detect radiation, and the FPGA system's ability to respond to strikes detected by the radiation sensor.

Unfortunately, it is impossible to produce the energy levels necessary to empirically test the system's SEE immunity on Earth. Cyclotrons and particle accelerators cannot reproduce the space environment accurately and are typically used to bombard sub-systems in modified form factors. The Aluminum lid on most commercial FPGAs is enough to protect the silicon from strikes, and is removed before testing to simulate higher energy bombardment. [18] These types of tests do not meet the requirements of a full system prototype demonstration in an operational environment.

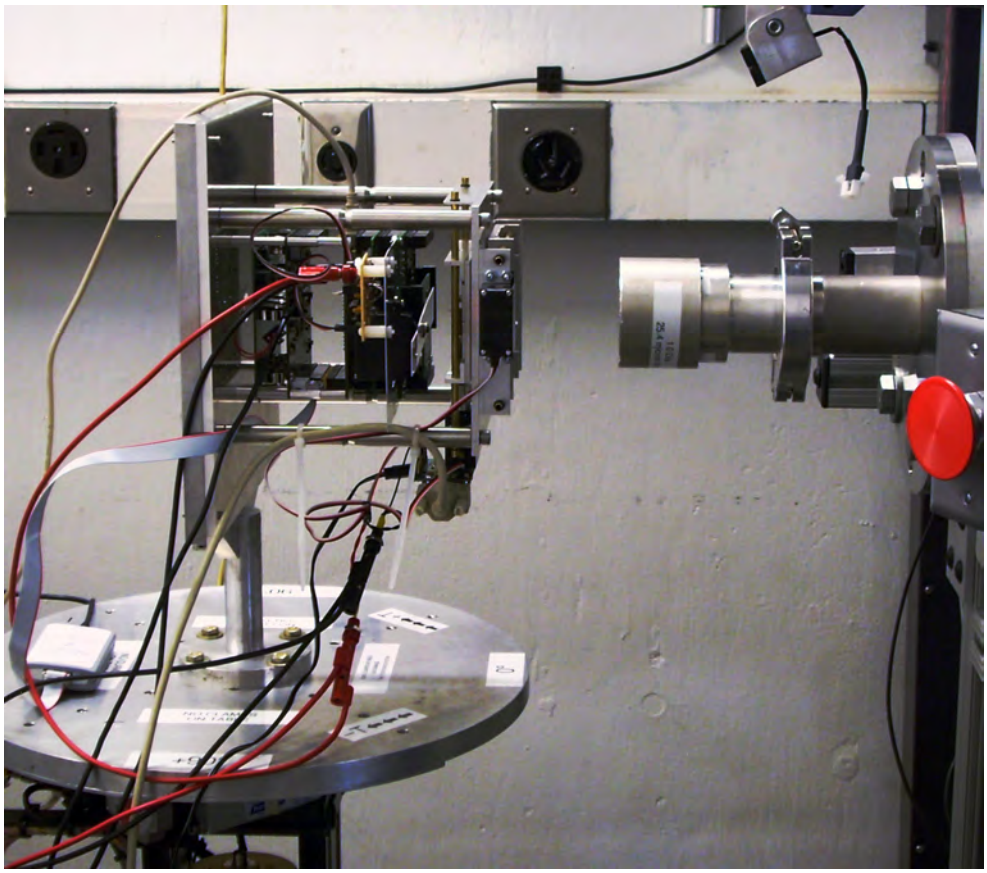

Figure 2.4: Prototype hardware stack mounted normal to cyclotron beam [9, 15]

To approach a representative radiation environment, Montana Space Grant Consortium Borealis research balloons were utilized in 2012 and 2013 to test the radiation tolerant design of the computer system. Radiation strikes were logged on these missions, but due to limited flight time and altitude only a handful of potential faults were identified.

The Borealis balloon flights were followed each year by a flight on NASA & Louisiana State University High Altitude Student Platforms (HASP) zero pressure balloons. These balloons tested system performance and duration capabilities on 10-14 hour float times at 120,000 ft. Several potential radiation strikes were gathered during these flight due to the extended float time at altitude. The 2012 flight results

Figure 2.5: Research Balloon Launches [9, 15]

were impeded by an insufficient power system. Upon being struck, the power supplies of the sensor amplifiers were unable to handle the current transients associated with the voltage level switches. These transients, in turn, caused instability in the voltage regulators due to insufficient phase margin in the regulator feedback path. Combined with other EMI issues present on the power board (coupling between regulator circuits), these transients caused other voltage rails to also collapse temporarily. This collapse was sufficient for loss of system functionality, but not long enough to trigger a full system reset. Figure 2.6 shows a FPGA core voltage collapsing after a simulated radiation strike on the radiation sensor. [15]

Unfortunately, no recorded radiation strikes resulted in FPGA configuration corruption. Configuration readback was not available during the flight, so it is possible that unused configuration bits were upset.

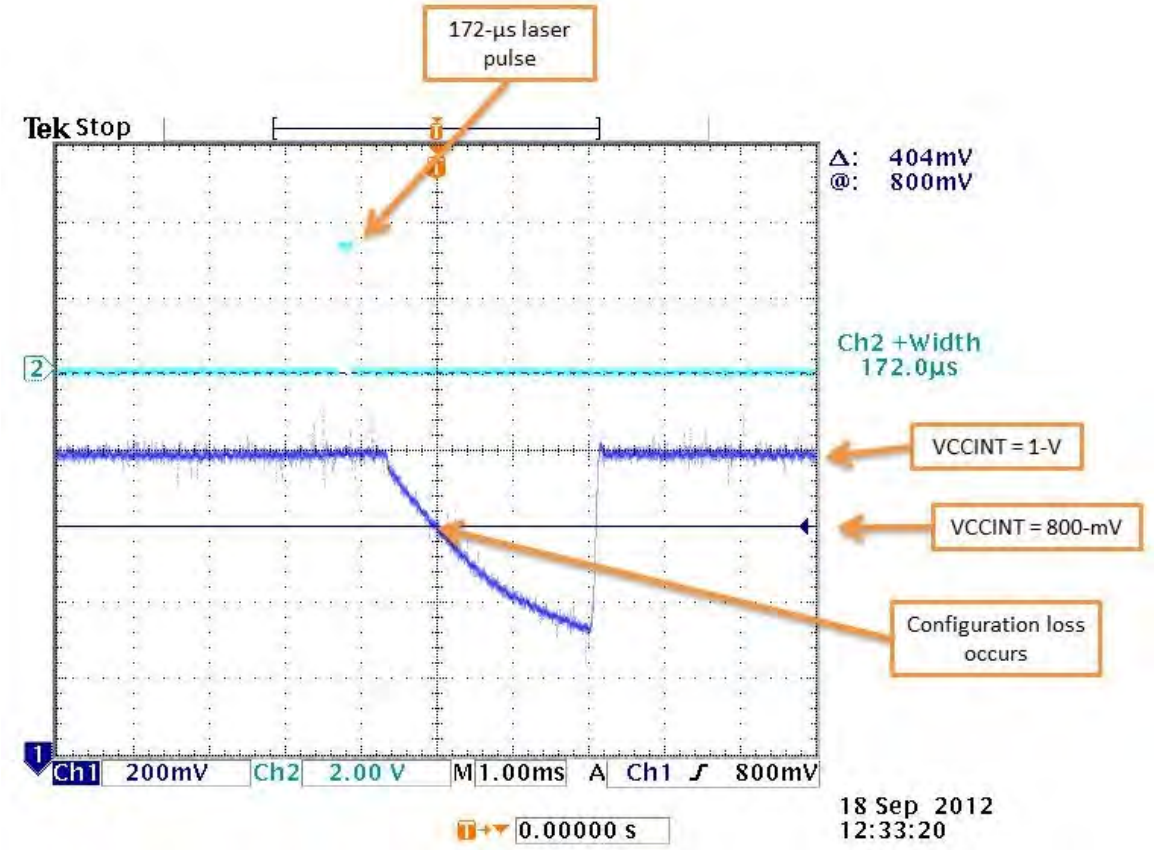

Figure 2.6: Collapse of a FPGA core voltage following a simulated radiation strike [15]

While high altitude balloons provide an easily accessible platform for flight testing, they do not reach a sufficiently high altitude to expose the system to a representative radiation environment, nor do they provide sufficient time to perform a comprehensive evaluation of the system's reliability. Sounding rocket tests provide sufficient altitude (100-300 km) for radiation exposure, but provide even less time for reliability testing. Ground based testing can provide infinite time for reliability tests, but effectively zero radiation exposure.

The requirements of high-altitude and long duration flights can only be met by a satellite mission. This research is focused on developing an experiment platform that is easily integrated with standard CubeSat mechanical and electrical interfaces. Our

research group received funding to prepare a computer system for a mission to the International Space Station, which is encapsulated in the radiation tolerant computer mission to the International Space Station (RTcMISS), pronounced Artemis. Artemis is envisioned to be an application-agnostic experiment platform, with robust user configurability, control, and feedback.

SYSTEM DESIGN

Previous iterations of the radiation hardened computer system were designed on a stacked printed circuit board (PCB) design. Functions of a complete system were divided into discrete areas of responsibility on separate PCBs, and then joined using standardized stacking connectors. To meet the requirements of a CubeSat mission or interfacing with the NanoRacks system aboard the ISS, a CubeSat form factor was adopted. [44] The boards in the stack are an FPGA board, where the experiment platform is located as well as all control and data logging functions, a radiation sensor board with amplifier circuitry, a power conditioning board to provide all voltage rails required by the FPGA board and radiation sensor board, and a board specifically for interfacing with the ISS.

FPGA Board

The FPGA board contains the experiment FPGA (Xilinx Virtex-6) and the system controller FPGA (Xilinx Spartan-6). Figures 3.5 and 3.6 show an annotated view of the physical FPGA board and a block diagram of the board I/O. When the power rails have stabilized, the Spartan-6 is automatically programmed by a user-generated bitstream on the Xilinx Platform Flash. This bitstream also includes the C code for the MicroBlaze soft-processor running on the Spartan-6. After the Spartan-6 is successfully programmed, LED1 will light up. The Spartan-6 then accesses the user-generated Virtex-6 bitstreams located on the FPGA SD Card to program the Virtex-6. LED2 will light up following successful configuration of the Virtex-6.

Figure 3.1: Artemis stack in CubeSat chassis

Virtex-6

The Virtex-6 FPGA is set up for a TMR+Scrubbing+Spares configuration. Three 16-bit buses (1 from each active tile) are output to the Spartan-6 using 2.5 V I/O. A 2-bit bus (Health_Tile) is the output of the voter on the Virtex-6 fabric. The voter outputs the number of the tile which is currently faulted (0x0,0x1,0x2) or 0x3 if all voter inputs agree. The TMR triad is controlled by the 12-bit active_tiles bus. the

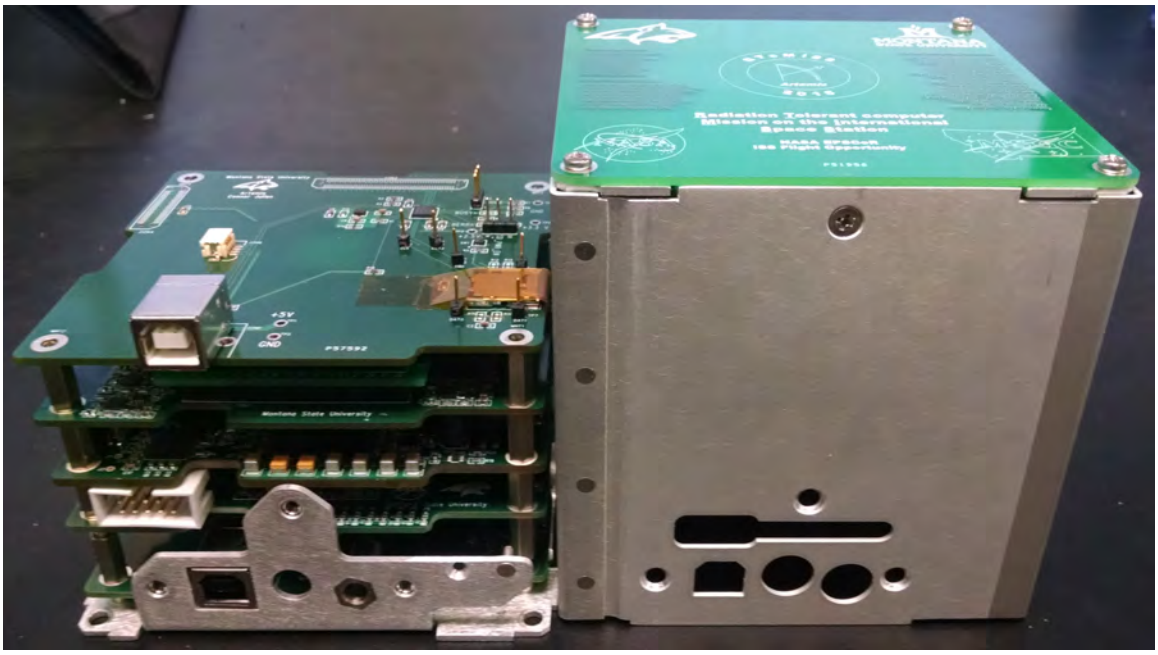

Figure 3.2: Artemis stack in base plate next to CubeSat chassis

3 nibbles ([11:8], [7:4], and [3:0]) designate the number of the tile to be used in the triad. If a tile that doesn't exist (e.g. tile 10 in a 9-tile system) is designated, that input into the voter is 0x0000.

The Virtex-6's I/O is primarily devoted to three 1 GB DDR2 RAM chips. Pinout restrictions for DDR memory can be found in [45]. Pinouts for each of the RAMs are included in Appendix A. The termination resistance of the I/O banks used for the DDR2 is set via external resistors connected to specific pins and set using pin constraints. Multiple banks on the same column can be set using Digital Cascade Impedance.

The Virtex-6 has 24 2.5 V GPIO via the 100 pin Sensor connector. These are intended for a expansion board, such as a camera or hyperspectral sensor. There also 4 unused 2.5 V I/O lines on the Misc I/O connector connected to the Virtex-6.

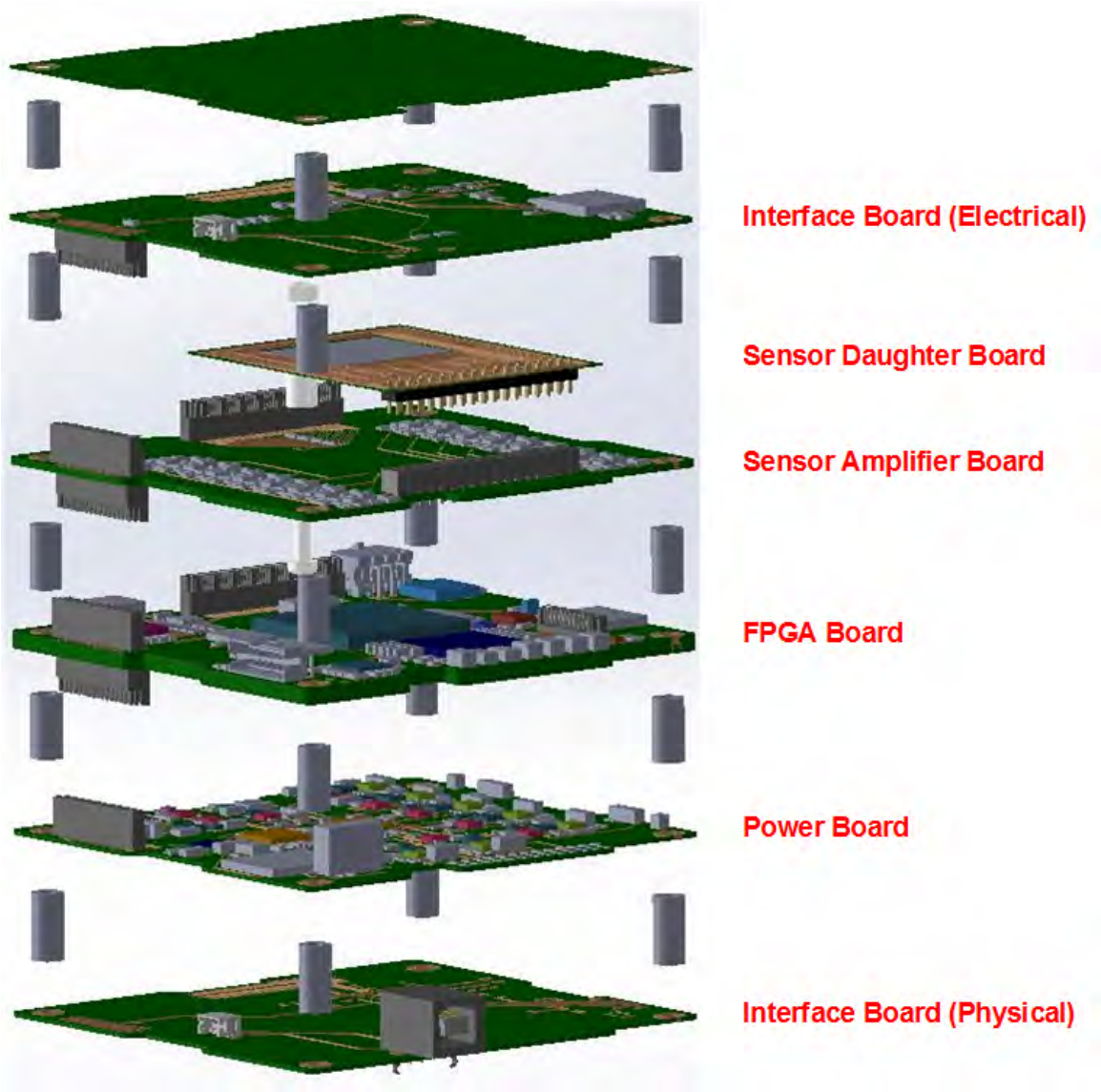

Figure 3.3: Exploded view of an Artemis CAD model

Configuration Select: The primary way to configure the Virtex-6 is through the Spartan-6 partial reconfiguration system. The SelectMAP port in a Slave configuration is used to accomplish this, with the Spartan-6 as master. The Virtex-6 uses a 3-bit bus to select the configuration mode, attached to switches 3, 4, and 5 on SW1. For most operations, the configuration mode should be 110. In case of

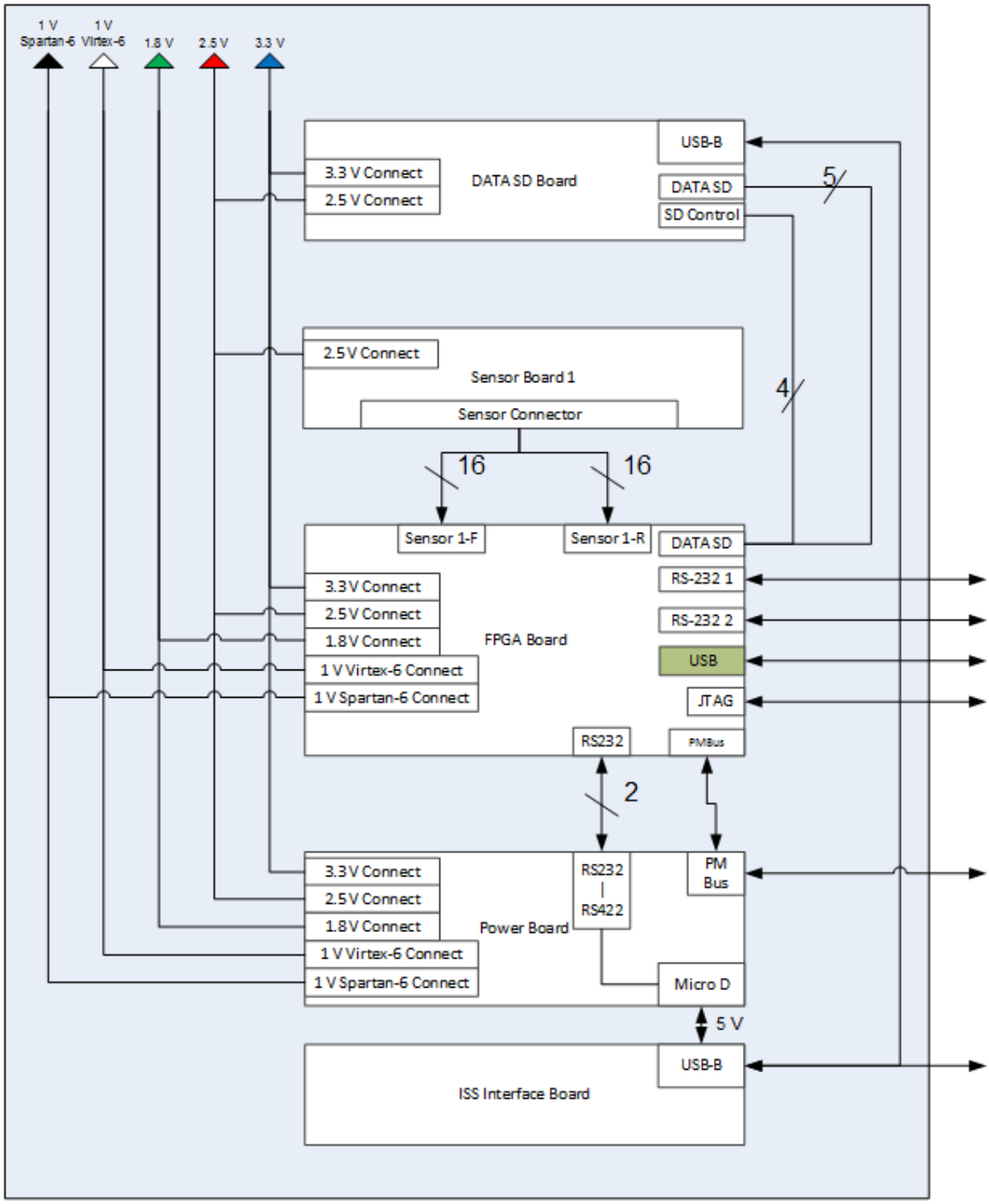

Figure 3.4: Block diagram showing the major I/O and power rails in the Artemis stack

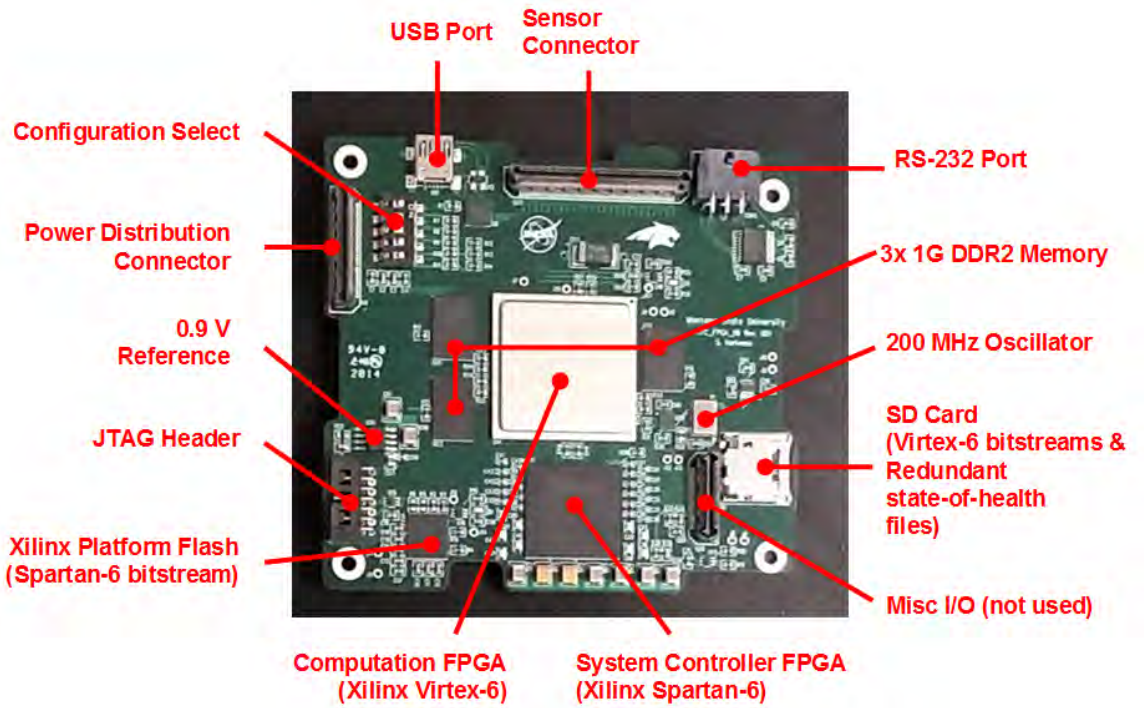

Figure 3.5: Annotated picture of a fully assembled FPGA board

benchtop testing, JTAG configuration can be selected with 101. Other configuration modes can be selected based on Table 3.1.

Table 3.1: Virtex-6 FPGA Configuration Modes [4]

Configuration Mode	**M[2:0]**	**Bus Width**	**CCLK Direction**
Master Serial	000	1	Output
Master SPI	001	1	Output
Master BPI-Up	010	8,16	Output
Master BPI-Down	011	8,16	Output
Master SelectMAP	100	8,16	Output
JTAG	101	1	Input (TCK)
Slave SelectMAP	110	8,16,32	Input
Slave Serial	111	1	Input

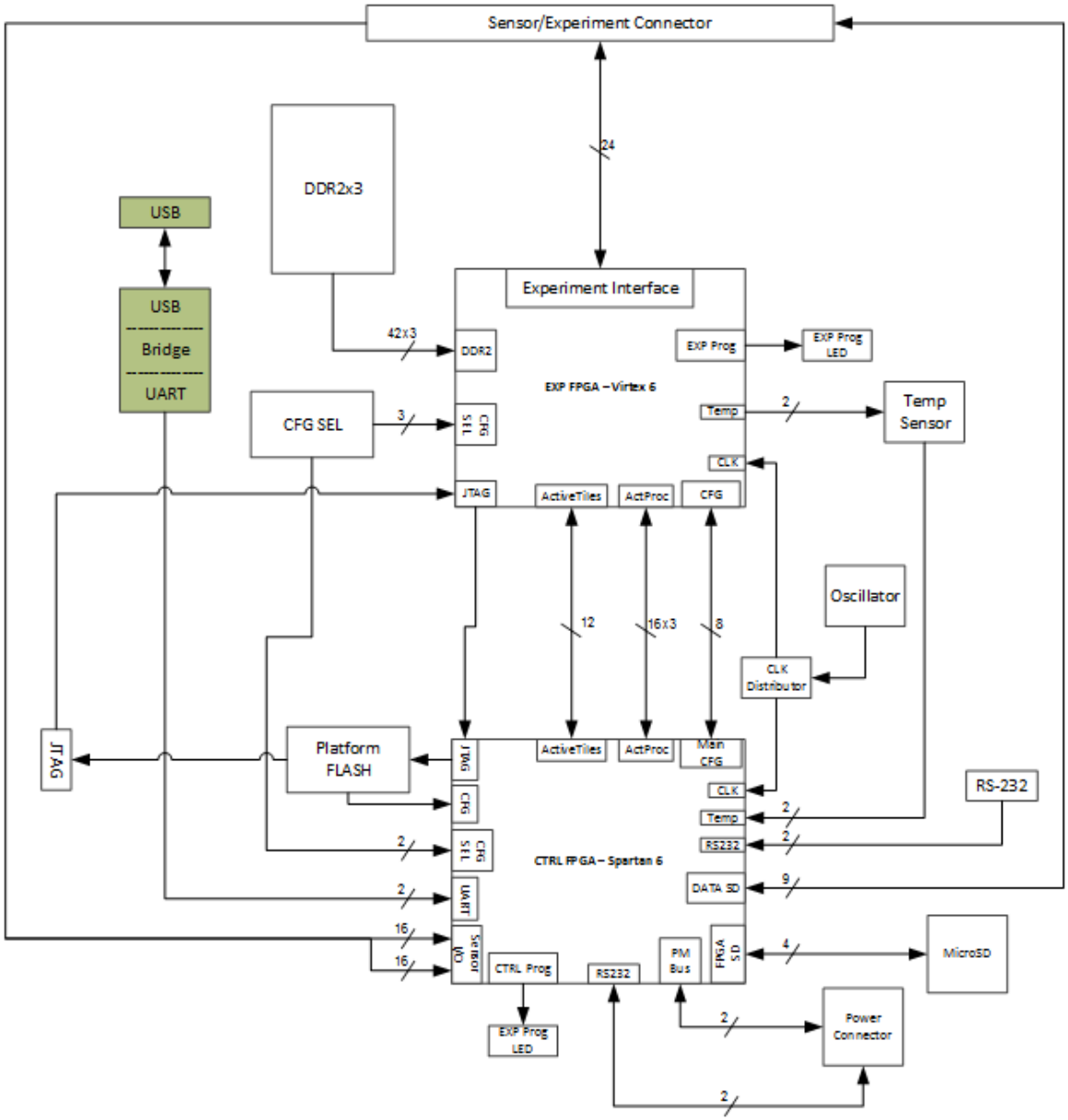

Figure 3.6: Block diagram of the FPGA board I/O

Spartan-6 and ControlOS

The Spartan-6 has I/O banks for 2.5 and 3.3 V I/O. The radiation sensor and DATA SD Card are connected to banks using 2.5 V logic, while the FPGA SD Card and misc I/O use 3.3 V logic. The radiation sensor is 2 16-bit vectors located on the

sensor connector, while the DATA SD card requires 9 bits on the connector (5 for the SD Card itself, and 4 for command and control). There are an additional 23 2.5 V I/O lines on the Sensor Connector connected to the Spartan-6.

The 200 MHz differential clock is brought into a Xilinx CoreGen Digital Clock Manager, which breaks out into 200, 100, 20, and 8 MHz single-ended clocks. The 200 MHz clock is used by the Microblaze. The 100 MHz clock is used by the radiation sensor accumulators. The 20 MHz clock is used by the FPGA SD card controller and Virtex-6 configuration block. Finally, the 8 MHz clock is used by the Soft-Error-Mitigation (SEM) controller.

MicroBlaze is a 32-bit Harvard architecture soft processor core. Softcore processors are unique in their ability to be expanded with custom instructions and hardware which allows for the strength of the FPGA to be exploited while maintaining simplicity of design to the user. MicroBlaze communicates to the VHDL half of a FPGA design through the use of peripherals, typically across an AXI bus. A peripheral is a 1-2 channel, 1-32 bit port with variable data direction. Virtual pins can be overloaded to function as both an input and an output, though this may introduce problems when adapting code from a traditional microcontroller. If the number of connections to the AXI bus (peripherals + Microblaze) is 16 or less, an AXI4lite interconnect can be used, with a drastically reduced footprint from a normal AXI interconnect. Xilinx EDK automatically creates the required addresses and define macros. These can then be used to initialize the peripheral in C.

Best practices for the use of peripherals in Microblaze include a separate header and code file for each peripheral. Individual functions for reading each input and writing each output are recommended, but will inevitably introduce extra clock cycles. Macros should be used to define SET and CLEAR masks for each output, and READ masks for each input. In addition, if virtual pins are overloaded with an input and

output, DDR masks should also be created. Two global variables the size of the virtual register (1-32 bit) should also be defined for the current state of outputs and the data direction.

Configuration Select: There are 2 methods for Spartan-6 configuration. The default method is a bitstream pre-loaded on the Xilinx Platform Flash. The bitstream contains all the configuration data for the Spartan-6, including user code intended for the Microblaze soft processor. The Spartan-6 uses a 2-bit bus to select the configuration mode, attached to switches 1 and 2 on SW1. For most operations, the configuration mode should be 01 or 11, as shown in Table 3.2. Special care must be taken when loading the PROM file to coordinate the Master and Slave designation with SW1 values. This bitstream can also be superseded by a user-configuration over the JTAG port for debugging and development purposes without changing SW1.

Table 3.2: Spartan-6 FPGA Configuration Modes [5]

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

Scheduler: ControlOS is implemented as a first-in-first-out (FIFO) cooperative scheduling kernel with 16 task slots. Tasks can be scheduled on 1s intervals, and are user-defined. This scheduling algorithm was chosen as it was simple to implement with low overhead. A task is defined by 3 parameters: a pointer to a function, the period (in seconds), and if the task is enabled or disabled. In this way, tasks can be modified during run-time.

PMBus: Power Management Bus is an open standard power-management protocol. It is a relatively slow speed two wire communications protocol based on I^2C. Unlike I^2C, PMBus defines a substantial number of domain-specific commands rather than just how to communicate. Typically, different aspects of a device (configuration, monitoring) will be subdivided into pages. With the correct logic, it is possible to properly control all aspects of the power controllers, such as re-sequencing in case a fault is detected on a voltage rail. For ControlOS, we are most interested in the information contained in the pages pertaining to the voltage rails. Note that the pages are 0-indexed, while the rail numbers are 1-indexed.

Voltage information is contained in a Linear-16 format. A floating point number can be retrieved out of Linear-16 using the equation $V = A \cdot 2^X$. A is the 16-bit unsigned mantissa obtained using the READ_VOUT command. X is the signed 5-bit 2s-complement exponent obtained using VOUT_MODE. The power controller automatically sets the exponent based on the expected range of the voltage (Table 3.3). [6]

Table 3.3: Exponent Influence on Voltage Range and Resolution [6]

Exponent	Range [V]		Resolution [mV]
-9	0	63.99805	1.9533
-10	0	31.99902	0.97656
-11	0	15.99951	0.48828
-12	0	7.99976	0.24414
-13	0	3.99988	0.12207
-14	0	1.99994	0.06104
-15	0	0.99997	0.03052

Current information is contained in a Linear-11 format. A floating point number can be retrieved out of Linear-11 using the equation $I = A \cdot 2^X$. A is the unsigned mantissa contained in the lower 11-bits of the 2-byte value returned by

the READ_IOUT command. X is the signed 2s-complement exponent contained in the upper 5-bits of the same 2-byte value.

The temperature of the power controllers is also contained in a Linear-11 format. The process for retrieving the floating point number is the same as for current, only using the READ_TEMPERATURE_2 command instead of the READ_IOUT command.

The maximum values for current and voltage are kept locally on the power controllers, and can be retrieved using the LOGGED_PAGE_PEAKS command. Minimum values are a simple comparison of the current value against the old minimum value. Averages are computed using the cumulative moving average in equation 3.1. The moving average is cleared after every write to the Data SD Card to avoid the chance of a integer overflow leading to a divide by zero condition.

$$y_n(x) = y_{n-1} + \frac{x_n - y_{n-1}}{n} \tag{3.1}$$

V6 Temperature: A MAX6627 is used to measure the die temperature of the Virtex-6 by polling an on-chip, diode-connected transistor. Data is provided as a signed 2s-complement value contained in the upper 13-bits of a 2-byte value across a 3-pin SPI . The resolution of the MAX6627 is .0625°C, so the 13 bit integer is multiplied by .0625 to form a floating point representation. [46]

Fault Injection: To simulate a fault on the Virtex-6, there are two possible methods. The more realistic of the methods is to pause one of the active TRM triad and allow the voter to mark the tile as faulted. A spare is brought online in the faulted tiles place, and the faulted tile marked for partial reconfiguration. This method will

only inject faults into the active triad. To test the spare tiles, the global variable denoting which tiles are faulted is modified to flag a partial reconfiguration.

Blind Scrubbing: Blind scrubbing can be performed on a per-tile basis, or for the entire Virtex-6 fabric, to scrub the areas not captured in a tile, such as the voter and interconnect to the voter. Blind scrubbing is actually performed the same as initial configuration, minus the peripheral initialization to connect to the SelectMAP port and FPGA SD card.

Readback Scrubbing: Readback scrubbing is performed by configuring the Virtex-6 to output readback data and comparing it to the golden copy located on the FPGA SD card. The golden copy is the same configuration data used by to perform partial reconfiguration interleaved with a bit mask. The mask specifies which bits are important, such as configuration memory contents, and which are not important, such as user memory. Currently, readback increments a counter, though it could be used in later software revisions to scrub only affected areas of configuration memory.

FPGA SD Card

The FPGA SD Card was designed to use a 4-pin SPI bus (MISO, MOSI, SCLK, CSn). Currently, this revision of the board does not have traces to utilize SD mode (DAT[0:3], CMD, SCLK, etc). This design was chosen to keep additional FPGA pins open for future expansion, as well as to use the much more simple SPI interface. SPI mode has a speed limit of 25 MHz, which is more than adequate for our purposes.

As an embedded processor, the MicroBlaze introduces severe delays when dealing with I/O. Therefore, a hardware SPI state-machine is implemented on the Spartan-6 to handle the full and partial reconfiguration operations. The state-machine

is controlled by ControlOS. ControlOS passes the state-machine a start address and length, and the state-machine proceeds to program the Virtex-6 through the SelectMAP port. [47] Table 3.4 shows the memory map of the FPGA SD card.

Table 3.4: Memory Map of the FPGA SD Card (2 GB)

Address	Contents
0x00000000 — 0x00000199	Reserved
0x00000200 — 0x00536380	Full Reconfiguration Bitstream
0x00536800 — 0x0089EE5C	Partial Reconfiguration Bitstreams
0x10000000 — 0x1009C66C	Interleaved Full Bitstream & Readback Mask
0x20000000 — 0x20000199	Last Used Sector for RAW packets
0x20000200 — 0x775FFE00	RAW packets

The first portion of the FPGA SD card contains a full bitstream, used during initial configuration to program the entire FPGA. Following that are the bitstreams for each of the individual tiles. These partial bitstreams are used during partial reconfiguration operations to recover a faulted tile. Third come the interweaved readback bitstreams. Each byte of the bitstream is interleaved with a bit mask indicating if a configuration bit needs to be compared to what is in the Virtex-6 configuration SRAM. Some resources on the FPGA can be safely ignored, such as the RAM containing the heap and stack for an embedded microprocessor. Starting addresses and lengths of all the bitstreams must be hardcoded into the C software of ControlOS at this time. Further

improvements of the software would allow this information to be stored in the first sector of the FPGA SD card.

The remainder of the FPGA SD Card contains the raw TLM packets (MODE, TILE, HEALTH, SENSOR). Unlike the DATA SD Card, the packets on this card don't contain the packet index or the descriptive title of each field in the packet. Each byte is written to the FPGA SD Card as an unsigned 8-bit integer (u8) for the purposes of creating a CRC.

Appendix B contains the structure for all packets used by Artemis. The first sector of the packet section of the SD card contains the last written to address. This prevents ControlOS from overwriting data in the case of power-cycling.

Radiation Sensor & Amplifier Board

The physical radiation sensor is mounted on a breakout board, where silver epoxy and wire-bonds route the channels onto a 100 mil header. From there, the signals are conditioned by an amplifier chain detailed in Figure 3.8. The bias voltage is 15V for the frontside channels and GND for the backside channels. The in-line capacitor is used to reject any DC. Stage one of the amplifier chain is an integrating amplifier tuned to stretch out the radiation strike current pulse into a signal which the Spartan-6 FPGA can sample. The 2nd stage rectifies the signal. The 3rd stage converts the signal into a LVCMOS25 digital signal, perfect for sampling.

Due to size restrictions of the PCB area, the number of amplifier chains had to be reduced to 8 front and 8 rear channels. Adjacent pairs of channels are tied together, effectively reducing the sensor from 16x16 pixels to 8x8 pixels. Zero Ω resistor blocks are used to determine whether the sensor is Sensor 1 or 2. This feature was included

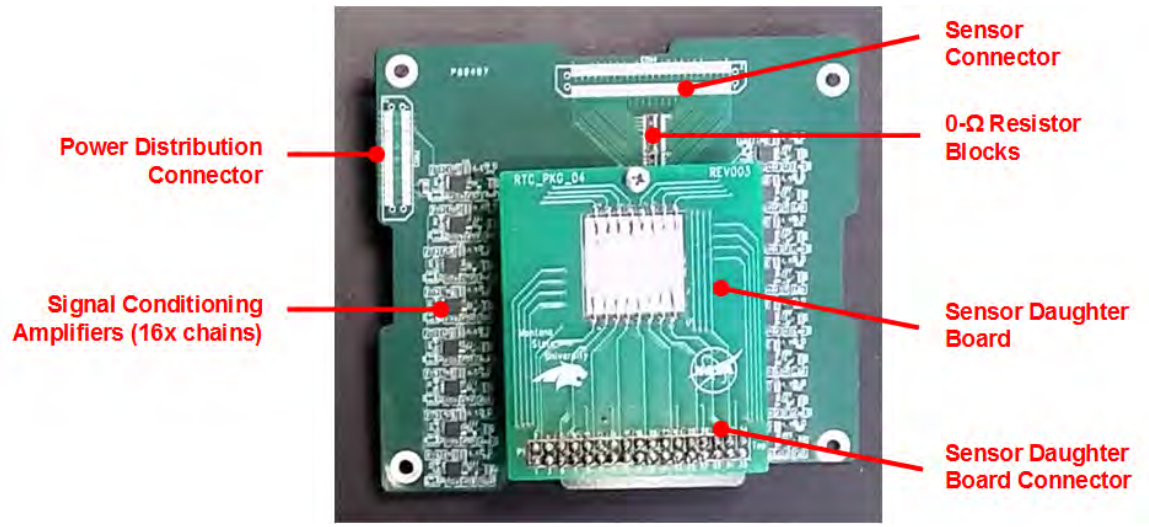

Figure 3.7: Annotated picture of a fully assembled Sensor board

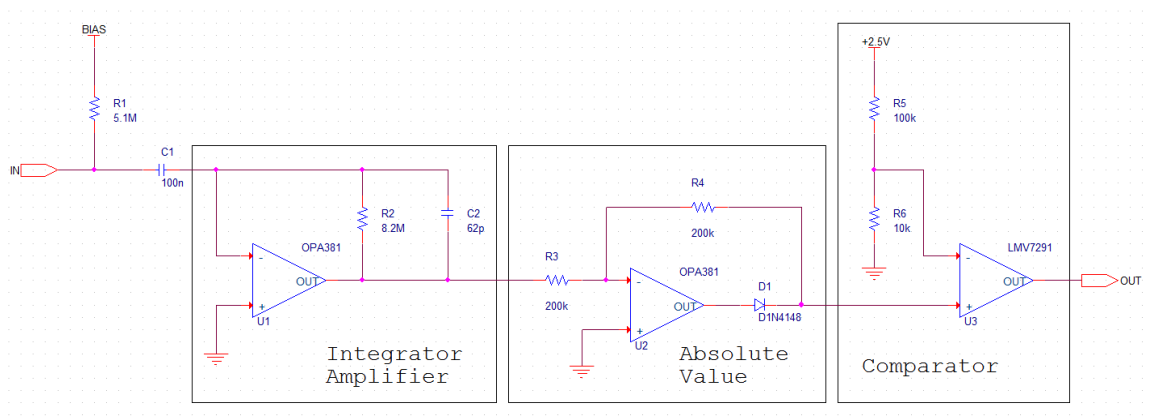

Figure 3.8: Pulse-Stretching Analog-to-Digital Amplifier Circuit

to reduce design costs on Sensor 2 specific PCB. Instead, the resistor blocks can be switched to their alternate outputs.

Power Board

The Artemis power regulation board was greatly based upon previous work. [9] To save space, the high-voltage input capability has been removed, restricting the

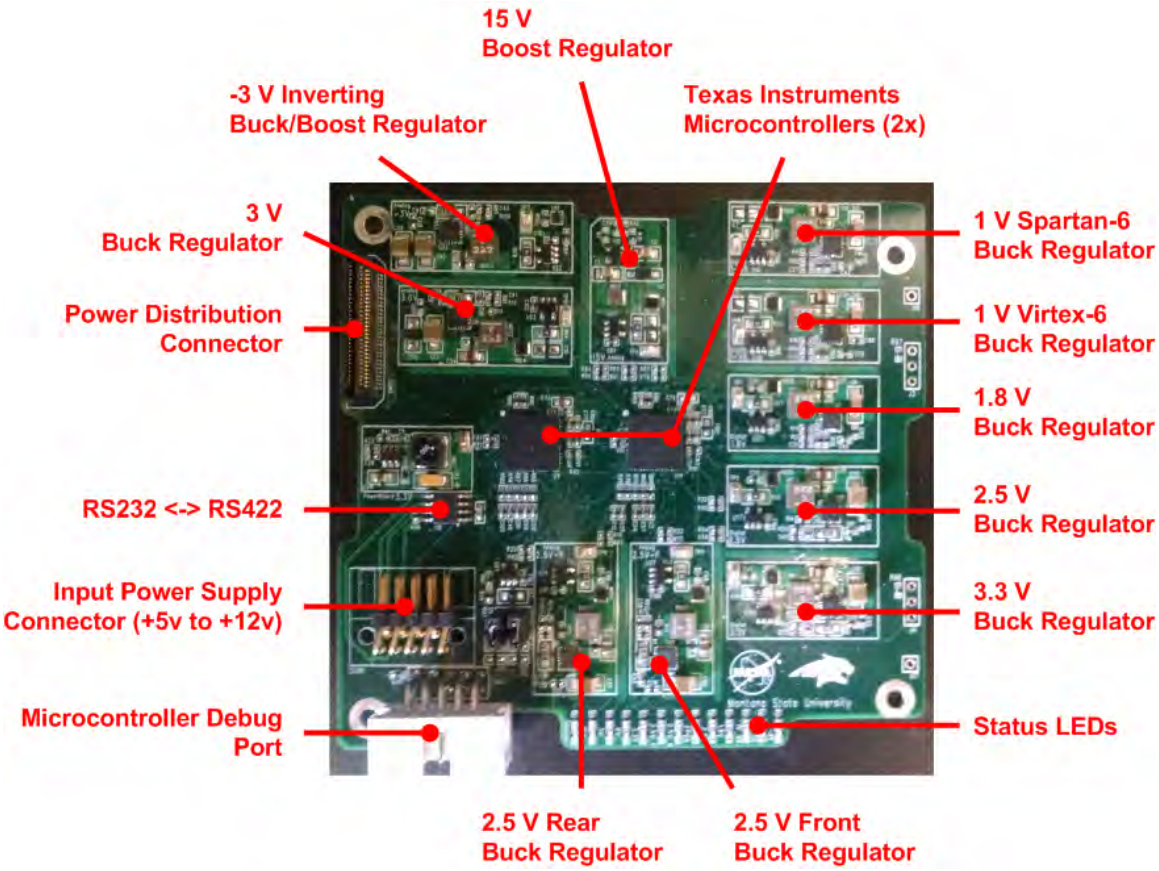

Figure 3.9: Annotated picture of a fully assembled power conditioning board

input voltage of Artemis to 4-13 V DC. The power board has 11 voltage rails: 5 digital, 5 analog, and a 3.3 V to supply the power controllers and regulators. With the exception of the 15 and -3 V analog rails, all the supplies are buck regulators. For a more detailed look at power supply design, refer to [9]. The power board includes internal monitoring, fault protection, and closed loop margining. Fault protection eliminates the danger of shorting the board during bench testing, while closed loop margining has increased voltage stability vs temperature or load and allows for on-board adjustment of the regulators.

The power supplies (notably the 3 2.5 V regulators) are separated because of the desire to analyze the power draw of different aspects of Artemis. The comparator

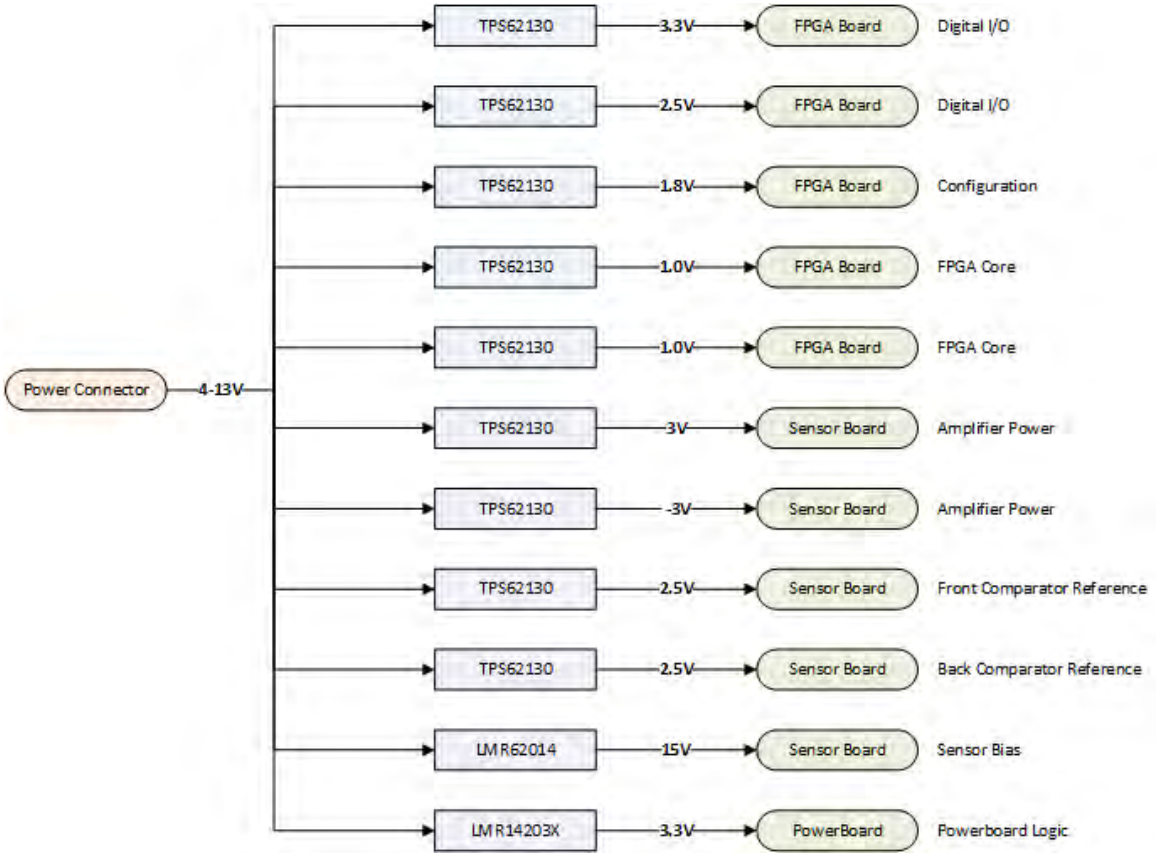

Figure 3.10: Simple block diagram of power conditioning board

voltages ($2.5V_F$ and $2.5V_R$) are separated from each other and the digital I/O 2.5V rail to examine the operation of the sensor. If $2.5V_F$ is drawing more current than $2.5V_R$, we can conclude we are being bombarded by particles without the energy to fully penetrate the sensor, for instance.

On the power connector, the power supplies are separated by ground lines to provide a proper return path.

Regulator Circuit Designs

The use of power controllers necessitates the use of a separate power supply which is not enabled by said controllers. The LMR14203 was selected for the simplicity of implementation and reduced power requirements.

To generate the bias voltage used by the sensor, a TI LMR62014 boost regulator is used. This regulator is initially tuned to 15 V, and can only supply small currents in the case of a runaway condition on the sensor.

To generate a -3 V rail, the TPS62130 is used in an inverting Buck-Boost topology [48]. In this architecture, the output of the regulator circuit is defined as ground and the regulators ground reference is defined as the output. Referencing to the output has the effect of reducing the output current available, due to the output inductor being attached to global, reducing the rated current of the supply from 3 A to 1.3 A. In addition, the margining and enable inputs are referenced to the negative voltage output and must be shifted to remain functional. The margining was deemed unnecessary, and the enable logic shifted using a simple MOSFET circuit.

All other regulators use the TPS62130 switching regulator. The TPS62130 was chosen due to its high rated efficiency, small size, and high switching frequency, which allows for the use of small inductors. The only difference between each circuit is the selection of resistor values for feedback and margining. This greatly reduces layout time, and inherently separates noisy regions of the circuit. An optional LC ripple filter was placed at the end of the circuit, which can be bypassed by using a 0 Ω resistor in place of the inductor. Finally, the current sense resistor and op-amp used for current monitoring end the chain.

Regulator Layout Designs

Switching regulators are highly susceptible to EMI, and generate a large amount of EMI. To combat this, traces with high dV/dT should be made as short as possible and kept away from feedback traces. To reduce noise coupling between regulators, three layout techniques are used. First, a ground plane is placed on layer 2 to separate deeper layers from switching noise generated on the top layer. Second, ground rings surround each regulator. Finally, each regulator has an individual trace to the 3.3 V rail supply, isolating any feedback on regulator inputs.

The switching supply for the power controllers is a very simple LMR14203 regulator circuit. The feedback trace was placed further along the output to reduce the ripple that may be present directly at the inductor.

The TPS62130 regulators were all designed with a standard layout. [49] This largely followed the reference design by keeping the FB and VOS nodes isolated from the SW node by separation across a ground plane or copper pour. Some care was also taken to ensure the margining voltage was placed away from the switch node due to its influence on the output voltage.

Power Controller Settings

Power monitoring, rail sequencing, fault detecting, and margining are managed by two TI UCD90124A devices. Each contains twelve, 12-bit ADC lines for voltage, current, and temperature monitoring. They also have GPIO for supply enables and PWM outputs to facilitate closed loop margining. The power controllers can communicate using the PMBUS protocol over I2C, with the Spartan-6 FPGA or a TI USB-GPIO device operating as the master device. Initial configuration of the power controllers is accomplished using TI Fusion GUI interface software. The I2C

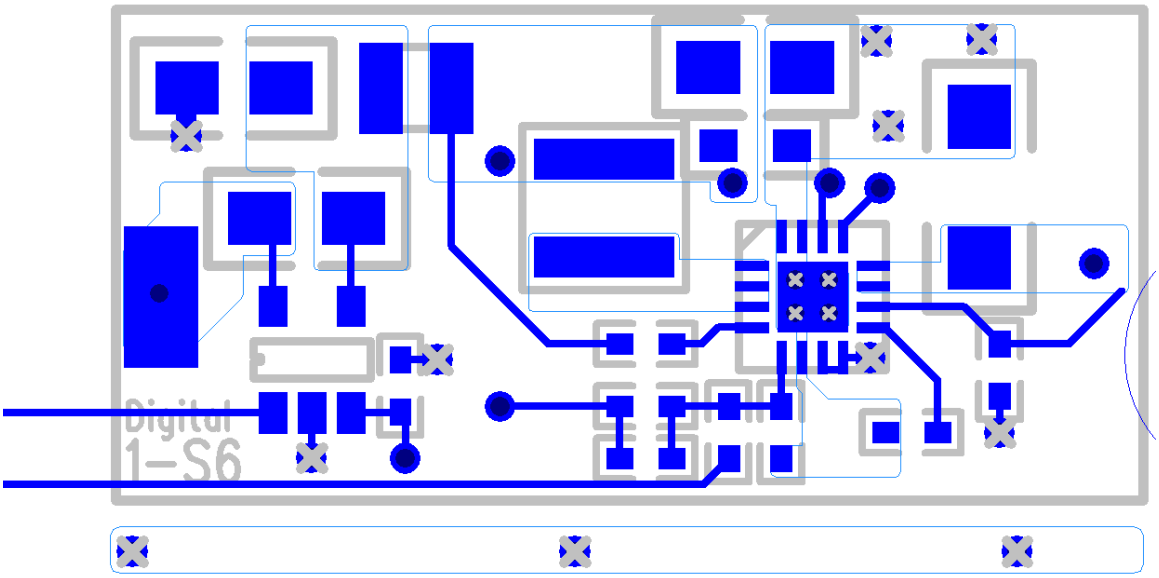

Figure 3.11: Standard Layout of a TPS62130 Buck Regulator

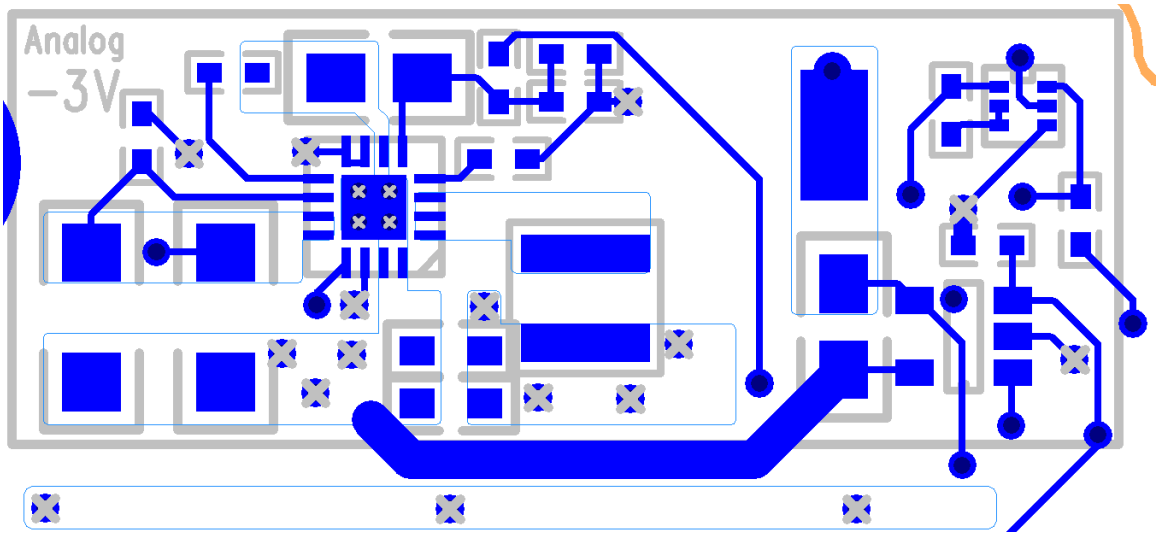

Figure 3.12: Standard Layout of a TPS62130 Inverting Buck/Boost Regulator

addresses are configured with resistors and are set such that the digital voltages and analog voltages are separated onto I2C address 0x52 and 0x53 respectively.

Each supply on the power board has a Red LED associated with it, which will light up in the event the rail is faulted. There are an additional Green LED for each

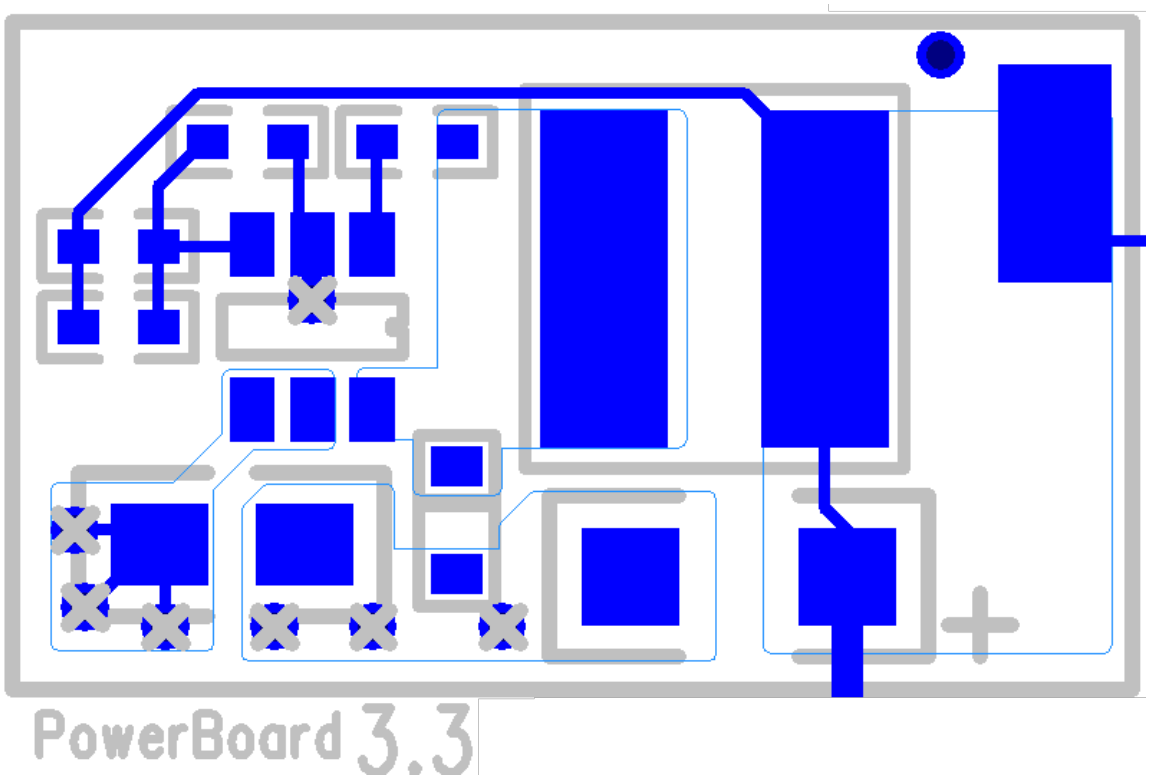

Figure 3.13: Standard Layout of a LMR14203 Buck Regulator

power controller to signify that every rail for each controller is operating normally. These LEDs are hooked to General Purpose Output pins on the power controllers and user configured.

Voltage monitoring is performed by ADCs on the power controllers; however the ADCs are limited to a maximum voltage of 2.5 V. For any voltage that may exceed this, an internal voltage divider is used to restrict the maximum range. Current monitoring is performed using an inline current sense resistor with a differential op-amp measuring the voltage across it. All the op-amps are calibrated for a 20x gain, with a 50 mΩ resistor this results in a 1000 mΩ resistor for all current measurements.

Voltage rails are sequenced primarily to reduce the transient currents pulled when a regulator is activated, thereby reducing the likelihood of faulting a current limit

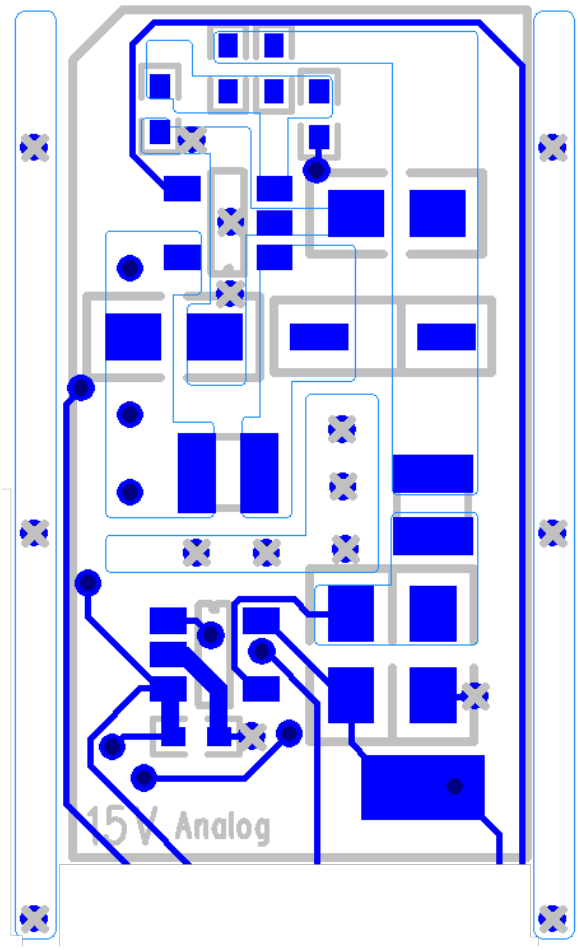

Figure 3.14: Standard Layout of a LMR62014 Boost Regulator

on the input supply. In addition, however, the recommended power-on sequence for Virtex-6 device is V_{CCINT}, V_{CCAUX}, and V_{CCO}, i.e. 1 V, 2.5 V, and 1.8 V. [50] The Spartan-6 has no power-on sequence requirements. [51]

ISS Interface Board

The Interface board connects to the stack via multiple interfaces. At the bottom of the stack, the Interface board breaks the USB 5V power and Ground out to the Power board using wires. The 4 wires of the USB port (VCC, GND, D+, D-) are

Table 3.5: Power Controller Pin Parameters

(a) Digital Controller

#	Rail	Voltage	Temp	Current	Enable	Margin	LED
1	Vin	56	62	5			11
2	3V3	59		63	33	17	12
3	2V5	6		1	34	18	13
4	1V8	54		2	35	19	14
5	1V0 S6	50		4	37	21	29
6	1V0 V6	52		3	36	20	25

(b) Analog Controller

#	Rail	Voltage	Temp	Current	Enable	Margin	LED
1	Analog		63				11
2	15V	6		59	33	17	12
3	2V5 F	5		1	35	19	13
4	2V5 R	4		62	34	18	14
5	3V0	50		2	36	20	29
6	-3V0	52		3	37	21	25

Table 3.6: Power Controller Voltage and Current Ratios

(a) Digital Controller

#	Rail	Voltage Ratio [V/V]	Current Ratio [mΩ]
1	Vin	0.0625	1000
2	3V3	0.5714	1000
3	2V5	0.6667	1000
4	1V8	1.0	1000
5	1V0 S6	1.0	1000
6	1V0 V6	1.0	1000

(b) Analog Controller

#	Rail	Voltage Ratio [V/V]	Current Ratio [mΩ]
1	Analog	N/A	N/A
2	15V	0.1333	1000
3	2V5 F	0.6667	1000
4	2V5 R	0.6667	1000
5	3V0	0.5714	1000
6	-3V0	0.5	1000

sent through the Hirose connector to the DATA SD board. On the SD DATA board, the MAX14502 USB-to-SD Card Reader with Bypass is utilized to make the DATA SD card appear as a removable drive to a PC.

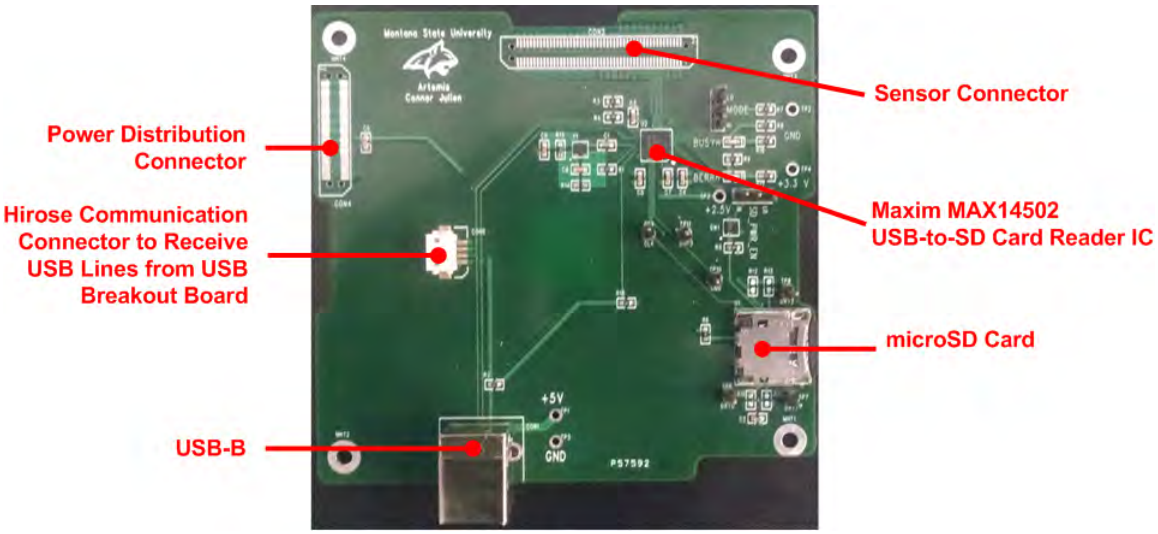

Figure 3.15: Annoted picture of a fully assembled Interface board

The MAX14502 has logic to allow multiplexing of the DATA SD card master. During the majority of operations, the USB is master (aka Card Reader mode), allowing the DATA SD card to appear as a removable drive. When a packet is written, ControlOS becomes the master (aka Pass-Thru mode) and writes text files. ControlOS uses a 4-pin SPI bus to communicate with the DATA SD card as well as a card detect (CD) signal, while USB is translated into SD commands to communicate with the DATA SD card. A load switch operated by ControlOS is interrupted whenever switching between Card Reader and Pass-Thru modes to power cycle the DATA SD card. While it is possible to enter SPI mode while in SD mode (but not vice-versa), it is possible the MAX14502 will be attempting to communicate to the DATA SD card when the MAX14502 is commanded to enter pass-thru mode. Any

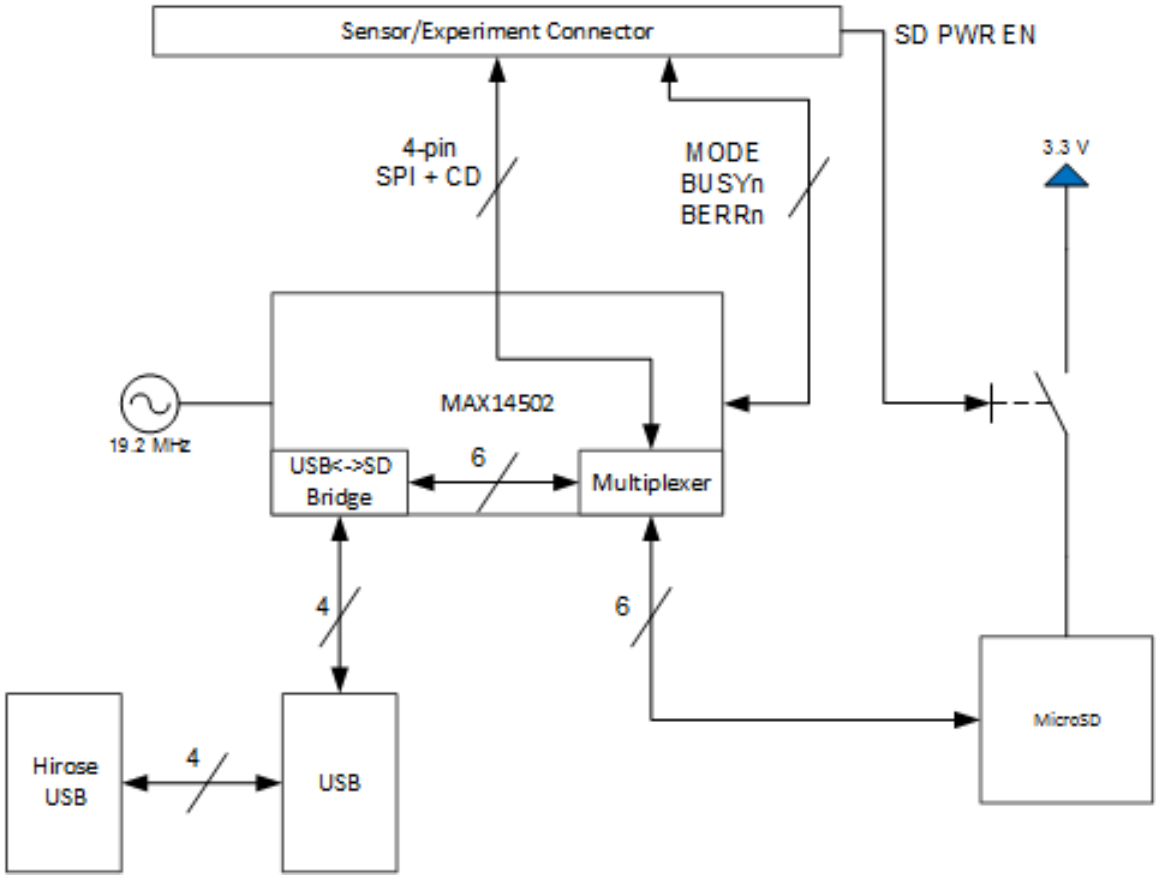

Figure 3.16: Block diagram of the Interface board I/O

attempts to enter SPI mode in the middle of an unknown SD command will likely fail.

Figure 3.17 shows a basic block diagram of initializing the SD card in SPI mode. The typical SPI command is 1 byte command, 4 bytes argument, and 1 byte CRC. The typical response is an R1 response of 1 byte. An R1 byte can take up to 9 full bytes of dummy clock cycles to respond. It is good practice to toggle the Chip-Select line between commands to clear any lingering commands. The Interface board has all SD pins broken out on standard pins for logic analyzer probes.

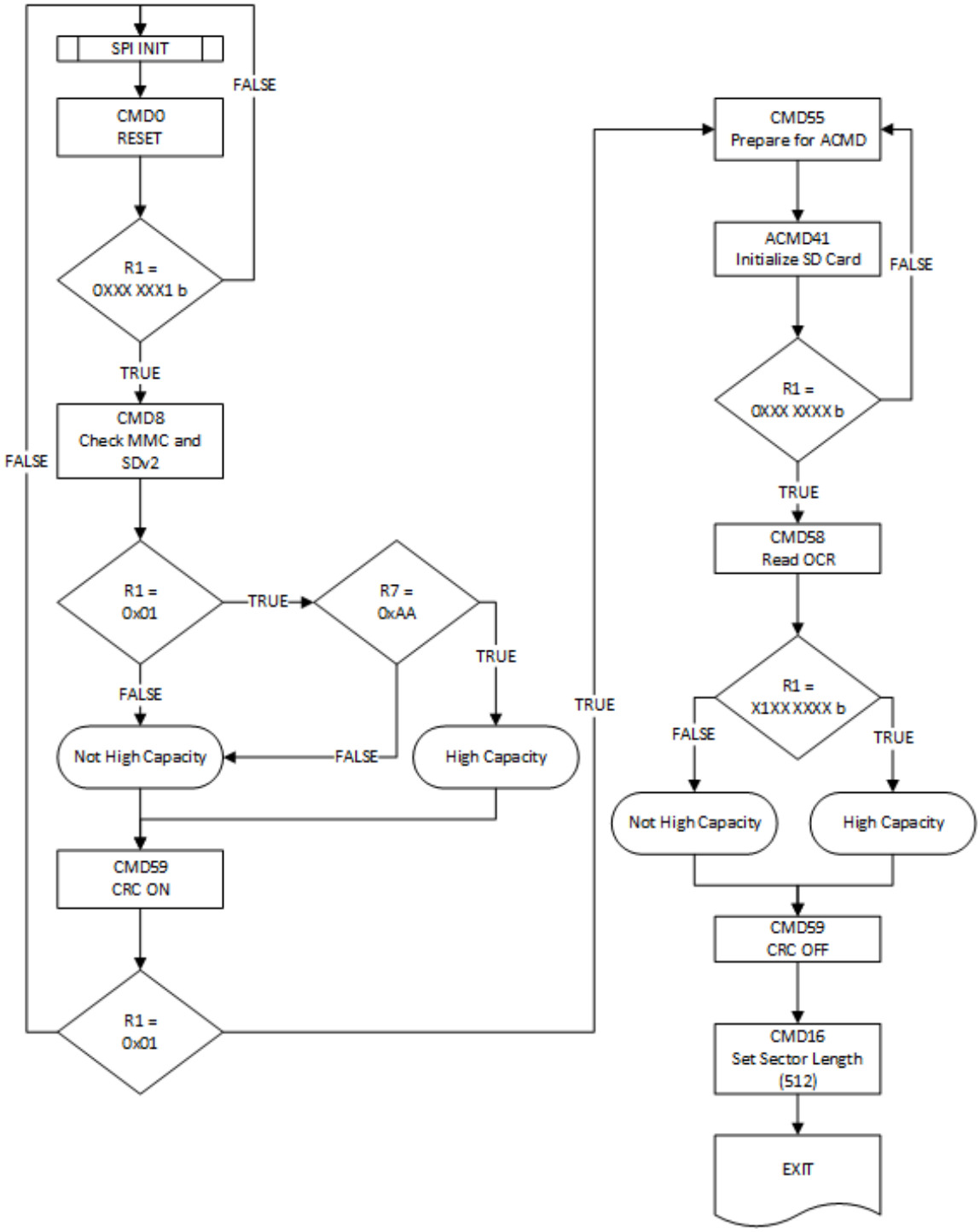

Figure 3.17: Block diagram of initializing a SD card in SPI mode

After successfully initializing the DATA SD card, the Master Boot Record and FAT Boot Record are examined to prepare the software to write files. Figure 3.18 shows a basic block diagram of reading the two boot records. After the FAT sector and DATA sector have been determined, all further file functions (opening, closing, reading, writing, renaming, deleting) are handled by FAT file system specific ⟨stdio.h⟩ functions (ffs_fopen, ffs_fclose, ffs_fgetc, ffs_fwrite, ffs_rename, ffs_remove).

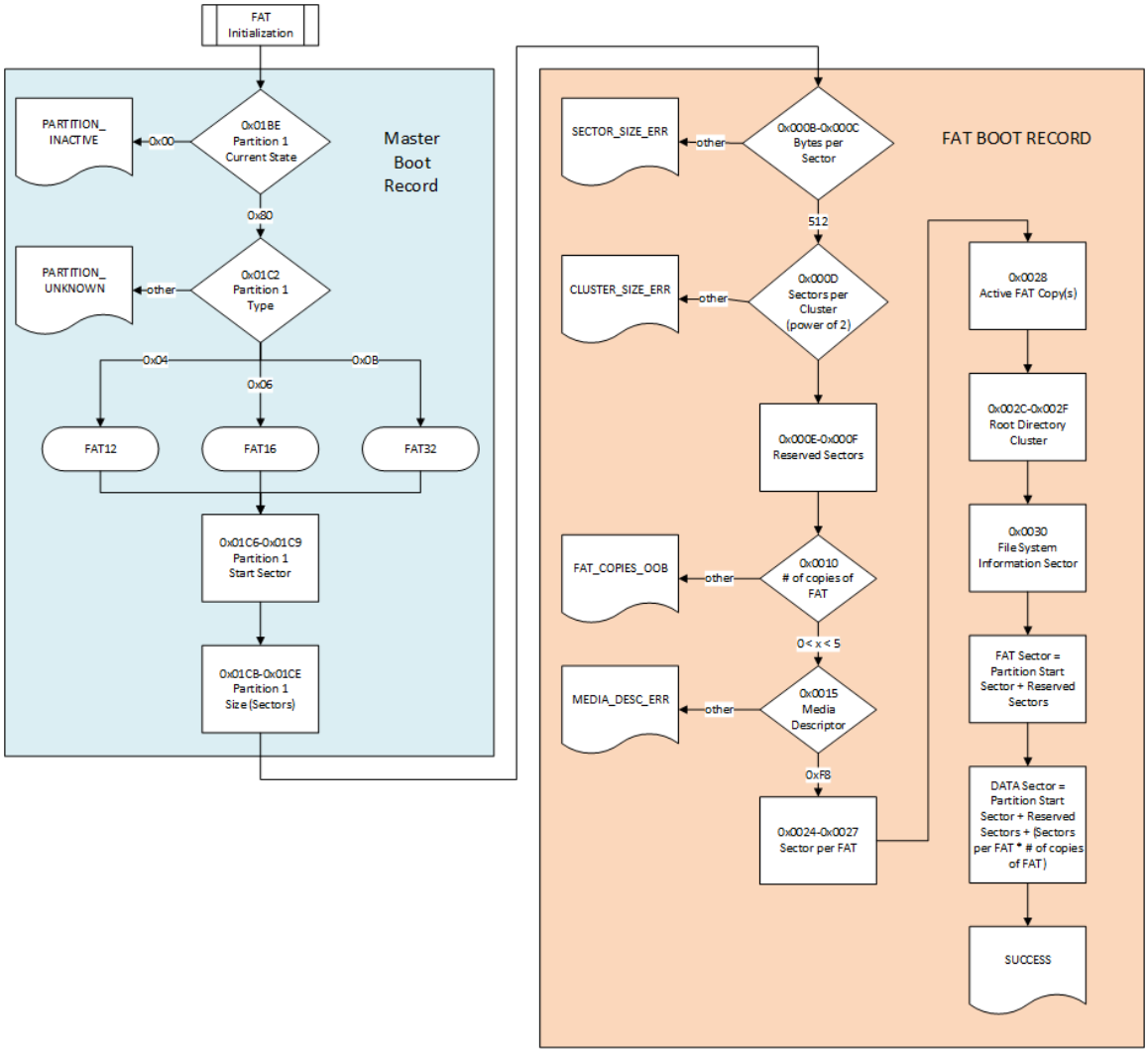

Figure 3.18: Block diagram of initializing a FAT32 formatted SD Card

Figure 3.19 shows a basic block diagram of the procedure to take control of the DATA SD card and write a file. The file is divided up into 4 different packets (see Appendix B for details), which are each written separately. This enables more coherent flow in the code, and has the added benefit of forcing a DATA SD card reinitialization.

One of the limitations of the file system are the MSDOS filenames. Filenames are restricted to 7 character filenames, a period, and a 3 character extension (e.g. txt, bin, raw). When writing a file, it is opened for appending. This is to avoid overwriting data in the event of a powercycle before the contents of the DATA SD card can be retrieved and removed. Unfortunately, a ffs_fprintf function does not exist, so a workaround was developed. A char[] array of sufficient size to hold any line we want to print is filled with a formatted string using snprintf, which also returns the length of the formatted string. the ffs_fwrite function is then passed the char[] array and the length of the string to write.

The rate at which files are written to the DATA SD card is configurable on the order of seconds. Each text file written is approximately 10 kB, so the length of time until a card of size S is filled can be approximated by $T(\Delta t, S) = \frac{S \cdot \Delta t}{10kB}$. It may be more appropriate to set the length of time as opposed to the writing interval, in which case $\Delta t(T, S) = \frac{T \cdot 10kB}{Size}$. For a six month time to fill a 2 GB SD card, the write interval should be > 78 seconds.

It is important that the SD card used be FAT32 formatted. A FAT16 card is limited to 512 files in the root directory. The root directory on FAT32, however, is an ordinary cluster chain. This difference allows the root directory to grow dynamically like a normal folder. FAT32 drives also includes redundancy in the boot record, making the less susceptible to single points of failure.

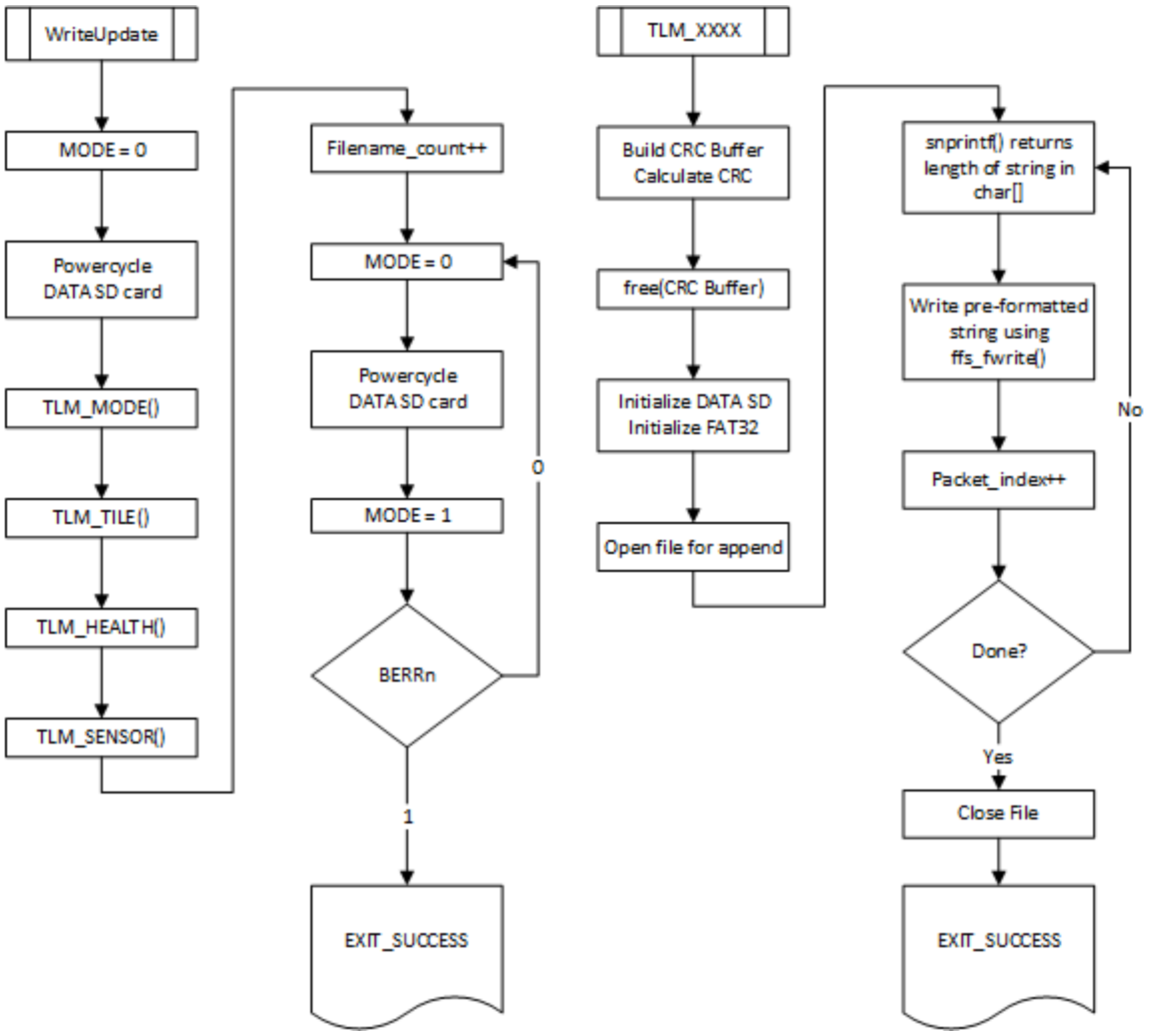

Figure 3.19: Block diagram of writing a data file

TEST RESULTS

Sounding Rocket

An Artemis prototype was flown on the Space Loft 9 (SL9) mission with UP Aerospace, directed by the NASA Flight Opportunities Program. This sounding rocket reached a maximum altitude of 100 km and free-fell to 15 thousand feet, whereupon a parachute would arrest the descent. The payload was expected to be powered multiple hours before flight, so 2 battery boards at 9 V were assembled, with an expected minimum of 8 hours of power.

A concern before flight was the time budget of Readback Scrubbing. The software implementation of Readback Scrubbing takes $\approx$ 20 minutes, near the duration of the entire flight. To avoid a situation where the only ControlOS operation during flight was Readback, a mechanical G-Switch was attached to the frame of the stack, connected to a VHDL latch on the Spartan-6. Upon detecting a G-Switch event, a 20 minute timer began counting down to begin Readback Scrubbing. Upon completion of Readback, the G-Switch latch was cleared.

Successes

After flight, the prototype Artemis stack was removed from the mounting plate and shipped back in a padded case. A cursory examination of the stack showed that all components had remained attached, and mechanical standoffs were still tight. Utilizing the same batteries as during flight, the stack was powered on, and both FPGAs successfully programmed. From these observations, it was concluded that Artemis successfully survived the physical rigors of a sounding rocket flight.

Figure 4.1: Artemis Prototype on SL9 Mounting Plate

We also correlated the data collected by the Artemis prototype against altitude and temperature data received from the onboard avionics. We indexed our data against the temperature data, as the Virtex-6 die temperature was the easiest data point to correlate. Figures 4.3 and 4.4 show our comparable data points. Collection of temperature data nd the Spartan-6 system counter prove that the Spartan-6 data collection functions were operating throughout the flight.

Figure 4.2: SL9 De-integration [16]

Failures

During before flight testing, a bug which 'cut off' the majority of every 7th packet was discovered. The packet was the correct length, but everything after a semi-random point early in the packet was untouched from the previous contents of the card (usually 0s). This was due to poor control of the data direction register responsible for the FPGA SD card and the scheduler. If the data-direction bit for the chip-select line on the SPI bus is set to input, the internal pull-up resistors of the FPGA SD card will pull CSn high, canceling all commands. If the function writing data out attempts to pick up where it left off, it will not attempt to drive CSn low again, leading it to write data to an unresponsive card.

After the flight, examination of the raw data saved to the FPGA SD card revealed a previously unexperienced bug. Instead of every 7th packet being incomplete, almost every packet was incomplete. This was further exacerbated by the sensor information

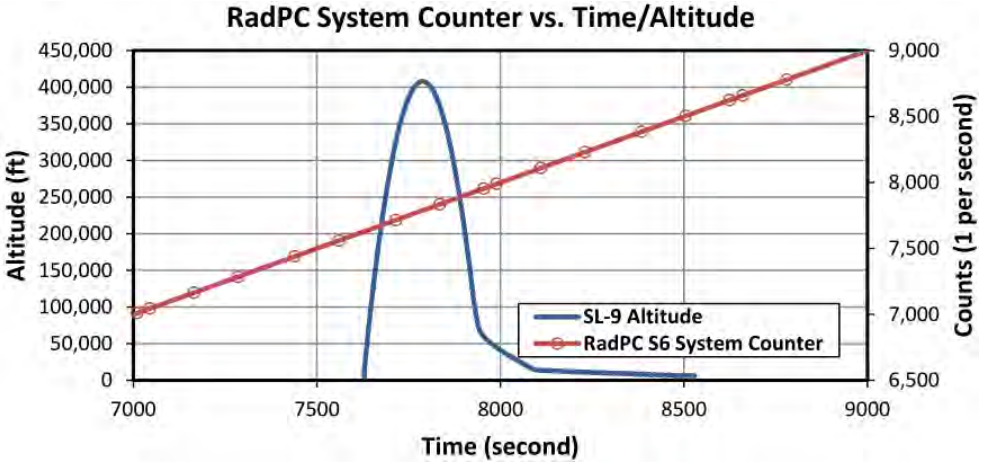

Figure 4.3: Spartan-6 1 second counter vs SL9 Altitude

being the first 84% of the packet, which are mostly expected to be 0, making it near impossible to determine where in the packet it had been 'cut-off'. Out of 4,475 attempts to write packets, none totally completed, but 52 made it to the point where all relevant data was written before the packet was cut off. Table 4.1 shows the structure of the packets written during the sounding rocket flight.

Graphing this data using MATLAB revealed results in Figures 4.5, 4.6, and 4.7. These figures include test data recorded before the flight as a control, represented as the much shallower slope of the Spartan-6 counter.

From Figure 4.5, we can see that the G-switch activated $\approx$ 2 hours, 12 minutes after power-on, and from that point the Virtex-6 die temperature quickly rises, to peak at 83.8125 °C. The $\approx$ 20 minute discontinuity in the roughly linear Spartan-6 counter is due to the readback scrubbing task, during which all other tasks are disabled to avoid an overflow of the scheduler queue.

From Figure 4.6 we see that the tiles quickly begin being marked as faulted, and for the most part do not recover. Unfortunately, it is impossible to tell if the tiles actually recover, due to a bug incrementing the watchdog counter. A watchdog task

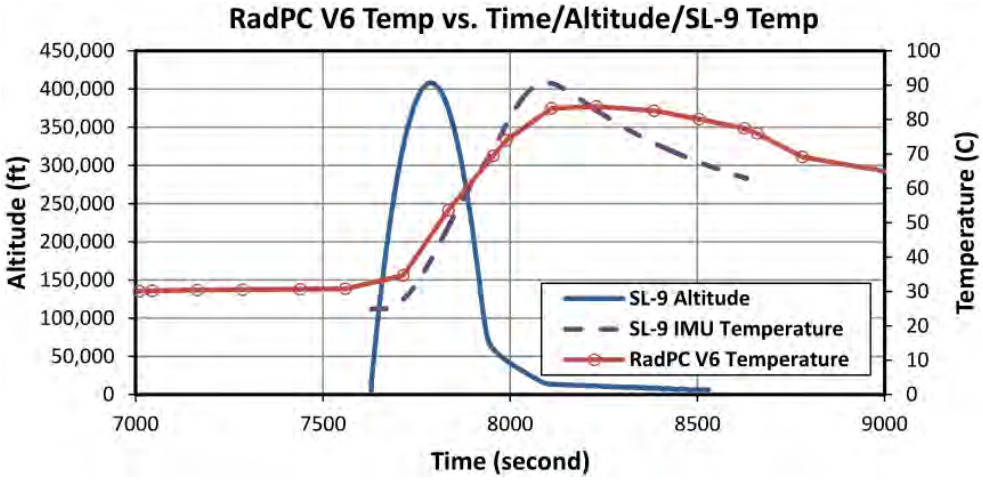

Figure 4.4: SL9 Temperature Comparison

scheduled for every 60 seconds checks the outputs of the tiles, and compares them to the last saved values. If they match, it is assumed the tile has stopped counting and is marked for PR, and a watchdog counter is incremented.

Finally, Figure 4.7 shows the active TMR triad. Unfortunately, due to the long time between packets, it is impossible to know how often the triad was forced to changed by in-system error injection and actual SEUs. It is also impossible to know if the triad is actually operational or crashed, due to the aforementioned watchdog bug.

Lessons Learned

The incomplete data logging of the sounding rocket launch was a primary impetus for the design of a robust data logging system. Post-flight, difficulties in extracting, parsing, and coherently displaying data also developed a requirement for an easy-to-use system. This directly led to investigations into a PC-compatible file system on external media.

Table 4.1: Prototype Artemis Packet Structure

Address (HEX)	Description
000	9-byte Start Pattern
009	16-bit Sensor Intersection Accumulators (x256)
209	16-bit Sensor Front-Side Accumulators (x16)
229	16-bit Sensor Back-Side Accumulators (x16)
249	16-bit Virtex-6 Die Temperature
24B	3-byte Voltage & 2-byte Current (Rail 1D)
250	3-byte Voltage & 2-byte Current (Rail 2D)
255	3-byte Voltage & 2-byte Current (Rail 3D)
25A	3-byte Voltage & 2-byte Current (Rail 4D)
25F	3-byte Voltage & 2-byte Current (Rail 5D)
264	3-byte Voltage & 2-byte Current (Rail 1A)
269	3-byte Voltage & 2-byte Current (Rail 2A)
26E	3-byte Voltage & 2-byte Current (Rail 3A)
273	3-byte Voltage & 2-byte Current (Rail 4A)
278	3-byte Voltage & 2-byte Current (Rail 5A)
27D	64-bit Spartan-6 System Counter
285	16-bit Active Tiles Flags
287	16-bit Faulted Tiles Flags
289	16-bit Readback Faults Counter
28B	8-bit G-Switch Status
28C	16-bit Watchdog Accumulator
28E	7-byte End Pattern

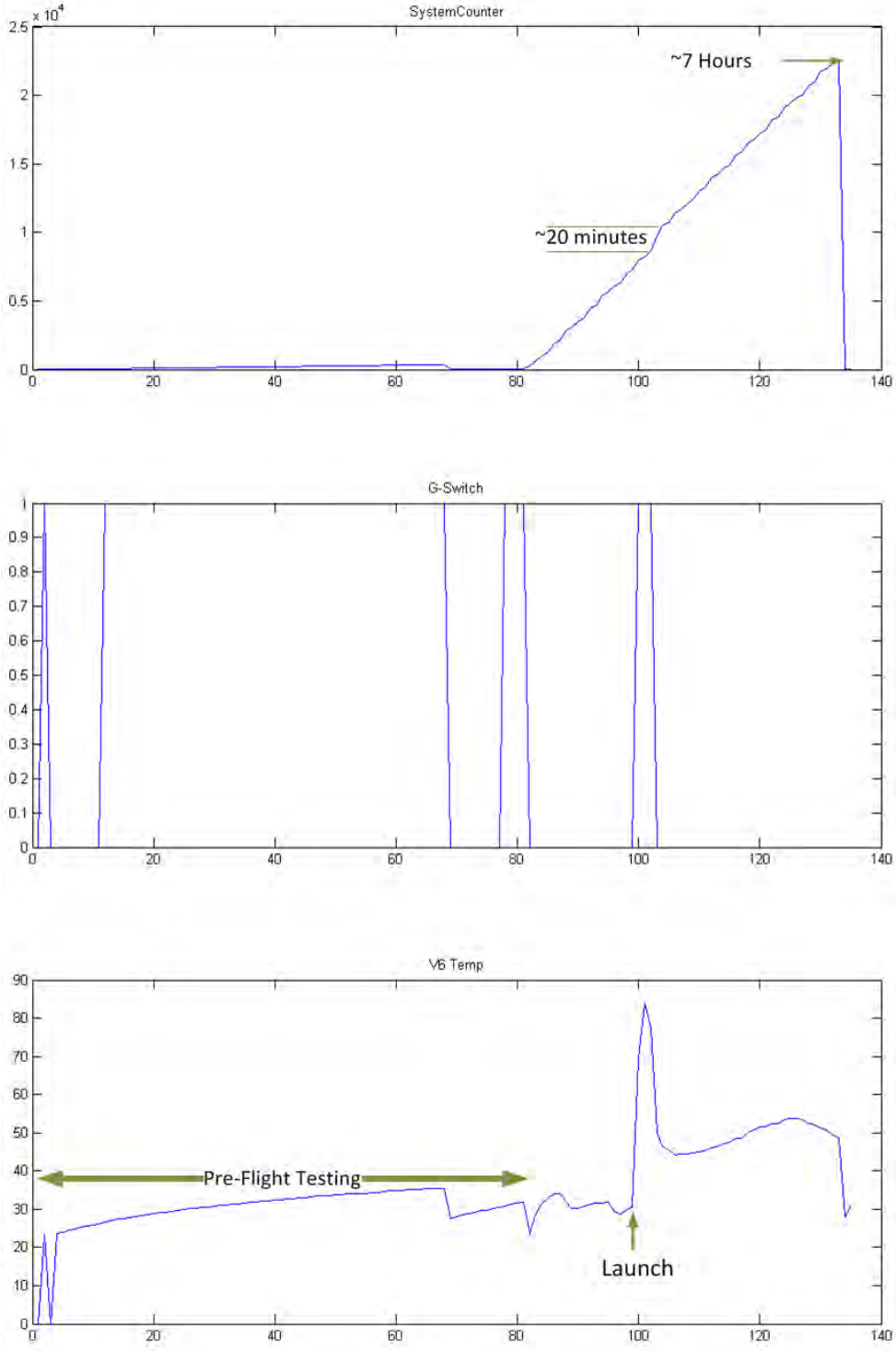

Figure 4.5: G-Switch and Virtex-6 Temperature Data

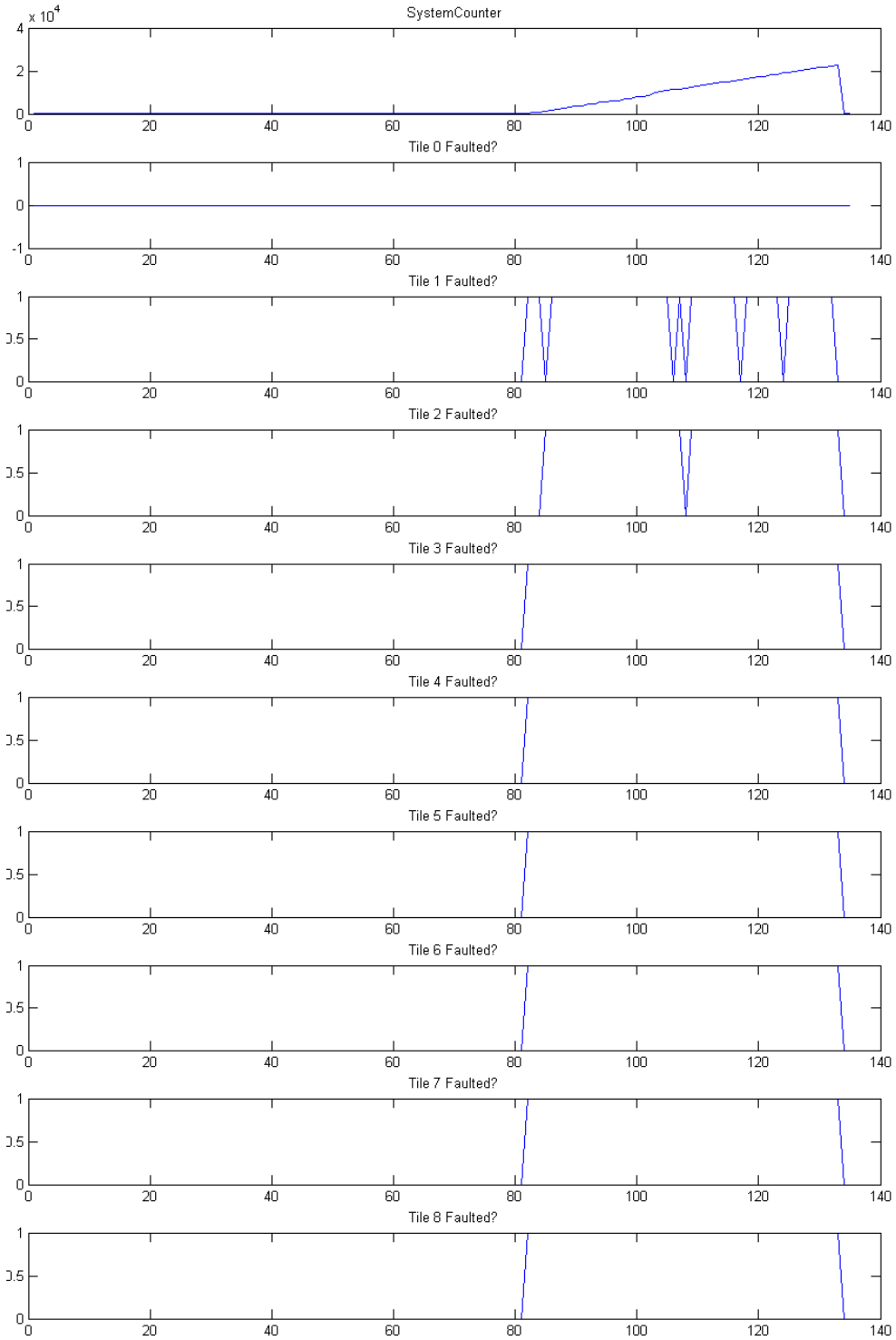

Figure 4.6: Faulted Tiles Data

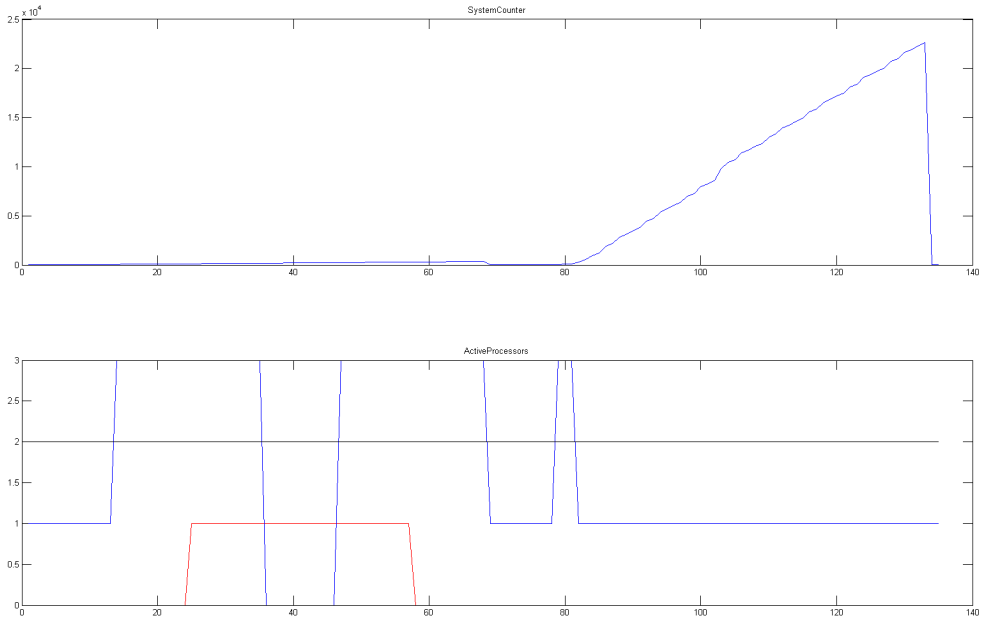

Figure 4.7: Active Tiles Data

High Altitude Balloon

An Artemis engineering model was flown on a Worldview Enterprises high altitude balloon platform. This balloon payload reached a maximum altitude of 120,000 ft, and remained at altitude for 10 hours. The Artemis payload was integrated into the Borealis payload, and ran off 5 V battery power from Borealis.

Figure 4.8: Worldview Balloon Launch [17]

Successes

Artemis was a sub-payload of Borealis, so mechanical and electrical interfaces were minimal. A USB-B to flying leads was used to supply 5V and GND, while the

Figure 4.9: Borealis & Artemis Payload [17]

Artemis chassis was friction locked into the Borealis chassis. Due to this simplicity, pre-flight testing of Artemis was simple verification of power-on and configuration.

Failures

The Artemis stack ran for $\approx$ 20 minutes before the system either crashed or hanged in an infinite loop. Review of the files on the DATA SD card and the packets on the FPGA SD card show normal operation until the cut off point. After the flight, the battery pack used to power Artemis was recharged with 1.59 A-h, which would provide $>$ 2 hours of normal operation.

A post-mortem of the power board discovered that the individual regulators had been mistakenly populated with TPS62150s, instead of TPS62130s. The TPS62150 is functionally identical to the TPS62130, but can only source up to 1 A, instead of

the 3 A the TPS62130 is capable of. The 2 rails that could have been responsible for a crash or hang are 1.0 V-S6 and 3.3 V, respectively. Obviously, if the 1.0 V-S6 rail was unable to supply sufficient current to the S6, the FPGA would cease operation. If the 3.3 V rail were to fail, no data would be logged, a full system crash for all intents and purposes.

Further review of the open-source code used to create and write files on the FAT32 file system revealed numerous loops where ControlOS would hang indefinitely if the proper response from the DATA SD card was not received. If the 3.3 V rail crashed in the middle of an SPI command, the DATA SD card would be unable to respond, trapping the processor in an infinite loop. Watchdog timers have since been implemented.

To plot the data recorded during the flight, a Visual C# GUI was developed. Figures 4.11, 4.12, and 4.13 show 3 of the more interesting events during flight. From the beginning of the flight, we see that the Virtex-6 is faulted before we can even begin writing files at the 30 second mark. In Figure 4.10, we see that 2 of the Virtex-6 tiles are behaving normally for the first 2 packets (1 minute) of flight, but quickly diverge. The length of time between scheduled partial reconfiguration of faulted tiles should have been long enough to allow reconfiguration of all tiles before the next scheduled event. Therefore, it an be concluded that an error in either the FPGA SD card controller or the Virtex-6 configuration controller occurred, leading to problems with partial reconfiguration.

Lessons Learned

These bugs were not detected due to the compressed schedule of the mission. Successful writing of files was only achieved < 48 hours before travel to the mission site. Immediately after post-mortem debug began, the incorrect regulators were

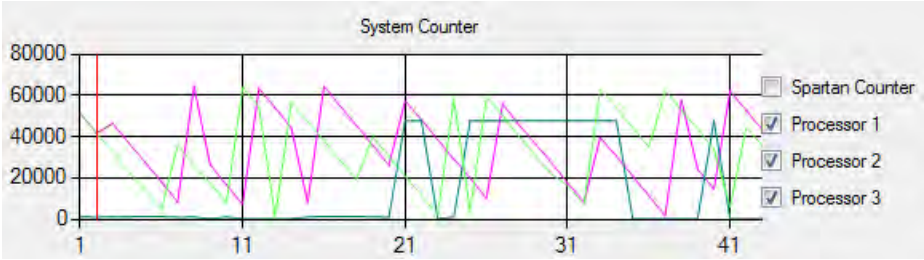

Figure 4.10: Worldview Flight Data Virtex-6 Counters

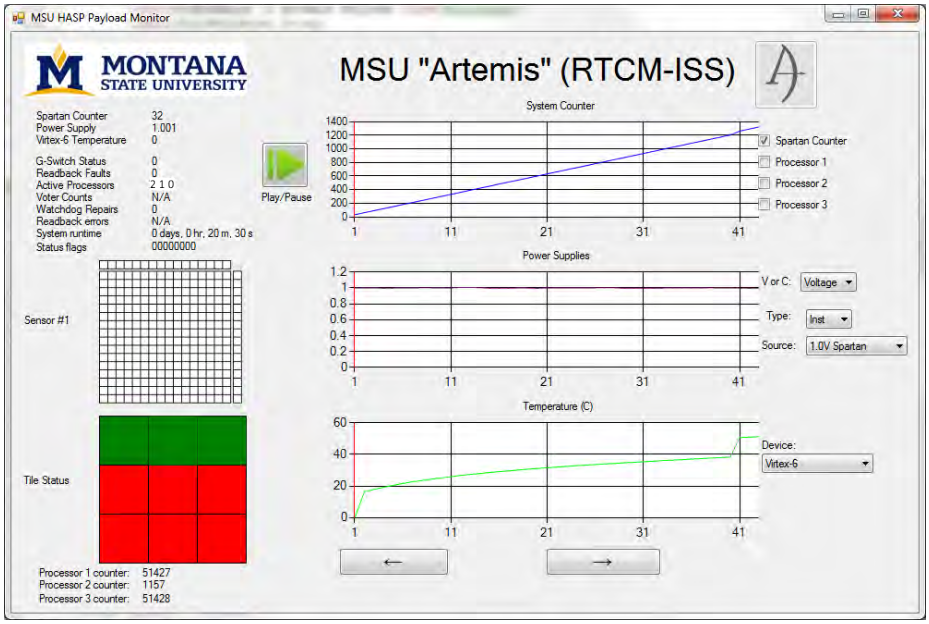

Figure 4.11: Worldview Flight Data File 0

discovered. Following their replacement, the infinite loops were discovered when the system hanged while connected to a PC. Finally, the 512 file limit in the root directory of FAT16 was only discovered when $>$ 4 hours had passed with a write period of 30 seconds.

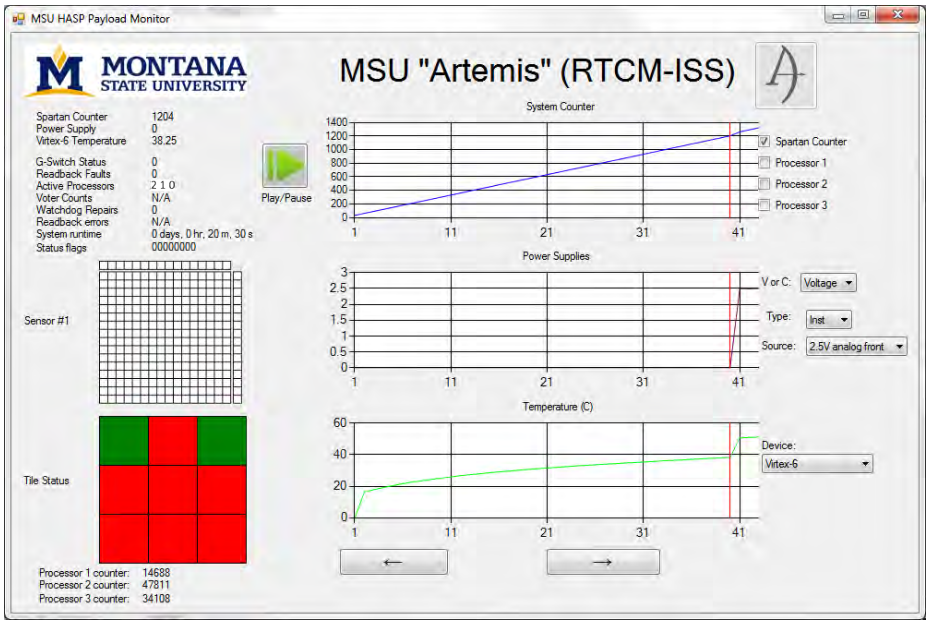

Figure 4.12: Worldview Flight Data File 39

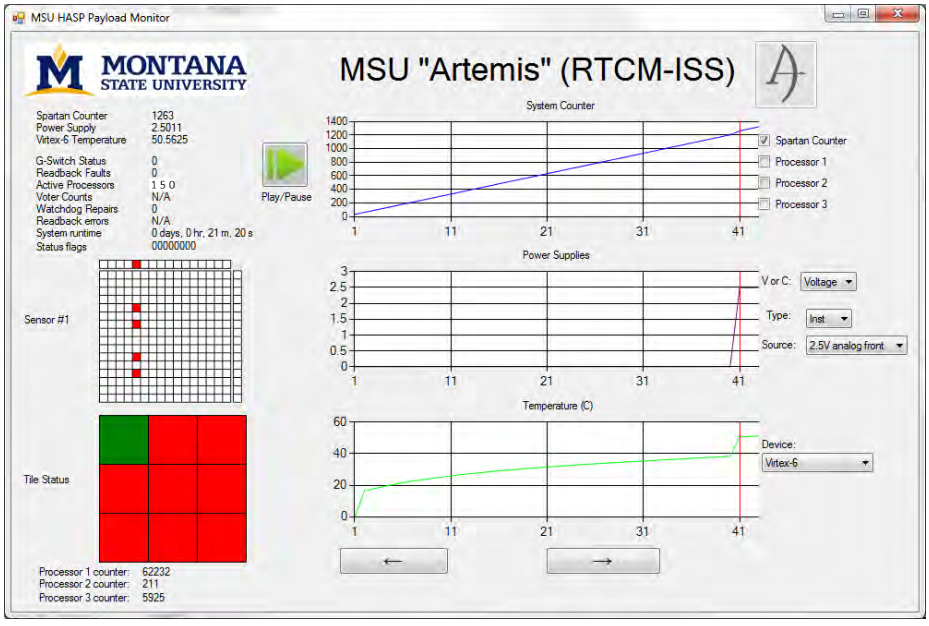

Figure 4.13: Worldview Flight Data File 41

Thermal Testing

Benchtop testing is useful, but lacks some of the unique conditions Artemis will face when part of a payload. Space environments are extremely hostile even without considering radiation effects. In atmosphere, convective cooling aided by conductive cooling help moderate the heat energy given off by electronics. In space applications, the reliance on radiative cooling makes thermal budgeting a much tighter design constraint. To better characterize Artemis, SSEL's thermal chamber was used to subject the computer system to more extreme temperatures.

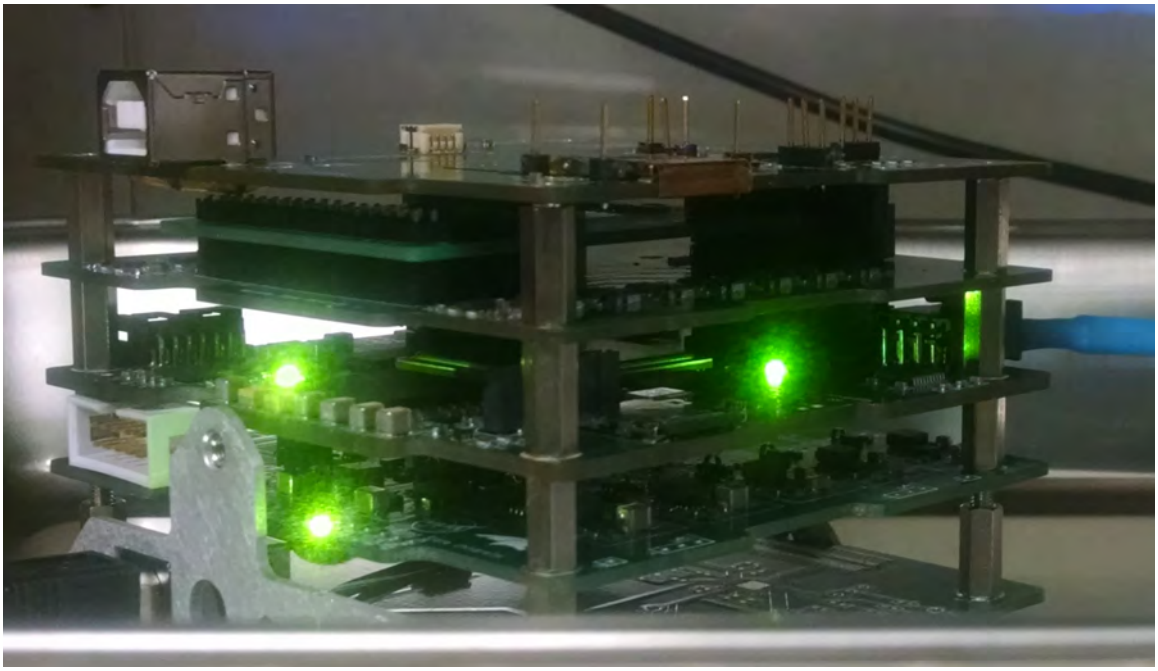

Figure 4.14: Side Profile Shot of Artemis in Thermal Chamber

Successes

The chamber temperature was raised to 40°C over 10 minutes, and allowed to run for another 10 minutes. Then the chamber temperature was lowered to 20°C. Upon

Figure 4.15: Fully Assembled Artemis ready for Thermal Test

reaching the desired temperature, the climate control was disabled, and the chamber allowed to return to room temperature. In total, the Artemis stack was run for 42.5 minutes.

Figure 4.17 shows the lock-stepped Virtex-6 processors and a sudden decrease in the current on the 1.8 V rail. The only components that utilize the 1.8 V rail are the Virtex-6, the DDR2, the 0.9 V regulator which provides a reference voltage for the DDR2, and the Spartan-6 PROM. Since the Virtex-6 only uses 1.8 V to connect to the DDR2, and the demonstrated experiment does not utilize the DDR2, the current draw should only be the quiescent currents for Virtex-6 IO and DDR2. The voltage comparator should also provide a static draw through the run-time of the experiment. The configuration PROM will require more power during initial configuration of the Spartan-6, but should then remain inactive during normal operation.

Figure 4.18 shows the linearly increasing Spartan-6 counter, and a correlated current draw to temperature on the Virtex-6. The decreasing temperature in the graph is due to the chamber temperature being lowered to 20°C.

Figure 4.16: Fully Assembled Artemis inside Thermal Chamber

In addition, by comparing Figures 4.17 and 4.18 we can see the active processors moved from 302 to 021. This shows fault injection is triggering as designed. With the knowledge that the active triad of processors remains in lockstep throughout the runtime, we can conclude that partial reconfiguration is also successful.

Failures

The DATA SD card's operation appears to be very closely tied to temperature. When the chamber temperature was above 50°C, it was almost guaranteed the system would not be able to detect the card and hang. Furher testing using localized heat and a external temperature sensor should be conducted to determine the exact cause of this bug. Initial suggestions include mechanical problems due to thermal expansion of the SD card holder, increased thermal noise ruining signal integrity, and exceeding the operating temperature of the MAX14502 or SD card.

Above 60°C, errors with the Virtex-6 begin to appear. The TMR triad continues counting, but quickly leaves lockstep, and is unable to recover even with partial

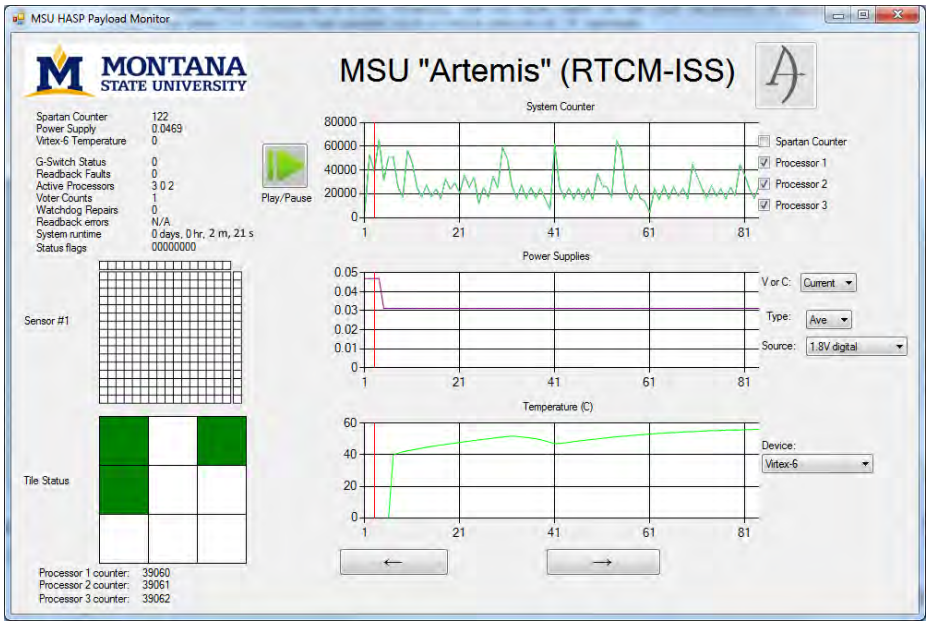

Figure 4.17: GUI display of file 3 of the Thermal Chamber test run

reconfiguration, similar to the results of the Worldview balloon test. This may be tied to the increasing current correlated to increasing temperature.

Lessons Learned

After the first run in the thermal chamber, it was discovered not a single file had been written to the SD card. As a PC connection still showed a healthy SD card, the cause could be narrowed to the Sensor connector or further bugs in ControlOS. Through extensive use of the Microblaze debugger, it was discovered the code was failing when attempting to read the Master Boot Record of the DATA SD card. For unknown reasons, the cards were being formatted by Windows 7 incorrectly, placing what seemed to be error messages where partition Boot Record addresses would normally be. To fix this, the Boot Record addresses were found manually and inserted in at the correct offset using a hex editor. See Appendix C for more details.

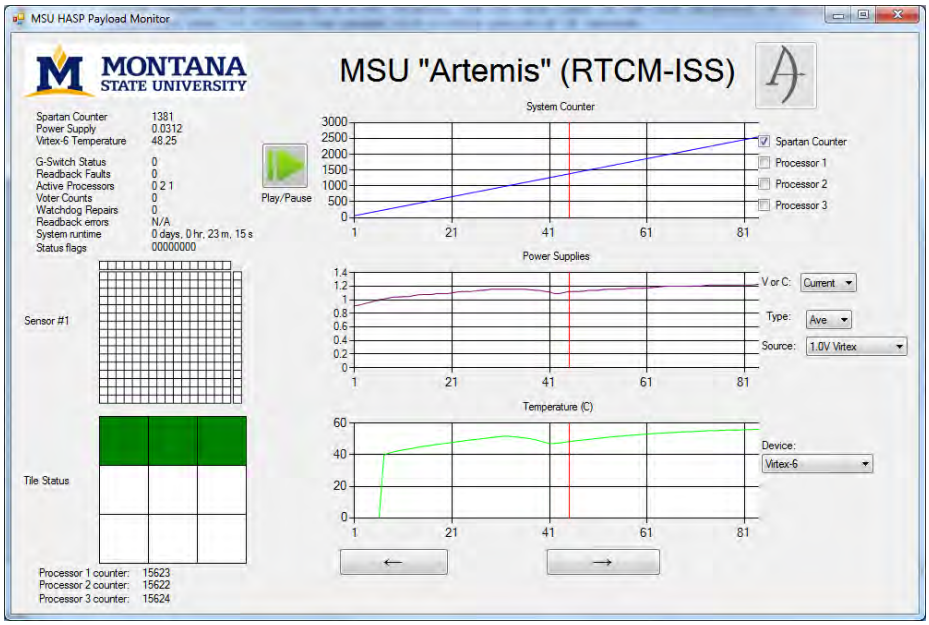

Figure 4.18: GUI display of file 45 of the Thermal Chamber test run

CONCLUSION AND FUTURE WORK

This reseach showed the design and construction of a radiation hardened FPGA computer system with a robust data-logging system. This expands the functionality and usability of the existing CubeSat design. To verify system design, the system was tested in laboratory environment (thermal chamber), and representative demonstrations (sounding rocket and high altitude balloon).

Some of the functions that ControlOS performs, specifically Readback Scrubbing and writing files to the DATA SD card, can be parallelized to improve performance and usability. Partial Reconfiguration and Blind Scrubbing are already parallelized with VHDL state machines which perform the bulk of the heavy computation. Readback Scrubbing could either be performed by a separate, smaller Microblaze running in parallel, or in a VHDL state machine. A second microblaze could poll the global variables recorded in the text files without modifying their contents, allowing new files to be written witout pausing the rest of the system. This would enable ControlOS to continue logging data while intensive functions were being performed.

REFERENCES CITED

[1] J. Barth, C. Dyer, and E. Stassinopoulos, "Space, atmospheric, and terrestrial radiation environments," *Nuclear Science, IEEE Transactions on*, vol. 50, pp. 466–482, 2003.

[2] A. Holmes-Siedle and L. Adams, *Handbook of Radiation Effects*. Oxford University Press, USA, 2002.

[3] H. H. Heckman and A. H. Armstrong, "Energy spectrum of geomagnetically trapped protons," *Journal of Geophysical Research*, vol. 67, no. 4, pp. 1255–1262, 1962. [Online]. Available: http://dx.doi.org/10.1029/JZ067i004p01255

[4] "Virtex-6 fpga configuration," Xilinx, November 2013.

[5] "Spartan-6 fpga configuration," Xilinx, January 2013.

[6] M. Vega, "Monitoring voltage, current, and temperature using the ucd90xxx devices," 2012.

[7] J. A. Eddy. (1979) A new sun. George C. Marshall Space Flight Center. [Online]. Available: http://history.nasa.gov/SP-402/contents.htm

[8] R. B. Horne, "Plasma astrophysics: Acceleration of killer electrons," *Nat Phys*, vol. 3, no. 9, pp. 590–591, Sep 2007. [Online]. Available: http://dx.doi.org/10.1038/nphys703

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

[10] R. Cliff, V. Danchenko, E. Stassinopoulos, M. Sing, G. Brucker, and R. S. Ohanian, "Prediction and measurement of radiation damage to cmos devices on board spacecraft," *Nuclear Science, IEEE Transactions on*, vol. 23, no. 6, pp. 1781–1788, 1976.

[11] S. Swordy, "Cosmic rays," August 2006. [Online]. Available: http://apod.nasa.gov/apod/ap060814.html

[12] B. J. LaMeres. (2012) Fpga-based radiation tolerant computing.

[13] J. H. Keys, Andrew and; Adams, J. D. Cressler, M. C. Patrick, M. A. Johnson, and R. C. Darty, "Radiation hardened electronics for space environments (rhese) project overview," in *Extreme Environments: Sixth International Planetary Probe Workshop*, 6 2008.

[14] T. M. Buerkle, "Ionizing radiation detector for enviromental awareness in fpga-based flight computers," Master's thesis, Montana State University, May 2012.

[15] J. A. Hogan, "Reliability analysis of radiation induced fault mitigation strategies in field programmable gate arrays," Ph.D. dissertation, Montana State University, April 2014.

[16] P. A. De Leon, "Sl9 photo album," October 2014, photo.

[17] @WorldViewVoyage. (2015). [Online]. Available: https://twitter.com/WorldViewVoyage/media

[18] M. Berg, H. Kim, C. Perez, and A. Phan, "Xilinx virtex 5 proton accelerated radiation test," NASA Goddard, Test Report, January 2010. [Online]. Available: http://radhome.gsfc.nasa.gov/radhome/papers/D062209_XCVLX30T.pdf

[19] T. Flatley, "Advanced hybrid on-board science data processor - spacecube 2.0," in *NASA/ESA Conference on Adaptive Hardware and Systems (AHS-2012)*, 2012. [Online]. Available: http://esto.nasa.gov/conferences/estf2011/papers/Flatley_ESTF2011.pdf

[20] M. Lin, T. Flatley, A. Geist, and D. Petrick, "Nasa gsfc development of the spacecube mini," in *25th Annual AIAA/USU Conference on Small Satellites*, 2011. [Online]. Available: http://esto.nasa.gov/conferences/estf2011/papers/Flatley_ESTF2011.pdf

[21] A. S. Keys, J. H. Adams, M. C. Patrick, M. A. Johnson, and J. D. Cressler, "A review of nasa's radiation-hardened electronics for space environments project," in *SPSC 2010*, September 2008.

[22] J. H. Adams, Jr., R. Silberberg, and C. H. Tsao, "Cosmic ray effects on microelectronics. part 1: The near-earth particle environment," *NASA STI/Recon Technical Report*, vol. 81, p. 34134, Aug. 1981. [Online]. Available: http://www.dtic.mil/cgi-bin/GetTRDoc?Location=U2&doc=GetTRDoc.pdf&AD=ADA103897

[23] "Linear energy transfer," ICRU, Tech. Rep., 1970.

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

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

[26] C. Gwyn, "Model for radiation-induced charge trapping and annealing in the oxide layer of mos devices," *Journal of Applied Physics*, vol. 40, no. 12, 1969.

[27] M. Gadlage, R. Schrimpf, J. Benedetto, P. Eaton, D. Mavis, M. Sibley, K. Avery, and T. Turflinger, "Single event transient pulse widths in digital microcircuits," *Nuclear Science, IEEE Transactions on*, vol. 51, pp. 3285–3290, 2004.

[28] F. Sexton, "Destructive single-event effects in semiconductor devices and ics," *Nuclear Science, IEEE Transactions on*, vol. 50, pp. 603–621, 2003.

[29] J. R. Schwank, "Space and military radiation effects in silicon-on-insulator devices." Sandia National Labs, Albuquerque, NM., Tech. Rep., Sep 1996. [Online]. Available: http://www.osti.gov/energycitations/servlets/purl/279707-Khmy0T/webviewable/279707.pdf

[30] S. Kerns, B. Shafer, J. Rockett, L.R., J. Pridmore, D. Berndt, N. van Vonno, and F. Barber, "The design of radiation-hardened ics for space: a compendium of approaches," *Proceedings of the IEEE*, vol. 76, no. 11, pp. 1470–1509, 1988.

[31] J. Rhea, "Bae systems moves into third generation rad-hard processors," *Military and Aerospace Electronics*, 2002.

[32] B. Pratt, M. Caffrey, P. Graham, K. Morgan, and M. Wirthlin, "Improving fpga design robustness with partial tmr," in *Reliability Physics Symposium Proceedings, 2006. 44th Annual., IEEE International*, March 2006, pp. 226–232.

[33] R. Lyons and W. Vanderkulk, "The use of triple-modular redundancy to improve computer reliability," *IBM Journal of Research and Development*, vol. 6, no. 2, pp. 200–209, April 1962.

[34] N. Rollins, M. J. Wirthlin, M. Caffrey, and P. Graham, "Evaluating tmr techniques in the presence of single event upsets," in *Proceedings of the 6th Annual International Conference on Military and Aerospace Programmable Logic Devices*, September 2003. [Online]. Available: http://klabs.org/richcontent/MAPLDCon03/abstracts/rollins_a.pdf

[35] F. Lima, L. Carro, and R. Reis, "Designing fault tolerant systems into sram-based fpgas," in *Design Automation Conference, 2003. Proceedings*, June 2003, pp. 650–655.

[36] P. Ostler, M. Caffrey, D. Gibelyou, P. Graham, K. Morgan, B. Pratt, H. Quinn, and M. Wirthlin, "Sram fpga reliability analysis for harsh radiation environments," *Nuclear Science, IEEE Transactions on*, vol. 56, no. 6, pp. 3519–3526, 2009.

[37] J. von Neumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," *Automata Studies*, pp. 43–98, 1956. [Online]. Available: http://www.mriedel.ece.umn.edu/wiki/images/a/af/Von_Neumann_Probabilistic_Logics_and_the_Synthesis_of_Reliable_Organisms_from_Unreliable_Components.pdf

[38] A. Moore, *FPGAs for Dummies*. John Wiley and Sons, New Jersey, 2014.

[39] E. Fong, M. Converse, and P. Denham, "An electrically reconfigurable programmable logic array using a cmos/dmos technology," *Solid-State Circuits, IEEE Journal of*, vol. 19, no. 6, pp. 1041–1043, 1984.

[40] P. Graham, M. Caffery, M. Wirthlin, D. E. Johnson, and N. H. Rollins, "Reconfigurable computing in space: From current technology to reconfigurable systems-on-a-chip," in *http://nhrollins.com/files/publications/second/ieeeaero03.pdf*, 2003. [Online]. Available: http://nhrollins.com/files/publications/second/ieeeaero03.pdf

[41] J. S. Hane, "A fault-tolerant computer architecture for space vehicle applications," Master's thesis, Montana State University, May 2012.

[42] "Logicore ip soft error mitigation controller v4.0," Xilinx, June 2013.

[43] E. C. Gowens, "Two dimensional radiation sensor development for use in space bound reconfigurable computers," Master's thesis, Montana State University, May 2011.

[44] SSEL, "Cubesat board outline," 2013.

[45] "Virtex-6 fpga memory interface solutions," Xilinx, March 2013.

[46] "Max6627/max6628 datasheet," 2014. [Online]. Available: http://www.maximintegrated.com/en/products/analog/sensors-and-sensor-interface/MAX6627.html

[47] M. Peattie, "Using a microprocessor to configure xilinx fpgas via slave serial or selectmap mode," 2009. [Online]. Available: http://www.xilinx.com/support/documentation/application_notes/xapp502.pdf

[48] "Using the tps62120 in an inverting buck-boost topology," Application Note, Texas Instruments, October 2013. [Online]. Available: http://www.ti.com/lit/an/slva478b/slva478b.pdf

[49] "Tps6213x 3v-17v 3a step-down converter in 3x3 qfn package," Texas Instruments, January 2013. [Online]. Available: http://www.ti.com/lit/ds/symlink/tps62130.pdf

[50] "Virtex-6 fpga data sheet: Dc and switching characteristics," Xilinx, May 2013.

[51] "Spartan-6 fpga data sheet: Dc and switching characteristics," Xilinx, October 2011.

[52] IBEX. Secure digital sd card source code driver project. [Online]. Available: http://www.embedded-code.com/manuals/secure-digital-card-driver/

APPENDICES

APPENDIX A

DDR2 CONSTRAINTS

Table A.1: RAM0.ucf

NET	"clk_ref_n"	LOC = "F21"; #Dummy Pin
NET	"clk_ref_p"	LOC = "F20"; #Dummy Pin
NET	"ddr2_addr[0]"	LOC = "AE14";
NET	"ddr2_addr[10]"	LOC = "AD15";
NET	"ddr2_addr[11]"	LOC = "AG16";
NET	"ddr2_addr[12]"	LOC = "AE17";
NET	"ddr2_addr[1]"	LOC = "AH14";
NET	"ddr2_addr[2]"	LOC = "AG14";
NET	"ddr2_addr[3]"	LOC = "AD16";
NET	"ddr2_addr[4]"	LOC = "AE15";
NET	"ddr2_addr[5]"	LOC = "AH15";
NET	"ddr2_addr[6]"	LOC = "AF15";
NET	"ddr2_addr[7]"	LOC = "AD17";
NET	"ddr2_addr[8]"	LOC = "AF16";
NET	"ddr2_addr[9]"	LOC = "AH16";
NET	"ddr2_ba[0]"	LOC = "AD12";
NET	"ddr2_ba[1]"	LOC = "AH13";
NET	"ddr2_ba[2]"	LOC = "AE12";
NET	"ddr2_cas_n"	LOC = "AE13";
NET	"ddr2_ck_n[0]"	LOC = "AG13";
NET	"ddr2_ck_p[0]"	LOC = "AF14";
NET	"ddr2_cke[0]"	LOC = "AD7";
NET	"ddr2_cs_n[0]"	LOC = "AD13";

Continued on next page

Table A.1 – continued from previous page

NET	"ddr2_dm[0]"	LOC = "AE10";
NET	"ddr2_dm[1]"	LOC = "AE9";
NET	"ddr2_dq[0]"	LOC = "AH10";
NET	"ddr2_dq[10]"	LOC = "AH8";
NET	"ddr2_dq[11]"	LOC = "AF9";
NET	"ddr2_dq[12]"	LOC = "AG9";
NET	"ddr2_dq[13]"	LOC = "AG8";
NET	"ddr2_dq[14]"	LOC = "AF7";
NET	"ddr2_dq[15]"	LOC = "AE7";
NET	"ddr2_dq[1]"	LOC = "AF11";
NET	"ddr2_dq[2]"	LOC = "AH11";
NET	"ddr2_dq[3]"	LOC = "AF12";
NET	"ddr2_dq[4]"	LOC = "AG12";
NET	"ddr2_dq[5]"	LOC = "AG11";
NET	"ddr2_dq[6]"	LOC = "AF10";
NET	"ddr2_dq[7]"	LOC = "AH9";
NET	"ddr2_dq[8]"	LOC = "AG7";
NET	"ddr2_dq[9]"	LOC = "AE8";
NET	"ddr2_dqs_n[0]"	LOC = "AA10";
NET	"ddr2_dqs_n[1]"	LOC = "AC11";
NET	"ddr2_dqs_p[0]"	LOC = "Y10";
NET	"ddr2_dqs_p[1]"	LOC = "AD11";
NET	"ddr2_odt[0]"	LOC = "AD10";

Continued on next page

Table A.1 – continued from previous page

NET	"ddr2_ras_n"	LOC = "AC9";
NET	"ddr2_we_n"	LOC = "AD8";
NET	"error"	LOC = "A26"; #Dummy Pin
NET	"phy_init_done"	LOC = "A27"; #Dummy Pin
NET	"sys_clk_n"	LOC = "G21";
NET	"sys_clk_p"	LOC = "H20";
NET	"sys_rst"	LOC = "D28";

Table A.2: RAM1.ucf

NET	"clk_ref_n"	LOC = "F21"; #Dummy Pin
NET	"clk_ref_p"	LOC = "F20"; #Dummy Pin
NET	"ddr2_addr[0]"	LOC = "D13";
NET	"ddr2_addr[10]"	LOC = "A15";
NET	"ddr2_addr[11]"	LOC = "C16";
NET	"ddr2_addr[12]"	LOC = "A17";
NET	"ddr2_addr[1]"	LOC = "C14";
NET	"ddr2_addr[2]"	LOC = "E13";
NET	"ddr2_addr[3]"	LOC = "A16";
NET	"ddr2_addr[4]"	LOC = "C15";
NET	"ddr2_addr[5]"	LOC = "E15";
NET	"ddr2_addr[6]"	LOC = "D15";
NET	"ddr2_addr[7]"	LOC = "D16";
NET	"ddr2_addr[8]"	LOC = "B16";

Continued on next page

Table A.2 – continued from previous page

NET	"ddr2_addr[9]"	LOC = "B17";
NET	"ddr2_ba[0]"	LOC = "B13";
NET	"ddr2_ba[1]"	LOC = "D12";
NET	"ddr2_ba[2]"	LOC = "C13";
NET	"ddr2_cas_n"	LOC = "A12";
NET	"ddr2_ck_n[0]"	LOC = "A14";
NET	"ddr2_ck_p[0]"	LOC = "B14";
NET	"ddr2_cke[0]"	LOC = "B12";
NET	"ddr2_cs_n[0]"	LOC = "D11";
NET	"ddr2_dm[0]"	LOC = "A10";
NET	"ddr2_dm[1]"	LOC = "A9";
NET	"ddr2_dq[0]"	LOC = "C10";
NET	"ddr2_dq[10]"	LOC = "F9";
NET	"ddr2_dq[11]"	LOC = "D8";
NET	"ddr2_dq[12]"	LOC = "E8";
NET	"ddr2_dq[13]"	LOC = "B9";
NET	"ddr2_dq[14]"	LOC = "E7";
NET	"ddr2_dq[15]"	LOC = "D7";
NET	"ddr2_dq[1]"	LOC = "D10";
NET	"ddr2_dq[2]"	LOC = "F10";
NET	"ddr2_dq[3]"	LOC = "F11";
NET	"ddr2_dq[4]"	LOC = "G11";
NET	"ddr2_dq[5]"	LOC = "E10";

Continued on next page

Table A.2 – continued from previous page

NET	"ddr2_dq[6]"	LOC = "E9";
NET	"ddr2_dq[7]"	LOC = "C9";
NET	"ddr2_dq[8]"	LOC = "F7";
NET	"ddr2_dq[9]"	LOC = "G7";
NET	"ddr2_dqs_n[0]"	LOC = "C8";
NET	"ddr2_dqs_n[1]"	LOC = "B7";
NET	"ddr2_dqs_p[0]"	LOC = "B8";
NET	"ddr2_dqs_p[1]"	LOC = "A7";
NET	"ddr2_odt[0]"	LOC = "A11";
NET	"ddr2_ras_n"	LOC = "B11";
NET	"ddr2_we_n"	LOC = "C11";
NET	"error"	LOC = "A26"; #Dummy Pin
NET	"phy_init_done"	LOC = "A27"; #Dummy Pin
NET	"sys_clk_n"	LOC = "G21";
NET	"sys_clk_p"	LOC = "H20";
NET	"sys_rst"	LOC = "D28";

Table A.3: RAM2.ucf

NET	"clk_ref_n"	LOC = "F21"; #Dummy Pin
NET	"clk_ref_p"	LOC = "F20"; #Dummy Pin
NET	"ddr2_addr[0]"	LOC = "AG26";
NET	"ddr2_addr[10]"	LOC = "AE24";
NET	"ddr2_addr[11]"	LOC = "AG28";

Continued on next page

Table A.3 – continued from previous page

NET	"ddr2_addr[12]"	LOC = "AD28";
NET	"ddr2_addr[1]"	LOC = "AF25";
NET	"ddr2_addr[2]"	LOC = "AE25";
NET	"ddr2_addr[3]"	LOC = "AF26";
NET	"ddr2_addr[4]"	LOC = "AE27";
NET	"ddr2_addr[5]"	LOC = "AG27";
NET	"ddr2_addr[6]"	LOC = "AF27";
NET	"ddr2_addr[7]"	LOC = "AD27";
NET	"ddr2_addr[8]"	LOC = "AE28";
NET	"ddr2_addr[9]"	LOC = "AH28";
NET	"ddr2_ba[0]"	LOC = "AD23";
NET	"ddr2_ba[1]"	LOC = "AH24";
NET	"ddr2_ba[2]"	LOC = "AE23";
NET	"ddr2_cas_n"	LOC = "AG24";
NET	"ddr2_ck_n[0]"	LOC = "AB24";
NET	"ddr2_ck_p[0]"	LOC = "AC24";
NET	"ddr2_cke[0]"	LOC = "AD22";
NET	"ddr2_cs_n[0]"	LOC = "AF24";
NET	"ddr2_dm[0]"	LOC = "AD20";
NET	"ddr2_dm[1]"	LOC = "AE19";
NET	"ddr2_dq[0]"	LOC = "AG21";
NET	"ddr2_dq[10]"	LOC = "AF19";
NET	"ddr2_dq[11]"	LOC = "AG19";

Continued on next page

Table A.3 – continued from previous page

NET	"ddr2_dq[12]"	LOC = "AH19";
NET	"ddr2_dq[13]"	LOC = "AH20";
NET	"ddr2_dq[14]"	LOC = "AG17";
NET	"ddr2_dq[15]"	LOC = "AF17";
NET	"ddr2_dq[1]"	LOC = "AH21";
NET	"ddr2_dq[2]"	LOC = "AF22";
NET	"ddr2_dq[3]"	LOC = "AG22";
NET	"ddr2_dq[4]"	LOC = "AF21";
NET	"ddr2_dq[5]"	LOC = "AE22";
NET	"ddr2_dq[6]"	LOC = "AF20";
NET	"ddr2_dq[7]"	LOC = "AE20";
NET	"ddr2_dq[8]"	LOC = "AG18";
NET	"ddr2_dq[9]"	LOC = "AH18";
NET	"ddr2_dqs_n[0]"	LOC = "AC20";
NET	"ddr2_dqs_n[1]"	LOC = "AE18";
NET	"ddr2_dqs_p[0]"	LOC = "AD21";
NET	"ddr2_dqs_p[1]"	LOC = "AD18";
NET	"ddr2_odt[0]"	LOC = "AG23";
NET	"ddr2_ras_n"	LOC = "AB18";
NET	"ddr2_we_n"	LOC = "AH23";
NET	"error"	LOC = "A26"; #Dummy Pin
NET	"phy_init_done"	LOC = "A27"; #Dummy Pin
NET	"sys_clk_n"	LOC = "G21";

Continued on next page

Table A.3 – continued from previous page

NET	”sys_clk_p”	LOC = ”H20”;
NET	”sys_rst”	LOC = ”D28”;

APPENDIX B

ARTEMIS PACKET STRUCTURE

Table B.1: TLM_MODE.txt

Address (HEX)	Description	Sample Value
000	SYNC	FA
001	HEADER	TLM_MODE
002	MODE	00
004	CRC	2A
005	CRC	F8

Table B.2: TLM_TILE.txt

Address (HEX)	Description	Sample Value
000	SYNC	FA
001	HEADER	TLM_TILE
002	S6_COUNT	0000001744
006	ACT_TILES	0210
008	FAULTED_TILES	000
00A	READBACK_FAULTS	0
00C	GSWITCH	0
00D	WATCHDOG	00011
00F	ACT_PROC1	00
010	ACT_PROC2	01
011	ACT_PROC3	02
012	ACT_PROC1_CNT	65535
014	ACT_PROC2_CNT	65535
016	ACT_PROC3_CNT	65535

Continued on next page

Table B.2 – continued from previous page

Address (HEX)	Description	Sample Value
018	CRC	D3
019	CRC	00

Table B.3: TLM_HEALTH.txt

Address (HEX)	Description	Sample Value
000	SYNC	FA
001	HEADER	TLM_HEALTH
002	VOLTAGE_INST_BATT	8.7559
006	VOLTAGE_AVE_BATT	8.5561
00A	VOLTAGE_MAX_BATT	8.8350
00E	VOLTAGE_MIN_BATT	0.0000
012	VOLTAGE_INST_15V0A	15.0088
016	VOLTAGE_AVE_15V0A	14.9959
01A	VOLTAGE_MAX_15V0A	15.1777
01E	VOLTAGE_MIN_15V0A	14.7373
022	VOLTAGE_INST_-3V0A	0.2979
026	VOLTAGE_AVE_-3V0A	0.3001
02A	VOLTAGE_MAX_-3V0A	0.3091
02E	VOLTAGE_MIN_-3V0A	0.2920
032	VOLTAGE_INST_3V3D	3.2903
036	VOLTAGE_AVE_3V3D	3.2958
03A	VOLTAGE_MAX_3V3D	3.3108

Continued on next page

Table B.3 – continued from previous page

Address (HEX)	Description	Sample Value
03E	VOLTAGE_MIN_3V3D	3.2859
042	VOLTAGE_INST_3V0A	3.0510
046	VOLTAGE_AVE_3V0A	3.0082
04A	VOLTAGE_MAX_3V0A	3.1013
04E	VOLTAGE_MIN_3V0A	2.9282
052	VOLTAGE_INST_2V5D	2.4835
056	VOLTAGE_AVE_2V5D	2.4982
05A	VOLTAGE_MAX_2V5D	2.5267
05E	VOLTAGE_MIN_2V5D	2.4828
062	VOLTAGE_INST_2V5FA	2.5055
066	VOLTAGE_AVE_2V5FA	2.4997
06A	VOLTAGE_MAX_2V5FA	2.5103
06E	VOLTAGE_MIN_2V5FA	2.4854
072	VOLTAGE_INST_2V5RA	2.4908
076	VOLTAGE_AVE_2V5RA	2.4989
07A	VOLTAGE_MAX_2V5RA	2.5092
07E	VOLTAGE_MIN_2V5RA	2.4846
082	VOLTAGE_INST_1V8D	1.7908
086	VOLTAGE_AVE_1V8D	1.7978
08A	VOLTAGE_MAX_1V8D	1.8098
08E	VOLTAGE_MIN_1V8D	1.7878
092	VOLTAGE_INST_1V0SD	0.9924

Continued on next page

Table B.3 – continued from previous page

Address (HEX)	Description	Sample Value
096	VOLTAGE_AVE_1V0SD	1.0021
09A	VOLTAGE_MAX_1V0SD	1.0176
09E	VOLTAGE_MIN_1V0SD	0.9900
0A2	VOLTAGE_INST_1V0VD	0.1636
0A6	VOLTAGE_AVE_1V0VD	0.6923
0AA	VOLTAGE_MAX_1V0VD	1.6780
0AE	VOLTAGE_MIN_1V0VD	0.1599
0B2	CURRENT_INST_BATT	0.6406
0B6	CURRENT_AVE_BATT	0.4577
0BA	CURRENT_MAX_BATT	1.2188
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	0.0895
0EA	CURRENT_MAX_3V3D	0.1094

Continued on next page

Table B.3 – continued from previous page

Address (HEX)	Description	Sample Value
0EE	CURRENT_MIN_3V3D	0.0625
0F2	CURRENT_INST_3V0A	0.0000
0F6	CURRENT_AVE_3V0A	0.0000
0FA	CURRENT_MAX_3V0A	0.0000
0FE	CURRENT_MIN_3V0A	0.0000
102	CURRENT_INST_2V5D	0.1875
106	CURRENT_AVE_2V5D	0.2687
10A	CURRENT_MAX_2V5D	0.3438
10E	CURRENT_MIN_2V5D	0.1719
112	CURRENT_INST_2V5FA	0.0781
116	CURRENT_AVE_2V5FA	0.0698
11A	CURRENT_MAX_2V5FA	0.0781
11E	CURRENT_MIN_2V5FA	0.0625
122	CURRENT_INST_2V5RA	0.0781
126	CURRENT_AVE_2V5RA	0.0781
12A	CURRENT_MAX_2V5RA	0.0781
12E	CURRENT_MIN_2V5RA	0.0781
132	CURRENT_INST_1V8D	0.0312
136	CURRENT_AVE_1V8D	0.0312
13A	CURRENT_MAX_1V8D	0.0312
13E	CURRENT_MIN_1V8D	0.0312
142	CURRENT_INST_1V0SD	0.1719

Continued on next page

Table B.3 – continued from previous page

Address (HEX)	Description	Sample Value
146	CURRENT_AVE_1V0SD	0.1845
14A	CURRENT_MAX_1V0SD	0.2188
14E	CURRENT_MIN_1V0SD	0.1562
152	CURRENT_INST_1V0VD	0.0469
156	CURRENT_AVE_1V0VD	0.8028
15A	CURRENT_MAX_1V0VD	1.3438
15E	CURRENT_MIN_1V0VD	0.0469
162	V6_TEMPERATURE	0.0000
164	PC1_TEMPERATURE	0.0000
166	PC1_EXT_TEMPERATURE	0.0000
168	PC2_TEMPERATURE	0.0000
16A	PC2_EXT_TEMPERATURE	0.0000
16C	SYSTEM_RUNTIME	0000d, 02h, 33m, 01s
174	SYSTEM_STATUS_FLAG	00000000
175	CRC	F2
176	CRC	B8

Table B.4: TLM_SENSOR.txt

Address (HEX)	Description	Sample Value
000	SYNC	FA
001	HEADER	TLM_SENSOR
002	X00	00000

Continued on next page

Table B.4 – continued from previous page

Address (HEX)	Description	Sample Value
004	X01	00000
006	X02	00000
008	X03	00000
00A	X04	00000
00C	X05	00000
00E	X06	00000
010	X07	00000
012	X08	00000
014	X09	00000
016	X10	00000
018	X11	00000
01A	X12	00000
01C	X13	00000
01E	X14	00000
020	X15	00000
022	Y00	00000
024	Y01	00000
026	Y02	00000
028	Y03	00000
02A	Y04	00000
02C	Y05	00000
02E	Y06	00000

Continued on next page

Table B.4 – continued from previous page

Address (HEX)	Description	Sample Value
030	Y07	00000
032	Y08	00000
034	Y09	00000
036	Y10	00000
038	Y11	00000
03A	Y12	00000
03C	Y13	00000
03E	Y14	00000
040	Y15	00000
042	X00Y00	00000
044	X00Y01	00000
046	X00Y02	00000
048	X00Y03	00000
04A	X00Y04	00000
04C	X00Y05	00000
04E	X00Y06	00000
050	X00Y07	00000
052	X00Y08	00000
054	X00Y09	00000
056	X00Y10	00000
058	X00Y11	00000
05A	X00Y12	00000

Continued on next page

Table B.4 – continued from previous page

Address (HEX)	Description	Sample Value
05C	X00Y13	00000
05E	X00Y14	00000
060	X00Y15	00000
062	X01Y00	00000
064	X01Y01	00000
066	X01Y02	00000
068	X01Y03	00000
06A	X01Y04	00000
06C	X01Y05	00000
06E	X01Y06	00000
070	X01Y07	00000
072	X01Y08	00000
074	X01Y09	00000
076	X01Y10	00000
078	X01Y11	00000
07A	X01Y12	00000
07C	X01Y13	00000
07E	X01Y14	00000
080	X01Y15	00000
082	X02Y00	00000
084	X02Y01	00000
086	X02Y02	00000

Continued on next page

Table B.4 – continued from previous page

Address (HEX)	Description	Sample Value
088	X02Y03	00000
08A	X02Y04	00000
08C	X02Y05	00000
08E	X02Y06	00000
090	X02Y07	00000
092	X02Y08	00000
094	X02Y09	00000
096	X02Y10	00000
098	X02Y11	00000
09A	X02Y12	00000
09C	X02Y13	00000
09E	X02Y14	00000
0A0	X02Y15	00000
0A2	X03Y00	00000
0A4	X03Y01	00000
0A6	X03Y02	00000
0A8	X03Y03	00000
0AA	X03Y04	00000
0AC	X03Y05	00000
0AE	X03Y06	00000
0B0	X03Y07	00000
0B2	X03Y08	00000

Continued on next page

Table B.4 – continued from previous page

Address (HEX)	Description	Sample Value
0B4	X03Y09	00000
0B6	X03Y10	00000
0B8	X03Y11	00000
0BA	X03Y12	00000
0BC	X03Y13	00000
0BE	X03Y14	00000
0C0	X03Y15	00000
0C2	X04Y00	00000
0C4	X04Y01	00000
0C6	X04Y02	00000
0C8	X04Y03	00000
0CA	X04Y04	00000
0CC	X04Y05	00000
0CE	X04Y06	00000
0D0	X04Y07	00000
0D2	X04Y08	00000
0D4	X04Y09	00000
0D6	X04Y10	00000
0D8	X04Y11	00000
0DA	X04Y12	00000
0DC	X04Y13	00000
0DE	X04Y14	00000

Continued on next page

Table B.4 – continued from previous page

Address (HEX)	Description	Sample Value
0E0	X04Y15	00000
0E2	X05Y00	00000
0E4	X05Y01	00000
0E6	X05Y02	00000
0E8	X05Y03	00000
0EA	X05Y04	00000
0EC	X05Y05	00000
0EE	X05Y06	00000
0F0	X05Y07	00000
0F2	X05Y08	00000
0F4	X05Y09	00000
0F6	X05Y10	00000
0F8	X05Y11	00000
0FA	X05Y12	00000
0FC	X05Y13	00000
0FE	X05Y14	00000
100	X05Y15	00000
102	X06Y00	00000
104	X06Y01	00000
106	X06Y02	00000
108	X06Y03	00000
10A	X06Y04	00000

Continued on next page

Table B.4 – continued from previous page

Address (HEX)	Description	Sample Value
10C	X06Y05	00000
10E	X06Y06	00000
110	X06Y07	00000
112	X06Y08	00000
114	X06Y09	00000
116	X06Y10	00000
118	X06Y11	00000
11A	X06Y12	00000
11C	X06Y13	00000
11E	X06Y14	00000
120	X06Y15	00000
122	X07Y00	00000
124	X07Y01	00000
126	X07Y02	00000
128	X07Y03	00000
12A	X07Y04	00000
12C	X07Y05	00000
12E	X07Y06	00000
130	X07Y07	00000
132	X07Y08	00000
134	X07Y09	00000
136	X07Y10	00000

Continued on next page

Table B.4 – continued from previous page

Address (HEX)	Description	Sample Value
138	X07Y11	00000
13A	X07Y12	00000
13C	X07Y13	00000
13E	X07Y14	00000
140	X07Y15	00000
142	X08Y00	00000
144	X08Y01	00000
146	X08Y02	00000
148	X08Y03	00000
14A	X08Y04	00000
14C	X08Y05	00000
14E	X08Y06	00000
150	X08Y07	00000
152	X08Y08	00000
154	X08Y09	00000
156	X08Y10	00000
158	X08Y11	00000
15A	X08Y12	00000
15C	X08Y13	00000
15E	X08Y14	00000
160	X08Y15	00000
162	X09Y00	00000

Continued on next page

Table B.4 – continued from previous page

Address (HEX)	Description	Sample Value
164	X09Y01	00000
166	X09Y02	00000
168	X09Y03	00000
16A	X09Y04	00000
16C	X09Y05	00000
16E	X09Y06	00000
170	X09Y07	00000
172	X09Y08	00000
174	X09Y09	00000
176	X09Y10	00000
178	X09Y11	00000
17A	X09Y12	00000
17C	X09Y13	00000
17E	X09Y14	00000
180	X09Y15	00000
182	X10Y00	00000
184	X10Y01	00000
186	X10Y02	00000
188	X10Y03	00000
18A	X10Y04	00000
18C	X10Y05	00000
18E	X10Y06	00000

Continued on next page

Table B.4 – continued from previous page

Address (HEX)	Description	Sample Value
190	X10Y07	00000
192	X10Y08	00000
194	X10Y09	00000
196	X10Y10	00000
198	X10Y11	00000
19A	X10Y12	00000
19C	X10Y13	00000
19E	X10Y14	00000
1A0	X10Y15	00000
1A2	X11Y00	00000
1A4	X11Y01	00000
1A6	X11Y02	00000
1A8	X11Y03	00000
1AA	X11Y04	00000
1AC	X11Y05	00000
1AE	X11Y06	00000
1B0	X11Y07	00000
1B2	X11Y08	00000
1B4	X11Y09	00000
1B6	X11Y10	00000
1B8	X11Y11	00000
1BA	X11Y12	00000

Continued on next page

Table B.4 – continued from previous page

Address (HEX)	Description	Sample Value
1BC	X11Y13	00000
1BE	X11Y14	00000
1C0	X11Y15	00000
1C2	X12Y00	00000
1C4	X12Y01	00000
1C6	X12Y02	00000
1C8	X12Y03	00000
1CA	X12Y04	00000
1CC	X12Y05	00000
1CE	X12Y06	00000
1D0	X12Y07	00000
1D2	X12Y08	00000
1D4	X12Y09	00000
1D6	X12Y10	00000
1D8	X12Y11	00000
1DA	X12Y12	00000
1DC	X12Y13	00000
1DE	X12Y14	00000
1E0	X12Y15	00000
1E2	X13Y00	00000
1E4	X13Y01	00000
1E6	X13Y02	00000

Continued on next page

Table B.4 – continued from previous page

Address (HEX)	Description	Sample Value
1E8	X13Y03	00000
1EA	X13Y04	00000
1EC	X13Y05	00000
1EE	X13Y06	00000
1F0	X13Y07	00000
1F2	X13Y08	00000
1F4	X13Y09	00000
1F6	X13Y10	00000
1F8	X13Y11	00000
1FA	X13Y12	00000
1FC	X13Y13	00000
1FE	X13Y14	00000
200	X13Y15	00000
202	X14Y00	00000
204	X14Y01	00000
206	X14Y02	00000
208	X14Y03	00000
20A	X14Y04	00000
20C	X14Y05	00000
20E	X14Y06	00000
210	X14Y07	00000
212	X14Y08	00000

Continued on next page

Table B.4 – continued from previous page

Address (HEX)	Description	Sample Value
214	X14Y09	00000
216	X14Y10	00000
218	X14Y11	00000
21A	X14Y12	00000
21C	X14Y13	00000
21E	X14Y14	00000
220	X14Y15	00000
222	X15Y00	00000
224	X15Y01	00000
226	X15Y02	00000
228	X15Y03	00000
22A	X15Y04	00000
22C	X15Y05	00000
22E	X15Y06	00000
230	X15Y07	00000
232	X15Y08	00000
234	X15Y09	00000
236	X15Y10	00000
238	X15Y11	00000
23A	X15Y12	00000
23C	X15Y13	00000
23E	X15Y14	00000

Continued on next page

Table B.4 – continued from previous page

Address (HEX)	Description	Sample Value
240	X15Y15	00000
242	CRC	E8
243	CRC	18

APPENDIX C

FAT32 INFORMATION

SD cards are organized into 512-byte 'sectors'. Therefore, when referencing sectors it can be safely assumed that increment the sector is the same as incrementing the address 512 bytes (or 0x0200).

FAT files systems use 'little endian' byte ordering.

For the purposes of this research, it is assumed there is only one partition on the FAT32 formatted SD card. This is a much simpler implementation, and the code only assumes there to be one partition. For information on multiple partitions, refer to [52].

Table C.1: Layout of a FAT32 Volume

Start Address	**Size**	**Contents**
0x0000	512 bytes	Master Boot Record (Not dependent on file system. Contains the addresses of the partitions.)
Partition Start Address + 0x0000	512 bytes	The Boot Record. First sector of the partition.
Partition Start Address + 0x0200	As specified in the Boot Record	FAT table 1
Partition Start Address + 0x0200 + (Size of FAT Table · (FAT table #X - 1))	As specified in the Boot Record	FAT table #X (specified by 'Number of Copies of FAT' in Boot Record. A value of 2 is normal)

Continued on next page

Table C.1 – continued from previous page

Start Address	**Size**	**Contents**
Partition Start Address + 0x0200 + (Size of FAT Table · Number of Copies of FAT)	The remaining space in the partition	Data Area

Table C.2: Master Boot Record

Address	**Size**	**Offset**	**Content**
0x0000	446		Boot Up Executable Code and Data
0x01BE	1	0x00	Partition 1 - Current State (0x00=Inactive, 0x80=Active)
0x01BF	1	0x01	Partition 1 - Beginning of Partition - Head (**Deprecated**)
0x01C0	2	0x02	Partition 1 - Beginning of Partition - Cylinder [15:6] - Sector [5:0] **Deprecated**
0x01C2	1	0x04	Partition 1 - Type of Partition (0x0B=FAT32 or 0x0C=FAT32)
0x01C3	1	0x05	Partition 1 - End of Partition - Head (**Deprecated**)
0x01C4	2	0x06	Partition 1 - End of Partiton - Cylinder [15:6] - Sector [5:0] **Deprecated**
0x01C6	4	0x08	Partition 1 - Start Sector
0x01CA	4	0x0C	Partition 1 - Number of Sectors

Continued on next page

Table C.2 – continued from previous page

Address	**Size**	**Offset**	**Content**
0x01CE	16		Partition 2
0x01DE	16		Partition 3
0x01EE	16		Partition 4
0x01FE	1		Boot Signature 0x55
0x01FF	1		Boot Signature 0xAA

Table C.3: FAT32 Boot Record

Offset	**Contents**
0x0000-0x0002	Jump Code + NOP
0x0003-0x000A	8 byte OEM Name
0x000B-0x000C	Bytes Per Sector
0x000D	Sectors Per Cluster (Restricted to powers of 2 (1, 2, 4, 8, 16, 32))
0x000E-0x000F	Reserved Sectors
0x0010	Number of Copies of FAT. (A value of 2 is recommended — values other than 2 are possible but are not recommended by Microsoft)
0x0011-0x0012	Maximum Root Directory Entries (not applicable for FAT32)
0x0013-0x0014	Number of Sectors in Partition Smaller than 32MB (not applicable for FAT32)
0x0015	Media Descriptor (F8h for Hard Disks)

Continued on next page

Table C.3 – continued from previous page

Offset	**Contents**
0x0016-0x0017	Sectors Per FAT (not applicable for FAT32 bigger field below)
0x0018-0x0019	Sectors Per Track
0x001A-0x001B	Number of Heads
0x001C-0x001F	Number of Hidden Sectors in Partition
0x0020-0x0023	Number of Sectors in Partition
0x0024-0x0027	Number of Sectors Per FAT
0x0028-0x0029	Flags: [15:8] Reserved [7] 1 = FAT Mirroring is Disabled, only 1 FAT is active 0 = FAT Mirroring is Enabled into all FATs [6:4] Reserved [3:0] Number of active FAT (0-#). Only valid if mirroring disabled.
0x002A-0x002B	Version of FAT32 Drive (high byte = major version, low byte = minor version)
0x002C-0x002F	Cluster Number of the Start of the Root Directory (Usually 2, but not required to be)
0x0030-0x0031	Sector Number of the File System Information Sector (Referenced from the start of the partition)

Continued on next page

Table C.3 – continued from previous page

Offset	**Contents**
0x0032-0x0033	Sector Number of the Backup Boot Sector (Referenced from the start of the partition)
0x0034-0x003F	Reserved (12 bytes)
0x0040	Logical Drive Number of Partition
0x0041	Unused
0x0042	Extended Signature (0x29)
0x0043-0x0046	Serial Number of Partition
0x0047-0x0051	11 byte volume name of the partition
0x0052-0x0059	8 byte FAT Name (FAT32)
0x005A-0x01FD	420 bytes of executable code and data
0x01FE	Boot Signature 0x55
0x01FF	Boot Signature 0xAA

Table C.4: FAT32 File Allocation Table

Offset	**FAT Entry**	**Contents**
0x00-0x03	1	Reserved. Contains the media type value in the low 8 bits and all other bits are 1
		Continued on next page

Table C.4 – continued from previous page

Offset	**FAT Entry**	**Contents**
0x04-0x07	2	Reserved - set on format to the EOC marker. The top 2 bits may be used as 'dirty volume' flags: [27] 1 = volume is 'clean' [26] 1 = no disk read/write errors encountered
0x08-0x0B	3	FAT entry for the 1st data cluster
0x0C-0x0F	4	FAT entry for the 2nd data cluster
—	—	
0xXX-0xXX	X	FAT entry for the last data cluster

FAT32 uses up to 4 FATs, though the number is recommended to be 2 due to old systems that assume a value of 2. This provides a backup in case of corruption in one of the tables. Each additional FAT follows straight on after the previous. The possible values for a FAT entry for a data cluster are located in Table C.5.

Table C.5: FAT32 Table Entry Values

0x00000000	The cluster is free
0x00000001	Reserved
0x00000002- 0x0000FFF0	This cluster is used. The value indicates the next cluster number for the file.
0xFFFFFFF7	Cluster is bad
0xFFFFFFF8- 0xFFFFFFFF	End of Clusterchain (EOC)

A FAT directory entry is simply a 'file' containing a linear list of 32 byte entries. The only special directory, which must always be present, is the root directory. For FAT32 the root directory can be of variable size and is a cluster chain just like any other directory. The first cluster of the root directory is specified in the Boot Record.

Table C.6: FAT Directory Entry

Offset	**Contents**
0x00-0x07	8 character filename
0x08-0x0A	3 character filename extension
0x0B	Attributes [7] - 0 [6] - 0 [5] - Archive [4] - Directory [3] - Volume Label [2] - System [1] - Hidden [0] - Read Only
0x0C	0
0x0D	Created time - ms
0x0E-0x0F	Created time - hour and minute
0x10-0x11	Created date
0x12-0x13	Last accessed date

Continued on next page

Table C.6 – continued from previous page

Offset	**Contents**
0x14-0x15	Extended Attribute (reserved for OS/2, always 0) High word of cluster for FAT32 volumes
0x16-0x17	Time of last write to file
0x18-0x19	Date of last write to file
0x1A-0x1B	Start cluster (referenced from the start of the data area of the volume)
0x1C-0x1F	File size