TRIGA: Telecommunications Protocol Processing Subsystem Using Reconfigurable Interoperable Gate Arrays

Jackson Pang, Paula J. Pingree, J. Leigh Torgerson
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109
{Jackson.Pang,Paula.J.Pingree,Jordan.L.Torgerson}@jpl.nasa.gov

Abstract

We present the Telecommunications protocol processing subsystem using Reconfigurable Interoperable Gate Arrays (TRIGA), a novel approach that unifies fault tolerance, error correction coding and interplanetary communication protocol offloading to implement CCSDS File Delivery Protocol and Datalink layers. The new reconfigurable architecture offers more than one order of magnitude throughput increase while reducing footprint requirements in memory, command and data handling processor utilization, communication system interconnects and power consumption.

1. Introduction

The vision of a manned mission to the Moon, Mars and continued space exploration within our solar system has created a clear need for an interplanetary communications network consisting of telecommunications relay satellites, spacecraft, deep space probes, and planetary rovers. The evolution of spacecraft science instrumentation and the need for higher telecommunication bandwidths are placing an increased burden on the flight processors aboard NASA spacecraft. With these new telecommunication system requirements, the need for a high-speed, reliable, and efficient communication protocol has become apparent. An international committee of space agencies known as the Consultative Committee for Space Data Systems (CCSDS) developed the CCSDS File Delivery Protocol (CFDP), which provides reliable communication in high-delay networks along with the CCSDS packet and data link layers. However, with the increased complexity of this protocol stack came an increased burden on the flight processor, taking valuable processing cycles away from flight control software and data processing tasks.

The advent of the reconfigurable hardware and its use in NASA's Deep Space Missions such as the twin Mars Exploration Rovers and the Mars Reconnaissance Orbiter's Electra Software Defined Radio has provided us valuable experience in using SRAM-based FPGAs in deep space environment. The increasing capabilities and the performance benefits that FPGAs offer allow us to expand their use in avionics and telecommunications. Although the SRAM-based FPGAs are more susceptible Single Event Upset episodes compared to the flash-based FPGAs that offer less logic gates and features, both JPL and the aerospace industry has teamed up to form the Single-Event Effects Consortium to study the SEU mitigation techniques such as the configuration memory scrubbing and Triple Module Redundancy(TMR) to extend the fault tolerance of SRAM-based FPGAs. In this paper, we present the Telecommunications protocol processing subsystem using Reconfigurable Interoperable Gate Arrays (TRIGA) that unifies the fault tolerance, error correction coding, CCSDS Datalink Layers and File Delivery Protocol Processing modules in the spacecraft communication subsystem into a single reconfigurable module that is fully interoperable with other CCSDS CFDP reference implementations. This new architecture offers more than one order of magnitude throughput increase while reducing footprint requirements in memory, processor utilization, communications system interconnects and power consumption. The remaining sections of the paper is organized as follows.

In Section 2, we briefly describe the CCSDS file delivery protocol stack and the processing burden that it imposes on the Command and Data Handling (C&DH) subsystem that motivated us to develop TRIGA. Section 3 explains the TRIGA architecture and the advantages it offers over the traditional CFDP software processing. In addition, the next-generation system architectures that incorporate TRIGA are presented in sections 4. Single Even Upset (SEU) mitigation solutions for TRIGA is described in Section 5. Section 6 summarizes our contributions and describes our future work.

2. Motivation

Although CCSDS protocol stack (Table 1) and its user interface procedures are standardized with a

Table 1. CCSDS Protocol Stack Functions

OSI Layers	CCSDS Layers	CCSDS Protocol	
		Uplink to Spacecraft	Downlink from Spacecraft
Application	Application	CFDP [1]	CFDP
Transport	Transport		
Network	Logical Data Path (LDP)	Space Packet [2]	Space Packet
Data Link	Data Link Protocol Sublayer	Telecommand Space Data Link Protocol [3]	Telemetry Space Data Link Protocol [4]
	Channel Coding Sublayer	Telecommand Synchronization and Channel Coding [5]	Telemetry Synchronization and Channel Coding [6]
Physical	Physical	UHF/KA-Band/S-Band	UHF/KA-Band/S-Band

vision to create seamless space communication, its implementation opens up many issues regarding the resource and processing requirement on the spacecraft C&DH computer. To date, two NASA Discovery missions, namely Deep Impact from JPL/NASA and MESSENGER from APL/JHU[1], have implemented a variant of CFDP as part of their spacecraft communication subsystem, but both implementations were forced to strip out many features of CFDP to make it reasonable for the available spacecraft resources. In addition, auxiliary hardware devices were also required to process the lower layers below CFDP transport layer of the CCSDS protocol stack.

2.1. Deep Impact CFDP Usage

In January 2005, Deep Impact mission launched the two joined spacecraft Flyby and Impactor. Six months later, the Impactor module separated and collided into comet Temple 1 while the Flyby module gathered the secrets of our universe's origin by examining the impact crater and ejecta using the high resolution imaging and infrared spectroscopy. Earlier in the mission phase for flight software requirement review, the deliberation was made to use CFDP for its wealth of features and for being the international CCSDS standard. The Deep Impact software implementation of CFDP is slightly modified in that the Earth to Flyby is in acknowledged mode, between Flyby and Impactor in unacknowledged mode, and both Flyby to Earth is done in unacknowledged mode.[2] On both the Impactor and the Flyby, CCSDS Datalink layer functions (i.e. framing, channel coding and synchronization) are accomplished in a hardware device called the Command and Telemetry Board (CTB). (See Figure 1) [7]

[1]Applied Physics Laboratory at the Johns Hopkins University
[2]For brevity, this usage of CFDP is herein referred to as DI-CFDP.

2.2. MESSENGER and CFDP

Launched in the summer of 2004, MESSENGER (MErcury Surface, Space ENvironment, GEochemistry and Ranging) spacecraft is headed to Mercury for a one-year science mission starting in 2011. To compliment the newly augmented file management system and the large data volume that the spacecraft will generate throughout the course of its mission, CFDP was selected as the file transfer mechanism. However, the shared processing resources available onboard (25MHz RAD7000 PPC CPU, 8MB RAM, 4MB EEPROM) for CFDP, Guidance and Control, and image compression made it prohibitive to use the full CFDP reference implementation maintained by JPL. Consequently, the customized version of APL CFDP called CFDP-lite was engineered to couple the CFDP closely with the constrained memory availability and other spacecraft file management tasks. In CFDP-lite, APL successfully implemented the unacknowledged downlink portion of software CFDP. Similarly on MESSENGER, a hardware module was used for the Data Link and Coding tasks. [8] This type of CFDP usage has been implemented on the Mars Reconnaissance Orbiter and future missions such as Mars Science Laboratory and JUNO.

2.3. CCSDS Link, Channel and Coding Layers

CCSDS also defines a set of lower protocol layers, namely Space Packet, Telecommand and Telemetry layers, to create the protocol stack that supports deep space telecommunication needs. From the processing point of view, these layers require two types of services, namely packetizing a frame with a given header and computationally intensive tasks such as Cyclic Redundancy Coding, Reed-Solomon and Convolutional Coding. Due to the inherently independent and parallel nature of these requirements, their functionalities were implemented in dedicated hardware for efficiency.

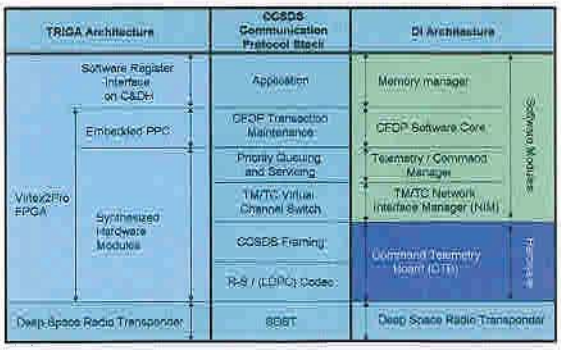


Figure 1. Deep Impact Communication Subsystem Components Redefined

By reflecting on the current implementation strategies of the CCSDS protocol stack, we can deduce that a unified communication protocol processing subsystem can alleviate many of the compromises we face regarding how we should tailor CFDP for a specific mission. CFDP as a complete file delivery protocol is designed well, but the approaches such as DI-CFDP and CFDP-lite removes CFDP's salient features such as multihop and automatic repeat request capabilities. Hence, CFDP cannot shine as a reliable transport protocol but merely as a remote file organizing scheme, which may hinder the wide deployment of CFDP due to misunderstanding of the potential benefits of CFDP. In addition, the recurring engineering and risk assessment costs related to producing a CFDP variant is driving the mission expenditure up.

At the system level, the requirement of dedicated hardware such as the Command and Telemetry Board used on Deep Impact for lower layer protocol processing adds design complexity, power consumption and real estate requirements, all of which are primary considerations made in the design and cost estimation of next generation spacecraft. An architecture that decouples file transfer functions from the rest of the C&DH system while integrating the full functionality of the file delivery protocol stack into one subsystem is highly desirable. Hence, we engineered TRIGA to provide the features of a reliable communication subsystem including the solutions to mitigate the radiation induced Single Event Upset scenarios discussed in Section 5.

3. The Zero-copy TRIGA Architecture

The TRIGA architecture is designed to support high throughput with minimal support from the C&DH system. At the very high level, TRIGA look much like a direct memory access engine. As shown in Figure 2, TRIGA interfaces directly to the data storage and the radio interface using high speed data channels. The

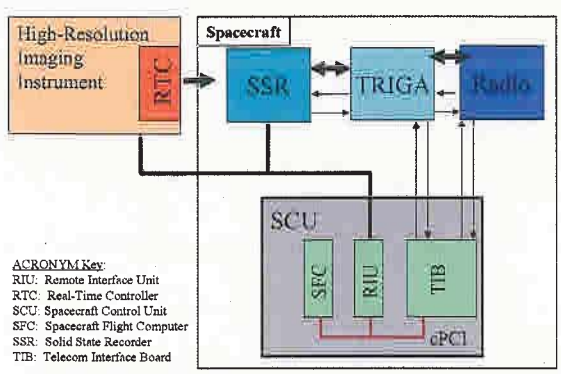


Figure 2. TRIGA Integrated in a Spacecraft C&DH Subsystem

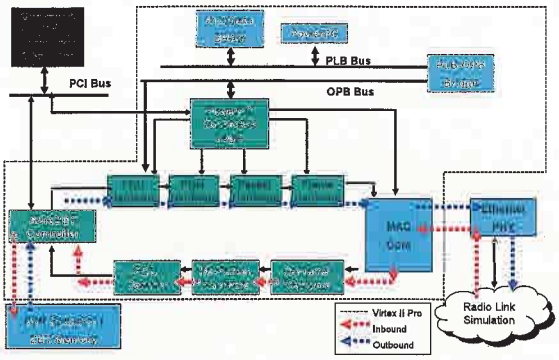


Figure 3. TRIGA System Bock Diagram

register interface allows the C&DH to have control and status access to TRIGA. To transfer a file, the C&DH system is only required to write four register that includes the CFDP Transaction ID, the starting address of the file to be transferred, length of the file and the type of transfer to perform. At the end of the transaction, the C&DH system is notified with the transaction status information.

By interfacing directly to the data sources such as the Solid State Recorder, TRIGA accomplishes zero-copy[3] inbound and outbound functions via the pipelined CCSDS hardware protocol stack. The PowerPC processor on TRIGA only controls and monitors the protocol stack hardware modules and never touches on the high-speed inband data. (Figure 3) Hence, the processing speed of the PowerPC is not critical to the high-performance CFDP protocol processing. In addition, the PowerPC processor can inject data into the pipeline for sending CFDP control packets such as ACK, NAK, EOF and FIN packets. [1] Engineering telemetry can also be inserted on the outbound path using a special Virtual Channel identifier to the Frame Hardware. Likewise on the inbound path, critical hardware telecommands can be directly dispatched from the Deframer hardware to the avionics system.

In contrast, a typical spacecraft that implements CCSDS protocol stack such as the Deep Impact Flyby, CFDP protocol processing core, channel prioritization and switching are done in software while CCSDS Link Layer and coding is done on a dedicated Command and Telemetry Board (CTB). (See Figure 1) This approach results in one or more data copies made between the original SSR storage and the digital modulation interface to be transmitted on the radio during the process of transmitting or reception of data from external entities to the spacecraft. For example, when CFDP packages segments of file data units into protocol data units

[3]We define zero-copy data movement as raw payload data files' bits being moved between the SSR and the digital radio modulation/demodulation unit to perform physical layer functions.

(PDU), the product gets stored in either the shared memory space or the SSR for the framing entity. Depending on whether the Datalink layer framing (TM/TC) is done in software or hardware, the resulting frames will be stored in memory until the opportunity and priority to transmit them arrive. The data copies that result from this approach increases the memory space requirement to run CFDP and also adds latency to the overall file transfer process.

4. System Integration

An example system integration for TRIGA in a spacecraft described in Figure 2 shows how the main data path from the SSR to the radio communication module will be bypassed from the Spacecraft Control Unit (SCU). Instead, the CPU intensive data movements in and out of the spacecraft will become TRIGA's responsibility. As a result, the SCU will accomplish the high data rate communication virtually at no cost.

Since TRIGA can support very high throughput, it is possible to put TRIGA on a gateway spacecraft that is equipped with multiple antennae and multiple SSRs. Acting as a long haul traffic convergence point for assets such as landers, other orbiters and astronauts, the gateway spacecraft can uses TRIGA to manage large number of CFDP transactions at high data rates. (Figure 4) In addition, the reconfigurability of TRIGA can serve to future near-earth missions such as deploying a fleet of landers, roaming astronauts and orbiters for exploring the moon. Since these missions will take advantage of IP-based networking solutions and use a different combination of protocols such as IP and HDLC on top of CCSDS Data Link Layer, TRIGA can easily provide the framework for quicker design and integration.

5. SEU Mitigation

Studies have shown that total dose radiation tolerance for current generation FPGAs such as the Xilinx Virtex-II Pro rated to tolerate 200 krad are acceptable for the majority of space applications.[9][10][11] The epitaxial layer on these radiation-tolerant parts make them essentially immune to Signel Event Latch-up. The problem remains in the area of Single Event Upset (SEU), which cause localized on-chip transient errors. Although SEU hardened devices are available, they are well below the current state-of-the-art SRAM-based FPGAs in terms of features, size and speed grade. The SEU in SRAM-based Virtex-II Pro FPGAs can be mitigated by using a combination of a processor lock-step-and-compare voting scheme using both PPC 405 processors, error detection and correction (EDAC) circuitry, watchdog timers and triple mode redundancy (TMR) logic. The degree of fault tolerance needed for

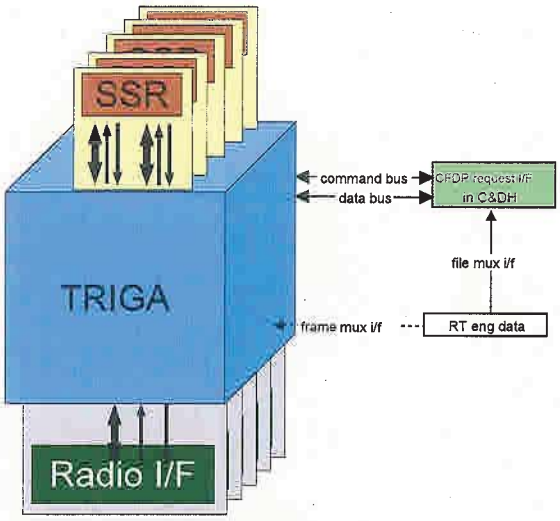


Figure 4. TRIGA use on a Relay Spacecraft where Many File Transfers Converge

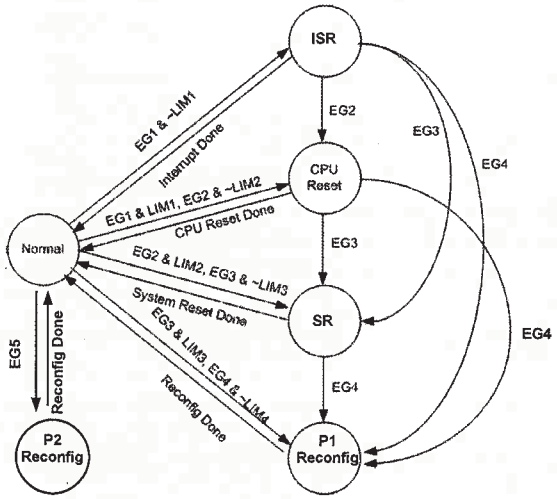
a particular design is determined by the system level requirements that define the level of time-criticality of the detection and correction of single event upsets.

We divide the TRIGA's SEU mitigation techniques into two partitions, namely the critical and non-critical partitions. The critical partition consists of the Telemetry and Telecommand layers due to their direct connection with flight hardware status and command processing.[12] The non-critical portion covers the layers above the data link layer where the error in the transmission is corrected by the CFDP's automatic repeat request procedures. The hardware components that fall under the critical partition are implemented using full TMR logic redundancy as opposed to the non-critical partition that only use selective TMR'ed components such as the module registers.

TRIGA's SEU mitigation design includes a set of hardware modules to detect and correct SEU induced errors. The design is a modified version of an existing SEU mitigation mechanism called *smCore*, which can be easily inserted into an FPGA design using the EDK-based design flow. [13] [14] The original *smCore* provides the following features:

- SEU Supervisory FSM *(sm-fsm)*
- Duplicate-and-compare for PPC405 *(sm-dpc)*
- CRC and readback-based configuration memory checker *(sm-cm-checker)*
- Duplicate-and-compare for Block RAM *(sm-dbram)*
- Watchdog timer for deadlocks *(sm-wdt)*
- Built-in fault injection for verification *(sm-fi)*

As the modified supervisory finite state machine that performs initialization and configuration of the SEU mitigation scheme, the TMR'ed *sm-fsm* manages



Acronyms

ISR	Interrupt Service Routine in PPC to handle EG1
SR	PPC System Reset
EG1	Error Group 1: OCM/PLB BRAM single-bit error
EG2	Error Group 2: CPU mismatch, watchdog timer timeout, OCM BRAM double bit error
EG3	Error Group 3: PLB/OPB bus error
EG4	Error Group 4: Configuration memory error in non TM/TC layers, PLB BRAM double-bit error, Configuration memory watchdog timer timeout
EG5	Error Group 5: Configuration memory error in TM/TC layers
LIM1	Repeated EG1 count threshold
LIM2	Repeated EG2 count threshold
LIM3	Repeated EG3 count threshold
LIM4	Repeated EG4 count threshold
P1	Non TM/TC hardware modules
P2	TM/TC hardware modules

Figure 5. SEU Mitigation Supervisory Finite State Machine

the end-to-end SEU tolerance scheme including detection, recovery and notification. The FSM consists of six states: Normal, Interrupt, CPU reset, System reset, Partition 1 and Partition 2 FPGA reconfiguration, representing the mitigation mechanisms in ascending levels of severity. Depending on the current state and the type of error detected, an appropriate recovery mechanism is invoked. (see Figure 5) We also modify *sm-cm-checker* to allow partial reconfiguration for the two SEU mitigation partitions, namely the TM/TC layers and higher layers.

In terms of TRIGA functional implementation, we modify logic that uses Lookup Tables (LUTs) either as small blocks of distributed RAM elements or as dynamically addressable shift registers (SRL) because their SEU induced errors cannot be detected and corrected through configuration memory readback and scrubbing. This deficiency is due to LUT-based RAM and SRL values being manipulated inside the configuration memory and such changes not being recorded in the configuration mask file that act as the reference for the configuration memory verification process. Instead, we only use block RAM for all RAM functions and flip-flops for shift registers. A few off-the-shelf EDK IP cores such as OPB-UART and PLB2OPB bridge requires modification due to their LUT-based RAM and SRL implementation.[15]

TRIGA presently includes both Mission-Critical and Non-Mission-Critical components. Brief interruptions in data transmission due to TRIGA FPGA reset or reconfiguration are allowed only for the higher protocol layers above TM/TC layers. Hence, an external device such as the flight processor or a SEU hardened FPGA is required to do prompt partial reconfiguration as well as scrubbing in addition to *(sm-cm-checker)* module, which oversees the higher layer's functional integrity. This requirment will not be prohibitive in the flight system architecture since we envision that the next generation spacecraft that outfits TRIGA will also have other FPGAs that need similar supervision.

6. Conclusion and Future Work

We presented a CCSDS communcation subsystem implementation on FPGA that will improve many of the CFDP implementation tradeoffs. The flexibility and ease of integration into a spacecraft C&DH subsystem is also described. The TRIGA SEU mitigation design and its rationale were also described so that the next version of TRIGA can be radiation tolerant. At Technology Readiness Level (TRL) 4, TRIGA reached the simulation medium's throughput ceiling of 100 Mb/s Ethernet for unacknowledged CFDP transactions.[4] The interoperability validation with the JPL's CFDP reference implementation in Class 2 acknowledged mode has also been completed. Preliminary simulation results for power consumption indicates that the non-TMR'ed TRIGA design consumes less than one order of magnitude power per megahertz of operating frequency compared to the RAD750 processor based communication subsystem. A more thorough evaluation of TRIGA's performance and other related telecommunication architectures are described in the different study.[16] Future work for TRIGA includes SEU mitigation implementation to expand the fault tolerance at TRL 5. A variant of TRIGA for the Ground System will also be implemented to demonstrate end-to-end high-speed CFDP file transfers.

Acknowledgment

The research and development described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

[4]TRL 4 defines a component tested using development board in a laboratory environemnt as opposed to TRL 5 where the same set of tests were run under relevant environment such as conditions that can induce SEU.

References

[1] *CCSDS File Delivery Protocol*, Blue Book, No. 3, Consulative Committee for Space Data Systems (CCSDS), June 2005. [Online]. Available: http://public.ccsds.org/publications/archive/727x0b3.pdf

[2] *CCSDS Space Packet Protocol*, Blue Book, No. 1, Consulative Committee for Space Data Systems (CCSDS), Sept. 2003. [Online]. Available: http://public.ccsds.org/publications/archive/133x0b1.pdf

[3] *Telecommand Space Data Link Protocol*, Blue Book, No. 1, Consulative Committee for Space Data Systems (CCSDS), Sept. 2003. [Online]. Available: http://public.ccsds.org/publications/archive/232x0b1.pdf

[4] *Telemetry Space Data Link Protocol*, Blue Book, No. 1, Consulative Committee for Space Data Systems (CCSDS), Sept. 2003. [Online]. Available: http://public.ccsds.org/publications/archive/132x0b1.pdf

[5] *Telecommand Synchronization and Channel Coding*, Blue Book, No. 1, Consulative Committee for Space Data Systems (CCSDS), Sept. 2003. [Online]. Available: http://public.ccsds.org/publications/archive/231x0b1.pdf

[6] *Telemetry Synchronization and Channel Coding*, Blue Book, No. 1, Consulative Committee for Space Data Systems (CCSDS), Sept. 2003. [Online]. Available: http://www.ccsds.org/documents/pdf/CCSDS-101.0-B-4.pdf

[7] "Deep impact cfdp," White Paper (JPL D-23954), Jet Propulsion Laboratory, Pasadena, CA, May 2005.

[8] C. J. Krupiarz, B. K. Heggestad, and R. D. Carper, "Flying CFDP on messenger," sep 2004, International Consortium for Telemetry Spectrum.

[9] H. Bogrow, "The continued evolution of re-configurable FPGAs for aerospace and defense strategic applications," in *Proc. of Military and Aerospace Programmable Logic Devices International Conference*, Wasington, D.C., Sept. 2005.

[10] D. Petrick and J. Howard, "Xilinx Virtex-II Pro PowerPC proton test results," in *Proc. of Military and Aerospace Programmable Logic Devices International Conference*, Wasington, D.C., Sept. 2005.

[11] K. A. LaBel, "NASA FPGA needs and activities," in *Proc. of Military and Aerospace Programmable Logic Devices International Conference*, Wasington, D.C., Sept. 2004.

[12] S. Chau, A. Ko, and K.-M. Cheung, "Mission operation for reconfigurable spacecraft," in *Proceedings of SpaceOps 2004*, Montreal, Canada, May 2004.

[13] Y. He, "smCore - Reusable SEU Mitigation Technology for FPGA-Based Avionics Design and Implementation Document," Jet Propulsion Laboratory, Pasadena, CA, JPL RTD Document Draft 1.1, May 2005.

[14] G. Bolotin, R. K. Watson, R. Petras, and M. Wang, "Advanced mobility avionics: A reconfigurable micro-avionics platform for the future needs of small planetary rovers and microspacecraft," in *Proc. Space 2004 Conference and Exhibit*, San Diego, CA, Sept. 2004, pp. AIAA–2004–5997.

[15] R. Jain, K. K. Ramakrishnan, and D. M. Chiu, "Congestion avoidance in computer networks with a connectionless network layer," Xilinx, Inc, San Jose, CA, Tech. Rep. XAPP197 (v1.0), Nov. 2001.

[16] J. Pang, A. Liddicoat, J. Ralston, P. Pingree, and J. L. Torgerson, "Evaluation of TRIGA," in *Proc. of Coolchips IX*, Yokohama, Japan.

TRIGA: Telecommunications Protocol Processing Subsystem Using Reconfigurable Interoperable Gate Arrays

Jackson Pang, Paula J. Pingree, J. Leigh Torgerson
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109
{Jackson.Pang,Paula.J.Pingree,Jordan.L.Torgerson}@jpl.nasa.gov

Abstract

We present the Telecommunications protocol processing subsystem using Reconfigurable Interoperable Gate Arrays (TRIGA), a novel approach that unifies fault tolerance, error correction coding and interplanetary communication protocol offloading to implement CCSDS File Delivery Protocol and Datalink layers. The new reconfigurable architecture offers more than one order of magnitude throughput increase while reducing footprint requirements in memory, command and data handling processor utilization, communication system interconnects and power consumption.

1. Introduction

The vision of a manned mission to the Moon, Mars and continued space exploration within our solar system has created a clear need for an interplanetary communications network consisting of telecommunications relay satellites, spacecraft, deep space probes, and planetary rovers. The evolution of spacecraft science instrumentation and the need for higher telecommunication bandwidths are placing an increased burden on the flight processors aboard NASA spacecraft. With these new telecommunication system requirements, the need for a high-speed, reliable, and efficient communication protocol has become apparent. An international committee of space agencies known as the Consultative Committee for Space Data Systems (CCSDS) developed the CCSDS File Delivery Protocol (CFDP), which provides reliable communication in high-delay networks along with the CCSDS packet and data link layers. However, with the increased complexity of this protocol stack came an increased burden on the flight processor, taking valuable processing cycles away from flight control software and data processing tasks.

The advent of the reconfigurable hardware and its use in NASA's Deep Space Missions such as the twin Mars Exploration Rovers and the Mars Reconnaissance Orbiter's Electra Software Defined Radio has provided us valuable experience in using SRAM-based FPGAs in deep space environment. The increasing capabilities and the performance benefits that FPGAs offer allow us to expand their use in avionics and telecommunications. Although the SRAM-based FPGAs are more susceptible Single Event Upset episodes compared to the flash-based FPGAs that offer less logic gates and features, both JPL and the aerospace industry has teamed up to form the Single-Event Effects Consortium to study the SEU mitigation techniques such as the configuration memory scrubbing and Triple Module Redundancy(TMR) to extend the fault tolerance of SRAM-based FPGAs. In this paper, we present the Telecommunications protocol processing subsystem using Reconfigurable Interoperable Gate Arrays (TRIGA) that unifies the fault tolerance, error correction coding, CCSDS Datalink Layers and File Delivery Protocol Processing modules in the spacecraft communication subsystem into a single reconfigurable module that is fully interoperable with other CCSDS CFDP reference implementations. This new architecture offers more than one order of magnitude throughput increase while reducing footprint requirements in memory, processor utilization, communications system interconnects and power consumption. The remaining sections of the paper is organized as follows.

In Section 2, we briefly describe the CCSDS file delivery protocol stack and the processing burden that it imposes on the Command and Data Handling (C&DH) subsystem that motivated us to develop TRIGA. Section 3 explains the TRIGA architecture and the advantages it offers over the traditional CFDP software processing. In addition, the next-generation system architectures that incorporate TRIGA are presented in sections 4. Single Even Upset (SEU) mitigation solutions for TRIGA is described in Section 5. Section 6 summarizes our contributions and describes our future work.

2. Motivation

Although CCSDS protocol stack (Table 1) and its user interface procedures are standardized with a

Table 1. CCSDS Protocol Stack Functions

OSI Layers	CCSDS Layers	CCSDS Protocol	
		Uplink to Spacecraft	Downlink from Spacecraft
Application	Application	CFDP [1]	CFDP
Transport	Transport		
Network	Logical Data Path (LDP)	Space Packet [2]	Space Packet
Data Link	Data Link Protocol Sublayer	Telecommand Space Data Link Protocol [3]	Telemetry Space Data Link Protocol [4]
	Channel Coding Sublayer	Telecommand Synchro-nization and Channel Coding [5]	Telemetry Syn-chronization and Channel Coding [6]
Physical	Physical	UHF/KA-Band/S-Band	UHF/KA-Band/S-Band

vision to create seamless space communication, its implementation opens up many issues regarding the resource and processing requirement on the spacecraft C&DH computer. To date, two NASA Discovery missions, namely Deep Impact from JPL/NASA and MESSENGER from APL/JHU[1], have implemented a variant of CFDP as part of their spacecraft communication subsystem, but both implementations were forced to strip out many features of CFDP to make it reasonable for the available spacecraft resources. In addition, auxiliary hardware devices were also required to process the lower layers below CFDP transport layer of the CCSDS protocol stack.

2.1. Deep Impact CFDP Usage

In January 2005, Deep Impact mission launched the two joined spacecraft Flyby and Impactor. Six months later, the Impactor module separated and collided into comet Temple 1 while the Flyby module gathered the secrets of our universe's origin by examining the impact crater and ejecta using the high resolution imaging and infrared spectroscopy. Earlier in the mission phase for flight software requirement review, the deliberation was made to use CFDP for its wealth of features and for being the international CCSDS standard. The Deep Impact software implementation of CFDP is slightly modified in that the Earth to Flyby is in acknowledged mode, between Flyby and Impactor in unacknowledged mode, and both Flyby to Earth is done in unacknowledged mode.[2] On both the Impactor and the Flyby, CCSDS Datalink layer functions (i.e. framing, channel coding and synchronization) are accomplished in a hardware device called the Command and Telemetry Board (CTB). (See Figure 1) [7]

[1] Applied Physics Laboratory at the Johns Hopkins University
[2] For brevity, this usage of CFDP is herein referred to as DI-CFDP.

2.2. MESSENGER and CFDP

Launched in the summer of 2004, MESSENGER (MErcury Surface, Space ENvironment, GEochemistry and Ranging) spacecraft is headed to Mercury for a one-year science mission starting in 2011. To compliment the newly augmented file management system and the large data volume that the spacecraft will generate throughout the course of its mission, CFDP was selected as the file transfer mechanism. However, the shared processing resources available onboard (25MHz RAD7000 PPC CPU, 8MB RAM, 4MB EEPROM) for CFDP, Guidance and Control, and image compression made it prohibitive to use the full CFDP reference implementation maintained by JPL. Consequently, the customized version of APL CFDP called CFDP-lite was engineered to couple the CFDP closely with the constrained memory availability and other spacecraft file management tasks. In CFDP-lite, APL successfully implemented the unacknowledged downlink portion of software CFDP. Similarly on MESSENGER, a hardware module was used for the Data Link and Coding tasks. [8] This type of CFDP usage has been implemented on the Mars Reconnaissance Orbiter and future missions such as Mars Science Laboratory and JUNO.

2.3. CCSDS Link, Channel and Coding Layers

CCSDS also defines a set of lower protocol layers, namely Space Packet, Telecommand and Telemetry layers, to create the protocol stack that supports deep space telecommunication needs. From the processing point of view, these layers require two types of services, namely packetizing a frame with a given header and computationally intensive tasks such as Cyclic Redundancy Coding, Reed-Solomon and Convolutional Coding. Due to the inherently independent and parallel nature of these requirements, their functionalities were implemented in dedicated hardware for efficiency.

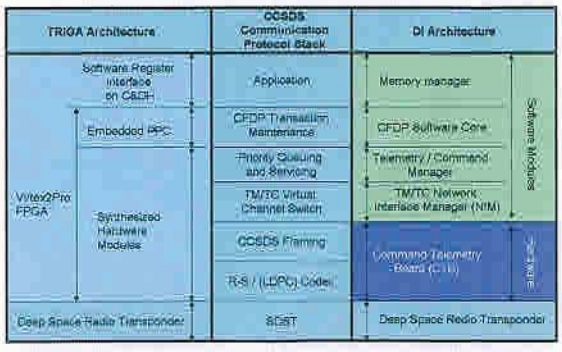


Figure 1. Deep Impact Communication Subsystem Components Redefined

By reflecting on the current implementation strategies of the CCSDS protocol stack, we can deduce that a unified communication protocol processing subsystem can alleviate many of the compromises we face regarding how we should tailor CFDP for a specific mission. CFDP as a complete file delivery protocol is designed well, but the approaches such as DI-CFDP and CFDP-lite removes CFDP's salient features such as multihop and automatic repeat request capabilities. Hence, CFDP cannot shine as a reliable transport protocol but merely as a remote file organizing scheme, which may hinder the wide deployment of CFDP due to misunderstanding of the potential benefits of CFDP. In addition, the recurring engineering and risk assessment costs related to producing a CFDP variant is driving the mission expenditure up.

At the system level, the requirement of dedicated hardware such as the Command and Telemetry Board used on Deep Impact for lower layer protocol processing adds design complexity, power consumption and real estate requirements, all of which are primary considerations made in the design and cost estimation of next generation spacecraft. An architecture that decouples file transfer functions from the rest of the C&DH system while integrating the full functionality of the file delivery protocol stack into one subsystem is highly desirable. Hence, we engineered TRIGA to provide the features of a reliable communication subsystem including the solutions to mitigate the radiation induced Single Event Upset scenarios discussed in Section 5.

3. The Zero-copy TRIGA Architecture

The TRIGA architecture is designed to support high throughput with minimal support from the C&DH system. At the very high level, TRIGA look much like a direct memory access engine. As shown in Figure 2, TRIGA interfaces directly to the data storage and the radio interface using high speed data channels. The

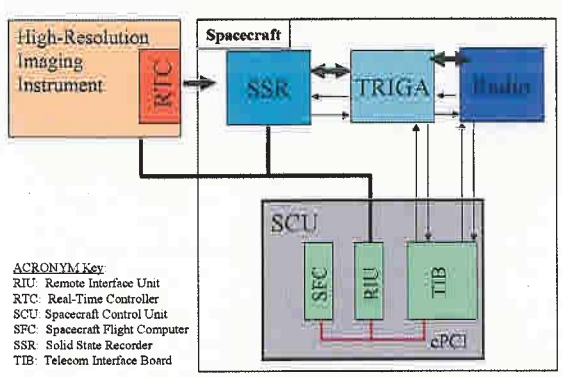


Figure 2. TRIGA Integrated in a Spacecraft C&DH Subsystem

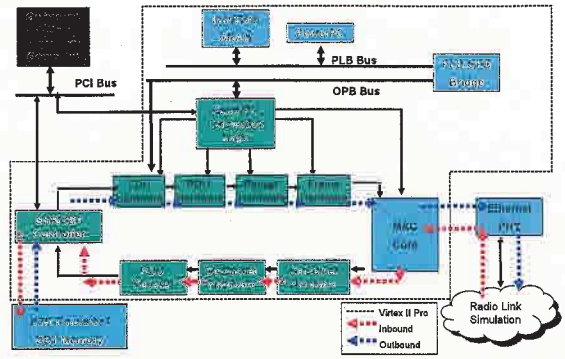


Figure 3. TRIGA System Bock Diagram

register interface allows the C&DH to have control and status access to TRIGA. To transfer a file, the C&DH system is only required to write four register that includes the CFDP Transaction ID, the starting address of the file to be transferred, length of the file and the type of transfer to perform. At the end of the transaction, the C&DH system is notified with the transaction status information.

By interfacing directly to the data sources such as the Solid State Recorder, TRIGA accomplishes zero-copy[3] inbound and outbound functions via the pipelined CCSDS hardware protocol stack. The PowerPC processor on TRIGA only controls and monitors the protocol stack hardware modules and never touches on the high-speed inband data. (Figure 3) Hence, the processing speed of the PowerPC is not critical to the high-performance CFDP protocol processing. In addition, the PowerPC processor can inject data into the pipeline for sending CFDP control packets such as ACK, NAK, EOF and FIN packets. [1] Engineering telemetry can also be inserted on the outbound path using a special Virtual Channel identifier to the Frame Hardware. Likewise on the inbound path, critical hardware telecommands can be directly dispatched from the Deframer hardware to the avionics system.

In contrast, a typical spacecraft that implements CCSDS protocol stack such as the Deep Impact Flyby, CFDP protocol processing core, channel prioritization and switching are done in software while CCSDS Link Layer and coding is done on a dedicated Command and Telemetry Board (CTB). (See Figure 1) This approach results in one or more data copies made between the original SSR storage and the digital modulation interface to be transmitted on the radio during the process of transmitting or reception of data from external entities to the spacecraft. For example, when CFDP packages segments of file data units into protocol data units

[3]We define zero-copy data movement as raw payload data files' bits being moved between the SSR and the digital radio modulation/demodulation unit to perform physical layer functions.

(PDU), the product gets stored in either the shared memory space or the SSR for the framing entity. Depending on whether the Datalink layer framing (TM/TC) is done in software or hardware, the resulting frames will be stored in memory until the opportunity and priority to transmit them arrive. The data copies that result from this approach increases the memory space requirement to run CFDP and also adds latency to the overall file transfer process.

4. System Integration

An example system integration for TRIGA in a spacecraft described in Figure 2 shows how the main data path from the SSR to the radio communication module will be bypassed from the Spacecraft Control Unit (SCU). Instead, the CPU intensive data movements in and out of the spacecraft will become TRIGA's responsibility. As a result, the SCU will accomplish the high data rate communication virtually at no cost.

Since TRIGA can support very high throughput, it is possible to put TRIGA on a gateway spacecraft that is equipped with multiple antennae and multiple SSRs. Acting as a long haul traffic convergence point for assets such as landers, other orbiters and astronauts, the gateway spacecraft can uses TRIGA to manage large number of CFDP transactions at high data rates. (Figure 4) In addition, the reconfigurability of TRIGA can serve to future near-earth missions such as deploying a fleet of landers, roaming astronauts and orbiters for exploring the moon. Since these missions will take advantage of IP-based networking solutions and use a different combination of protocols such as IP and HDLC on top of CCSDS Data Link Layer, TRIGA can easily provide the framework for quicker design and integration.

5. SEU Mitigation

Studies have shown that total dose radiation tolerance for current generation FPGAs such as the Xilinx Virtex-II Pro rated to tolerate 200 krad are acceptable for the majority of space applications.[9][10][11] The epitaxial layer on these radiation-tolerant parts make them essentially immune to Signel Event Latch-up. The problem remains in the area of Single Event Upset (SEU), which cause localized on-chip transient errors. Although SEU hardened devices are available, they are well below the current state-of-the-art SRAM-based FPGAs in terms of features, size and speed grade. The SEU in SRAM-based Virtex-II Pro FPGAs can be mitigated by using a combination of a processor lock-step-and-compare voting scheme using both PPC 405 processors, error detection and correction (EDAC) circuitry, watchdog timers and triple mode redundancy (TMR) logic. The degree of fault tolerance needed for

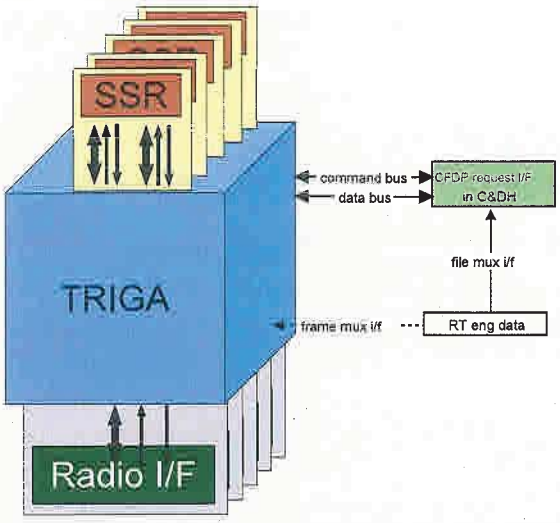


Figure 4. TRIGA use on a Relay Spacecraft where Many File Transfers Converge

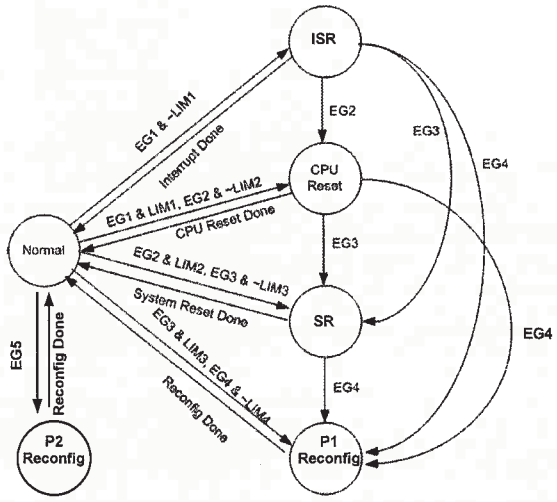
a particular design is determined by the system level requirements that define the level of time-criticality of the detection and correction of single event upsets.

We divide the TRIGA's SEU mitigation techniques into two partitions, namely the critical and non-critical partitions. The critical partition consists of the Telemetry and Telecommand layers due to their direct connection with flight hardware status and command processing.[12] The non-critical portion covers the layers above the data link layer where the error in the transmission is corrected by the CFDP's automatic repeat request procedures. The hardware components that fall under the critical partition are implemented using full TMR logic redundancy as opposed to the non-critical partition that only use selective TMR'ed components such as the module registers.

TRIGA's SEU mitigation design includes a set of hardware modules to detect and correct SEU induced errors. The design is a modified version of an existing SEU mitigation mechanism called *smCore*, which can be easily inserted into an FPGA design using the EDK-based design flow. [13] [14] The original *smCore* provides the following features:

- SEU Supervisory FSM *(sm-fsm)*
- Duplicate-and-compare for PPC405 *(sm-dpc)*
- CRC and readback-based configuration memory checker *(sm-cm-checker)*
- Duplicate-and-compare for Block RAM *(sm-dbram)*
- Watchdog timer for deadlocks *(sm-wdt)*
- Built-in fault injection for verification *(sm-fi)*

As the modified supervisory finite state machine that performs initialization and configuration of the SEU mitigation scheme, the TMR'ed *sm-fsm* manages



Acronyms

ISR	Interrupt Service Routine in PPC to handle EG1
SR	PPC System Reset
EG1	Error Group 1: OCM/PLB BRAM single-bit error
EG2	Error Group 2: CPU mismatch, watchdog timer timeout, OCM BRAM double bit error
EG3	Error Group 3: PLB/OPB bus error
EG4	Error Group 4: Configuration memory error in non TM/TC layers, PLB BRAM double-bit error, Configuration memory watchdog timer timeout
EG5	Error Group 5: Configuration memory error in TM/TC layers
LIM1	Repeated EG1 count threshold
LIM2	Repeated EG2 count threshold
LIM3	Repeated EG3 count threshold
LIM4	Repeated EG4 count threshold
P1	Non TM/TC hardware modules
P2	TM/TC hardware modules

Figure 5. SEU Mitigation Supervisory Finite State Machine

the end-to-end SEU tolerance scheme including detection, recovery and notification. The FSM consists of six states: Normal, Interrupt, CPU reset, System reset, Partition 1 and Partition 2 FPGA reconfiguration, representing the mitigation mechanisms in ascending levels of severity. Depending on the current state and the type of error detected, an appropriate recovery mechanism is invoked. (see Figure 5) We also modify *sm-cm-checker* to allow partial reconfiguration for the two SEU mitigation partitions, namely the TM/TC layers and higher layers.

In terms of TRIGA functional implementation, we modify logic that uses Lookup Tables (LUTs) either as small blocks of distributed RAM elements or as dynamically addressable shift registers (SRL) because their SEU induced errors cannot be detected and corrected through configuration memory readback and scrubbing. This deficiency is due to LUT-based RAM and SRL values being manipulated inside the configuration memory and such changes not being recorded in the configuration mask file that act as the reference for the configuration memory verification process. Instead, we only use block RAM for all RAM functions and flip-flops for shift registers. A few off-the-shelf EDK IP cores such as OPB-UART and PLB2OPB bridge requires modification due to their LUT-based RAM and SRL implementation.[15]

TRIGA presently includes both Mission-Critical and Non-Mission-Critical components. Brief interruptions in data transmission due to TRIGA FPGA reset or reconfiguration are allowed only for the higher protocol layers above TM/TC layers. Hence, an external device such as the flight processor or a SEU hardened FPGA is required to do prompt partial reconfiguration as well as scrubbing in addition to *(sm-cm-checker)* module, which oversees the higher layer's functional integrity. This requirment will not be prohibitive in the flight system architecture since we envision that the next generation spacecraft that outfits TRIGA will also have other FPGAs that need similar supervision.

6. Conclusion and Future Work

We presented a CCSDS communcation subsystem implementation on FPGA that will improve many of the CFDP implementation tradeoffs. The flexibility and ease of integration into a spacecraft C&DH subsystem is also described. The TRIGA SEU mitigation design and its rationale were also described so that the next version of TRIGA can be radiation tolerant. At Technology Readiness Level (TRL) 4, TRIGA reached the simulation medium's throughput ceiling of 100 Mb/s Ethernet for unacknowledged CFDP transactions.[4] The interoperability validation with the JPL's CFDP reference implementation in Class 2 acknowledged mode has also been completed. Preliminary simulation results for power consumption indicates that the non-TMR'ed TRIGA design consumes less than one order of magnitude power per megahertz of operating frequency compared to the RAD750 processor based communication subsystem. A more thorough evaluation of TRIGA's performance and other related telecommunication architectures are described in the different study.[16] Future work for TRIGA includes SEU mitigation implementation to expand the fault tolerance at TRL 5. A variant of TRIGA for the Ground System will also be implemented to demonstrate end-to-end high-speed CFDP file transfers.

Acknowledgment

The research and development described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

[4]TRL 4 defines a component tested using development board in a laboratory environemnt as opposed to TRL 5 where the same set of tests were run under relevant environment such as conditions that can induce SEU.

References

[1] *CCSDS File Delivery Protocol*, Blue Book, No. 3, Consulative Committee for Space Data Systems (CCSDS), June 2005. [Online]. Available: http://public.ccsds.org/publications/archive/727x0b3.pdf

[2] *CCSDS Space Packet Protocol*, Blue Book, No. 1, Consulative Committee for Space Data Systems (CCSDS), Sept. 2003. [Online]. Available: http://public.ccsds.org/publications/archive/133x0b1.pdf

[3] *Telecommand Space Data Link Protocol*, Blue Book, No. 1, Consulative Committee for Space Data Systems (CCSDS), Sept. 2003. [Online]. Available: http://public.ccsds.org/publications/archive/232x0b1.pdf

[4] *Telemetry Space Data Link Protocol*, Blue Book, No. 1, Consulative Committee for Space Data Systems (CCSDS), Sept. 2003. [Online]. Available: http://public.ccsds.org/publications/archive/132x0b1.pdf

[5] *Telecommand Synchronization and Channel Coding*, Blue Book, No. 1, Consulative Committee for Space Data Systems (CCSDS), Sept. 2003. [Online]. Available: http://public.ccsds.org/publications/archive/231x0b1.pdf

[6] *Telemetry Synchronization and Channel Coding*, Blue Book, No. 1, Consulative Committee for Space Data Systems (CCSDS), Sept. 2003. [Online]. Available: http://www.ccsds.org/documents/pdf/CCSDS-101.0-B-4.pdf

[7] "Deep impact cfdp," White Paper (JPL D-23954), Jet Propulsion Laboratory, Pasadena, CA, May 2005.

[8] C. J. Krupiarz, B. K. Heggestad, and R. D. Carper, "Flying CFDP on messenger," sep 2004, International Consortium for Telemetry Spectrum.

[9] H. Bogrow, "The continued evolution of re-configurable FPGAs for aerospace and defense strategic applications," in *Proc. of Military and Aerospace Programmable Logic Devices International Conference*, Wasington, D.C., Sept. 2005.

[10] D. Petrick and J. Howard, "Xilinx Virtex-II Pro PowerPC proton test results," in *Proc. of Military and Aerospace Programmable Logic Devices International Conference*, Wasington, D.C., Sept. 2005.

[11] K. A. LaBel, "NASA FPGA needs and activities," in *Proc. of Military and Aerospace Programmable Logic Devices International Conference*, Wasington, D.C., Sept. 2004.

[12] S. Chau, A. Ko, and K.-M. Cheung, "Mission operation for reconfigurable spacecraft," in *Proceedings of SpaceOps 2004*, Montreal, Canada, May 2004.

[13] Y. He, "smCore - Reusable SEU Mitigation Technology for FPGA-Based Avionics Design and Implementation Document," Jet Propulsion Laboratory, Pasadena, CA, JPL RTD Document Draft 1.1, May 2005.

[14] G. Bolotin, R. K. Watson, R. Petras, and M. Wang, "Advanced mobility avionics: A reconfigurable micro-avionics platform for the future needs of small planetary rovers and microspacecraft," in *Proc. Space 2004 Conference and Exhibit*, San Diego, CA, Sept. 2004, pp. AIAA–2004–5997.

[15] R. Jain, K. K. Ramakrishnan, and D. M. Chiu, "Congestion avoidance in computer networks with a connectionless network layer," Xilinx, Inc, San Jose, CA, Tech. Rep. XAPP197 (v1.0), Nov. 2001.

[16] J. Pang, A. Liddicoat, J. Ralston, P. Pingree, and J. L. Torgerson, "Evaluation of TRIGA," in *Proc. of Coolchips IX*, Yokohama, Japan.