CSP Hybrid Space Computing for STP-H5/ISEM on ISS

Christopher Wilson, Jacob Stewart, Patrick Gauvin, James MacKinnon, James Coole, Jonathan Urriste, Alan George
NSF Center for High-Performance Reconfigurable Computing (CHREC) — University of Florida
327 Larsen Hall, 968 Center Drive, Gainesville, FL, 32611; 352-392-9225
george@chrec.org

Gary Crum, Elizabeth Timmons, Jaclyn Beck, Tom Flatley
NASA Goddard Space Flight Center
8800 Greenbelt Rd, Greenbelt, MD, 20771; 301-286-3713
gary.a.crum@nasa.gov

Mike Wirthlin, Alex Wilson, Aaron Stoddard
NSF Center for High-Performance Reconfigurable Computing (CHREC) — Brigham Young University
459 Clyde Building, Brigham Young University, Provo, UT, 84602; 801-422-7601
wirthlin@ee.byu.edu

ABSTRACT

The Space Test Program (STP) at the Department of Defense (DoD) supports the development, evaluation, and advancement of new technologies needed for the future of spaceflight. STP-Houston provides opportunities for DoD and civilian space agencies to perform on-orbit research and technology demonstrations from the International Space Station (ISS). The STP-H5/ISEM (STP-Houston 5, ISS SpaceCube Experiment Mini) payload is scheduled for launch on the upcoming SpaceX 10 mission and will feature new technologies, including a hybrid space computer developed by the NSF CHREC Center, working closely with the NASA SpaceCube Team, known as the CHREC Space Processor (CSP). In this paper, we present the novel concepts behind CSP and the CSPv1 flight technologies on the ISEM mission. The ISEM-CSP system was subjected to environmental testing, including a thermal vacuum test, a vibration test, and two radiation tests, and results were encouraging and are presented. Primary objectives for ISEM-CSP are highlighted, which include processing, compression, and downlink of terrestrial-scene images for display on Earth, and monitoring of upset rates in various subsystems to provide environmental information for future missions. Secondary objectives are also presented, including experiments with features for fault-tolerant computing, reliable middleware services, FPGA partial reconfiguration, device virtualization, and dynamic synthesis.

I. INTRODUCTION

Validation of new-processing technology is one of the most crucial challenges to the future generation of space missions. Spacecraft technology has developed by stressing the importance of creating highly reliable and more affordable space systems. Prohibitive launch costs and increasing demands for higher-computational performance have promoted a rising trend towards development of smaller CubeSats featuring commercial technology on higher-risk missions and less-stringent standards as exemplified in [1] and [2]. Allowed by these advancements, it is now possible for a group of small satellites to perform the same mission tasks that would have required a costly, massively sized satellite in the past. This concept has been extensively studied in [3]. Commercial technology has the capability to provide vast increases in performance over traditional radiation-hardened devices, however, many of these devices are highly susceptible to radiation effects [4].

Much of the progress and direction of spacecraft technology can be attributed to the NASA's response to the National Research Council's (NRC) decadal survey for Earth science [5]. The decadal survey focuses on the needs and priorities of the scientific community to plan on key space-research areas and missions. In the midterm assessment of the original 2007 survey, there are two key findings that can be addressed with new-processing capability for Small Satellites.

> *"The nation's Earth observing system is beginning a rapid decline in capability as long-running missions end and key new missions are delayed, lost, or canceled."*

This first finding focuses on large, government-funded satellite missions and illustrates a problematic scenario for future Earth observation as the number of planned and funded missions decrease (Figure 1). In addition,

Earth science needs more data to sustain more powerful climate and weather models. The solution to the decrease in Earth-observing missions is to develop more small satellites, with high-processing capability. One major concern is with next-generation instruments that are capable of generating immense amount of data, a satellite may saturate its downlink bandwidth, therefore, to alleviate this issue, many of the algorithms can be performed with on-board processing so that only results need to be transmitted.

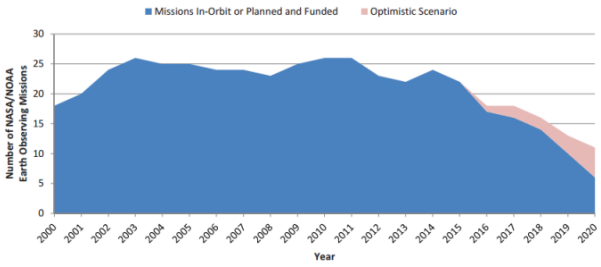

Figure 1: NASA/NOAA Earth-Observing Missions

Another major finding of the decadal survey:

> *"Alternative platforms and flight formations offer programmatic flexibility. In some cases, they may be employed to lower the cost of meeting science objectives and/or maturing remote sensing and in situ observing technologies."*

The alternative platforms mentioned in the survey include small satellites that can either act independently or work cooperatively to form a distributed science mission. To address these two findings of the decadal survey, this paper describes the CHREC Space Processor (CSP) in mission configuration on the International Space Station (ISS) for the Space Test Program - Houston 5 ISS SpaceCube Experiment Mini (STP-H5/ISEM).

The STP-H5 ISEM mission will provide validation for the new hybrid-processing technology and experimental research studied by the National Science Foundation (NSF) Center for High-Performance and Reconfigurable Computing (CHREC). The results of this mission will prove the technology is ready for future small-satellite missions to help solve the issues presented in the decadal survey. The ISEM-CSP flight box, along with the rest of the STP-H5 pallet, will be attached to the International Space Station (ISS), which provides a stable, long-lasting mission platform. Once the STP-H5 mission is launched and integrated onto the Express Logics Carrier (ELC) aboard the ISS, the CSP flight unit will be a continuous development platform for software testing, because new applications, design cores, and upgrades can be uploaded and tested on board. This mission is a first step for the demonstration of the technology and will prove that the CSPv1 space computer will support both high performance and high reliability for future small-satellite missions.

In this paper, we describe the mission objectives and configurations of the first version of our multifaceted-hybrid computer, CSPv1. The organization of the remainder of the paper is as follows. In Section II, we give a background of the enabling programs that allowed this mission to continue. Section III describes the overall CSP concept and the general features and fault-tolerant options of the CSPv1 flight board. In Section IV, we present environmental-testing results that have been performed to verify behavior of the flight box before launch. Section V describes the configuration of the flight hardware and software, in addition to the supporting software for the ground station. In Section VI, we describe the primary mission objectives. Section VII highlights novel research to be conducted as secondary mission objectives. Finally, Section VIII provides concluding remarks.

II. BACKGROUND

This section provides a cursory overview of the programs that have supported the upcoming ISEM-CSP launch. The launch opportunity is provided by the Space Test Program Houston office and the CSP flight box has been included on the ISEM experiment stack as a secondary science and technology payload.

Space Test Program (STP) Houston

The Space Test Program serves the Department of Defense (DoD) and its space science and technology community as the main provider of spaceflight. Officially, it is chartered by the Office of the Secretary of Defense to serve as:

> *"...the primary provider of mission design, spacecraft acquisition, integration, launch, and on-orbit operations for DOD's most innovative space experiments, technologies and demonstrations"*

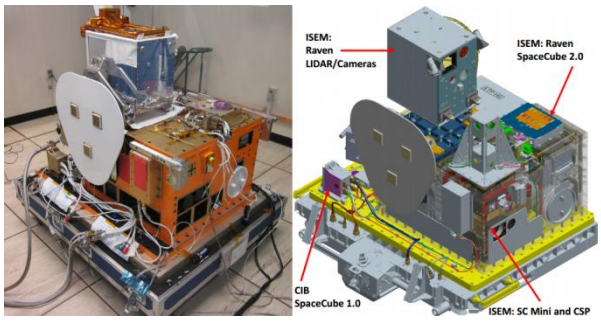

Figure 2: STP-H5 Pallet Layout, Integration Photo (Left), 3D Model (Right)

Formed in 1965, the Space Test Program has been providing access to space for the DoD development community, and it is responsible for many of the military satellite programs flying today including the Global Positioning System [6].

The Space Test Program Houston office is the sole interface to NASA for all DoD payloads on the International Space Station (ISS), and other human-rated launch vehicles, both domestic and international. The office's main goal is to provide timely spaceflight, assure the payload is ready for flight, and to provide management and technical support for the safety and integration processes [7].

The latest launch for STP is STP-H5, which will feature several payload experiments, ISEM-CSP included. STP-H5 is expected to launch early in 2016. The entire mission pallet is featured in Figure 2.

ISS SpaceCube Experiment Mini (ISEM)

NASA Goddard's Science Data Processing Branch and SpaceCube Team have included the ISEM-CSP flight box as a secondary module in the SpaceCube Mini Experiment on STP-H5. The NASA Goddard SpaceCube team is at the forefront of advanced-avionics solutions for space missions. Their primary goal is to enable new classes of future missions by developing new technology for small-spacecraft architectures, mission concepts, component-subsystem hardware, and deployment methods.

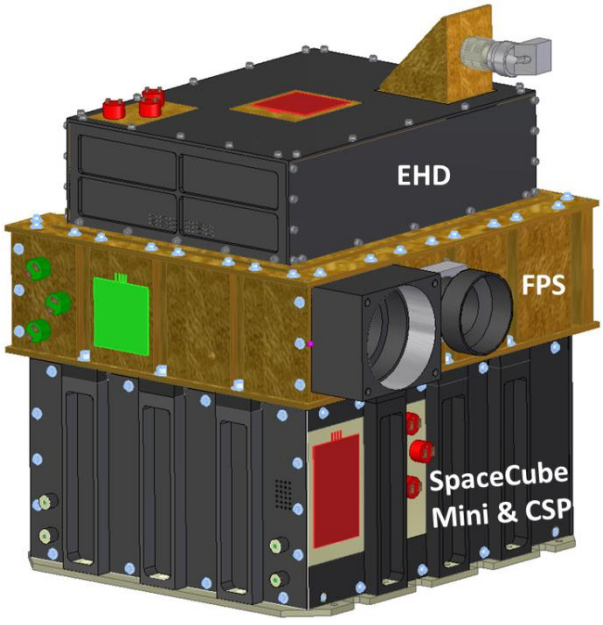

Figure 3: ISEM Experiment Stack

One of the most recognizable contributions that the branch has made to space development is the successful design and launch of SpaceCube, a family of high-performance reconfigurable systems, which has also inspired several design aspects of the CSPv1. SpaceCube has been featured as the prominent technology on several missions including the Hubble Servicing Mission 4, MISSE-7, STP-H4, and STP-H5/RAVEN, with more scheduled. The ISEM experiment on STP-H5 focuses on SpaceCube Mini, which serves as primary communication bus for some of the DoD payloads as well as the CHREC Space Processor. The SpaceCube Mini was designed as a near functional equivalent to the SpaceCube 2.0, but in a 1U CubeSat form factor. The STP-H5 mission is the first flight of the SpaceCube Mini and serves to increase the Technology Readiness Level (TRL) [8][9] of the design. The ISEM Experiment Stack is depicted in Figure 3, and also displays the Electro-Hydro Dynamic (EHD) thermal fluid pump experiment, and the Fabry-Perot Spectrometer (FPS) for atmospheric methane. Lastly, ISEM also includes the Innovative Coatings Experiment (ICE, not pictured) that intends to evaluate next-generation thermal coatings.

III. CHREC SPACE PROCESSOR (CSP)

The CSP system is a multifaceted, hybrid space computer developed by researchers in CHREC at the University of Florida and Brigham Young University, working closely with NASA Goddard Space Flight Center. This section will give an overview of the concept behind CSP as well as a discussion of features of the first flight model, CSPv1, and its fault-tolerant capabilities.

CSP: The Concept

CSP is a concept for a multifaceted, hybrid processing system. This concept centers on having both a hybrid-processor and hybrid-system architecture. The hybrid processor of the CSP is the Xilinx Zynq System on Chip (SoC). Having a processor device with mixed technology can provide immense computational benefits depending on an algorithm's structure. For example, with a mixed FPGA+CPU combination, a parallel algorithm can be hardware-accelerated on the FPGA fabric, while control-flow operations can be performed on the CPU cores. An SoC allows the users to optimize an algorithm to the specialized resources of the device.

The CSP concept also features a hybrid-system architecture, which is a combination of three themes: commercial-off-the-shelf (COTS) devices; radiation-hardened devices; and fault-tolerant computing strategies. Commercial devices have the energy and

performance benefits of the latest technology but are susceptible to radiation in space, whereas radiation-hardened devices are relatively immune to radiation but are more expensive, larger, and outdated in both speed and functionality. The keystone principle of the CSP concept is to have a device with commercial technology featured, for the best in high performance and energy efficiency, but supported by radiation-hardened devices monitoring and managing the COTS devices, and further augmented by fault-tolerant computing strategies. This concept is illustrated in Figure 4.

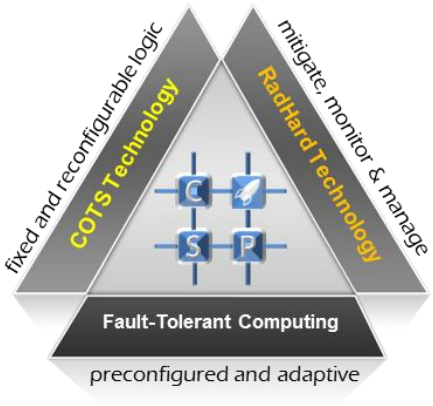

Figure 4: CSP Concept

CSPv1 Components and Features

CSPv1 is the first flight board evolved from the CSP concept and features a hybrid-processor and hybrid-system architecture. The processor architecture features fixed (dual ARM Cortex-A9/NEON cores) and reconfigurable (28 nm Artix-7 FPGA fabric) logic on the Xilinx Zynq-7020 device. The system architecture combines commercial and radiation-hardened electronics with techniques in fault-tolerant computing to achieve a system with a powerful combination of high speed and reliability with low power, size, weight, and cost.

The CSPv1 is designed to fit a 1U CubeSat form factor (10 cm × 10 cm). The design uniquely supports the ability to selectively populate several components with a radiation-hardened or commercial version on the same printed circuit board (PCB) design. This feature provides a spectrum of combinations to scale cost and reliability for different requirements. Figure 5 shows a populated board using all-COTS components, where the unpopulated regions are for the placement of equivalent radiation-hardened components.

All external connections to the CSPv1 board are made through a 160-pin Samtec Searay connector. There are 60 high-speed connections from the FPGA portion of the Zynq, where 48 pins can be configured as 24 differential pairs for high-speed interfaces. There are also 26 high-speed connections from the ARM portion of the Zynq that can be configured in a combination of varying communication interfaces including UART and I^2C.

The major subsystems of CSPv1 are fully detailed in [10] but as a quick summary include the Xilinx Zynq, 512 MB of DDR3 memory (supporting up to 1 GB), NAND flash memory (1-4 GB), watchdog controller, reset circuit, and power circuit. These units are supported by a lightweight Linux kernel, named Wumbo, and a variety of software and hardware services.

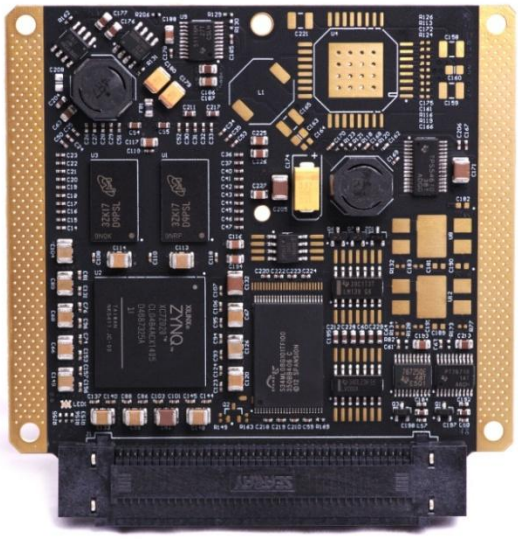

Figure 5: Front Side of COTS CSPv1 Board

CSPv1 Fault Tolerance

The CSPv1 hardware architecture was designed to support multiple levels of fault tolerance. The components used in each subsystem were selected to perform in harsh environments. If any major subsystem of the CSPv1 fails, the CSPv1 would become unusable, therefore, most of the subsystems either contain the inherent ability to recover from radiation effects or else have a radiation-hardened variant that can be populated on the board.

The Zynq has three internal watchdogs which can be used to detect and correct system faults. Additionally, an external supervisor circuit with hardware watchdog was integrated into the CSPv1 to monitor the processing device for radiation upsets that the processor is not able to mitigate internally.

The ARM side of the Zynq is connected to the non-volatile memory and is responsible for configuring the system, including the FPGA, on boot. As a precaution, for the critical booting process, CSPv1 repurposes built-in RSA authentication features of the Zynq to check boot images before startup. As an additional safety measure, multiple boot images can be stored in the non-volatile memory to be used as a fallback.

Once the boot image is verified and the device is booted, the CSPv1 runs a custom, lightweight Linux-based operating system (Wumbo). The Linux kernel represents a large portion of the software running on CSPv1. Optional improvements and modifications to the Linux kernel can be made to increase fault tolerance including: disabling the caches, enabling ECC on the DRAM, and reporting parity faults on the caches if enabled. Fault detection within the kernel is also improved with the addition of rebooting on kernel panics, soft and hard lockup detection, and the Error Detection and Correction (EDAC) module. Together, these improvements provide higher reliability, longer average system up-time, and more detailed system reports on upset events.

One of the main challenges for incorporating an FPGA device in a spacecraft system stems from the SRAM-based memory architecture, which makes it susceptible to Single-Event Effects (SEEs). These events, which are a common occurrence in a harsh-radiation environment, can manifest as bit flips in configuration or data memory, which can eventually lead to device failure. One solution to these issues is a technique known as configuration scrubbing, the process of quickly repairing these configuration-bit upsets in the FPGA before they accumulate and lead to a failure. CSPv1 features a readback scrubber with a variable scrub rate and detailed error messages. This scrubber periodically reads back the entire configuration memory and performs writes to configuration frames that correct configuration memory without disturbing other dynamic portions of memory. A new and more efficient hybrid scrubber that takes advantage of both built-in, single-bit correction and ECC will be integrated into CSPv1 in the coming months. This improved scrubber will reduce overhead significantly and improve error-correction latency.

IV. ENVIRONMENTAL TESTING

The CSP flight box on STP-H5 has undergone environmental testing with the rest of ISEM and will undergo additional testing during full integration of the STP-H5 pallet. Moreover, the CSPv1 board has undergone radiation testing at two different testing facilities.

Thermal, Vacuum, & Vibration Tests

Prior to delivery for integration with the full STP-H5 payload, ISEM was required to undergo a workmanship level *Random Vibration Test* and a *Thermal Cycle Test*. The Random Vibration Test is performed to identify latent defects and manufacturing flaws in electrical, electronic, and electromechanical hardware at the component level. The Thermal Cycle Test is performed to confirm expected performance of a device in a temperature range enveloping mission conditions. The minimum workmanship Random Vibration Test levels are listed in Table 1.

Table 1: Random Vibration Test Levels

20 Hz	**@ 0.01 g^2/Hz**
20 to 80 Hz	**@ +3dB/oct**
80 to 500 Hz	**@ 0.04 g^2/Hz**
500 to 2000 Hz	**@ -3dB/oct**
2000 Hz	**@ 0.01 g^2/Hz**
Overall Level	**= 6.8 g_{rms}**

The Random Vibration Test was performed unpowered, with a sine sweep prior to and after each axis. The results of a sine sweep are compared before and after the Random Vibration Test to verify there were no changes in frequencies. Any major changes would indicate an alteration in the structure and would need to be investigated. The workmanship vibration test of the ISEM assembly was performed successfully on all three axes, with no significant changes detected during the sine sweeps.

The ISEM assembly also underwent a full Thermal Vacuum (TVAC) Test, even though only a Thermal Cycle Test was required under the mission specification. A temperature profile range is selected based on the limits of the components involved and the expected temperatures on orbit, to expose the assembly to the maximum operational flexibility expected. The general profile consisted of two cycles in vacuum with a hot operational plateau of 50°C and a cold operational plateau of -10°C, at the ISEM baseplate interface. A full-functional performance test was performed at each plateau, with nominal on-orbit activities occurring during the temperature transitions. The test was performed using minimum and maximum input voltage at various stages in order to capture corner cases, as the specified input voltage could be subtly different based on power converter performance and signal integrity. The CSP performed nominally throughout the TVAC test, which indicates it is ready for mission exposure.

Radiation Tests

CSP was also tested against neutron radiation. High-energy neutron testing provides an estimation of system reliability in radiation-rich environments. The CSPv1 flight board was tested under a narrow beam for several days at the Los Alamos Neutron Science Center (LANSCE) in December of 2014 (shown in Figure 6). The recorded logs revealed the radiation-hardened watchdog timer rebooting the board and the EDAC Linux kernel module reporting ECC errors on the DRAM and parity errors in the L2 cache. Hundreds of errors reported by Linux kernel were logged over the serial terminal and analyzed later in the lab. Analysis of those logs indicated that about 75% of the reboots originated in L2 cache events, and it is suspected that a majority of the remaining events were caused by the L1 cache, which were not being reported at the time. There have been two publications made, which describe a more detailed analysis of these results and our conclusions and can be accessed in the presentation in [11].

Figure 6: CSPv1 LANSCE Neutron Test

Another neutron-beam test was performed in May 2015 at the TRIUMF facility in Vancouver for both the all-COTS CSPv1 board and a number of Zynq-based development boards in testing the cross-section for the caches and on-chip memory (Figure 7). Later analysis of the logs showed that the FPGA configuration-memory readback scrubber reported many single- and multi-bit upsets on the all-COTS CSPv1 board. Due to differences between the flight and COTS boards, we saw an increase in the frequency of having to manually power-cycle the COTS board in the beam. The cache and on-chip memory cross-section tests are currently under analysis, but show that the no-caches configuration makes a good case for improved reliability, although at the cost of performance.

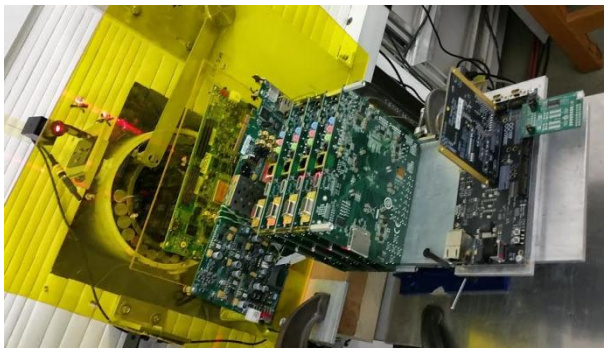

Figure 7: CSPv1 TRIUMF Neutron Test

V. ISEM MISSION CONFIGURATION

This section overviews the specific configurations of the ISEM-CSP flight box. The physical hardware is described followed by both the flight software on board the device and the ground-station software for commanding capability after launch.

Hardware Configuration

The STP-H5 ISEM-CSP flight box (Figure 8) is able to fit four boards in a 1U form-factor: two hybrid flight CSPv1 boards (CSP0, CSP1); one Power/Interface board; and one custom Backplane interconnect board. The two CSPv1 boards are set up in a master-slave configuration where CSP0 receives all ground commands and forwards requests to CSP1 as necessary. Due to the configuration, all ingoing and outgoing communication is directed by CSP0 through the Power/Interface board. The Backplane board is the central interconnect interface, connecting CSP0, CSP1, and the Power/Interface board together. Two SpaceWire and UART interfaces can be used to pass data between CSP0 and CSP1.

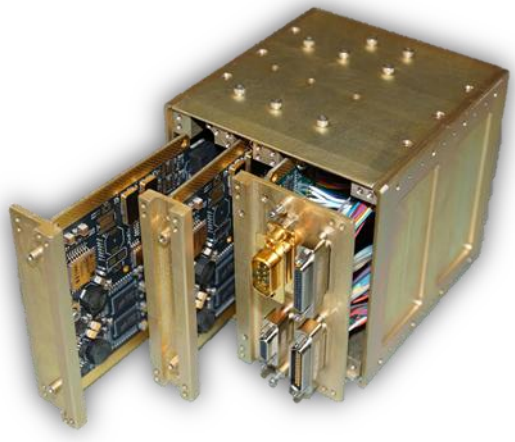

Figure 8: ISEM-CSP Flight Box

Four external connectors are provided on the CSP flight box: Camera Link; SpaceWire; power in; and debug I/O. These connections are made directly through the Power/Interface board, which functions as the system power supply as well as the CSP flight-box interface. The power-supply circuitry is hardened and provides the power rails necessary for the box to operate. External to the CSP flight box, a Sony 5-megapixel color camera is interfaced using a Camera Link FPGA core. Additionally, CSP0 contains a SpaceWire FPGA core to provide a communication interface to the SpaceCube Mini and ISS. A general connection overview for the ISEM stack is depicted in Figure 9.

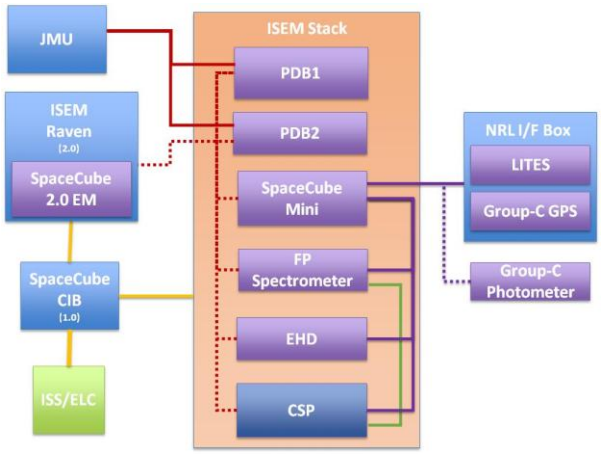

Figure 9: STP-H5/ISEM Configuration Overview

Software Configuration

The CSPv1s in the flight box also have a software configuration unique to the STP-H5 mission. Both CSPv1 flight boards are configured to boot from the on-board NAND flash. The Zynq's non-secure fallback feature is used to provide reliable booting. Four "golden" images are stored at the beginning of the flash. If the first image is corrupted in flight, then the Zynq BootROM loads the next image, and continues loading images until it finds a valid one. These images are in a read-only partition of the flash. The next partition contains space for updated boot images, which can be loaded post-launch. In each boot image, there is a First Stage Boot Loader, Second Stage Boot Loader (U-Boot), FPGA bitstream, and Wumbo Linux kernel. The Linux image uses an *initramfs* once booted and a non-volatile *JFFS2* flash filesystem is also mounted.

Contained in the Wumbo image is GSFC's C*ore Flight Executive* (cFE) along with several key *Core Flight System* (CFS) applications. cFE along with the *Operating System Abstraction Layer* (OSAL) are open source and can be found on SourceForge [13]. cFE is NASA Goddard's reusable flight software framework for local-device management, event generation, and software messaging, while CFS contains supporting applications and libraries [14]. Significant applications found in CFS that are used in the flight system are the Scheduler (SCH), Health Services (HS), File Manager (FM), and Stored Commands (SC). SCH is used mostly to schedule telemetry requests to our applications. HS is used primarily to handle watchdog interaction. FM is used to manipulate files in the NAND flash. Finally, SC is used to execute command sequences, such as an image capture at an absolute or relative time.

Custom CFS applications were developed for the ISEM-CSP mission and include: Command Ingest (CI); Telemetry Output (TO); File Transfer (FT); File Transfer Delivery Protocol (FTDP); FTDP Receive (FTDPRECV); FTDP Send (FTDPSEND); Image Processing (IP); Camera Control (CCTL); Self-Timer (SELF_TIMER); and CSP Health (CSPH). A custom communication library was designed as a frontend for CI and TO to the communication interface. Depending on compilation options, the backend can be either SpaceWire or POSIX sockets, and is designed to be transparent to applications. CCTL is used to interact with the camera, and communicates with SELF_TIMER to capture images at specified intervals. FTDP and FT are used for file upload and download, respectively. File uploads are performed over the Communications Interface Board (CIB) which acts as the interface between the ISS and all of the experiments on STP-H5, and downloads are streamed in with High-Rate Telemetry (HRT). IP creates thumbnails of captured images, which are streamed to the ground in JPEG2000 format. Lastly, CSPH streams health data such as device temperature, uptime, and memory and CPU utilization, from each of the two flight boards. An example diagram of the cFE architecture on CSP0 is featured in Figure 10.

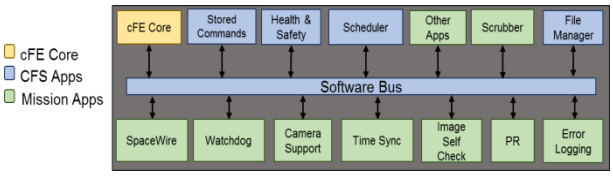

Figure 10: CFS Software Architecture on CSP

Ground-Station Software

To monitor the progress of the mission and perform all primary and secondary objectives, a ground station is setup with commanding software. The ground station deploys the *Telescience Resource Kit* (TReK) to receive and monitor packets sent from ISEM-CSP on board the ISS [15]. Packets can be received and sent through a graphical interface built to interact with the TReK software. This GUI was developed with the open source

Interoperable Remote Component (IRC) application framework with an example configuration for this mission provided by NASA Goddard [16]. The application framework uses XML descriptions that can be modified to easily parse, interpret, and display incoming data, as well as, send commands. IRC can be used to save and store commands through the GUI interface. The GUI also allows the operator to select and send commands. An example of the command GUI is shown in Figure 11. An example of the heath and status window generated by IRC is shown in Figure 12.

A Python extension was developed to interface with TReK using Python scripts. Python allows leveraging of the same scripts used during development without TReK present through the object-oriented structure it enables. One key Python script is the image viewer, which downloads and displays the thumbnail images streamed from the ISEM-CSP flight box.

Figure 11: Commanding Window

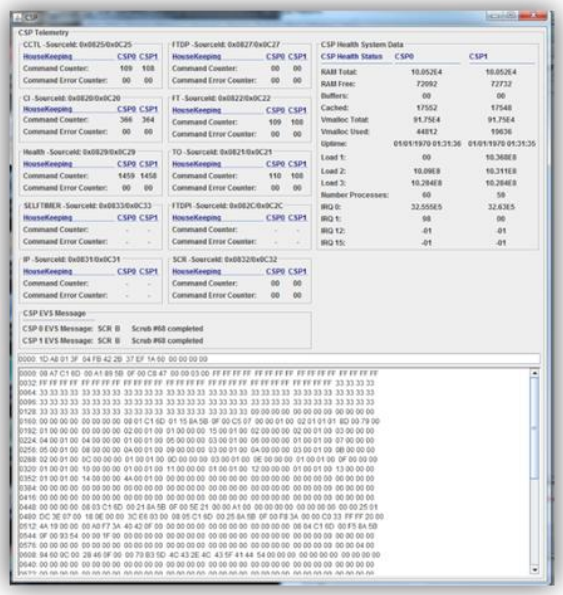

Figure 12: Health and Status Window

VI. PRIMARY OBJECTIVES

ISEM-CSP has several primary requirements to fulfill in order to declare mission success. The first objective of ISEM-CSP is to advance the TRL of the Xilinx Zynq SoC in Low Earth Orbit. This device is crucial for study in the development of a new generation of space computers. It is also one of many devices that are being considered for the next generation of the SpaceCube family of reconfigurable computers developed by NASA Goddard's Science Data Processing Branch.

Another key directive for the mission is to closely monitor and record the upset rates of both the processing system and programmable logic of the Zynq to provide environmental information in preparation for future missions. The main upset rates to be examined are the performance of the ARM cores, as well as, the L1 and L2 caches.

The final primary requirement is to perform image-processing techniques including noise reduction and image enhancement on terrestrial-scene data products. Image processing will be demonstrated with hardware acceleration in the FPGA fabric and compared with processing on the ARM cores with NEON acceleration. These high-resolution (5 megapixel) images can then be compressed using JPEG2000 or converted to .ppm for downlink as thumbnails or complete images and displayed on the ground-station system image viewer.

VII. SECONDARY OBJECTIVES

As a technology mission and experiment, ISEM-CSP has the freedom to explore additional research-oriented tasks as well as the ability to upload new applications and software, when not performing primary mission tasks. There are several secondary objectives that will be explored throughout the duration of the mission and are discussed in this section.

Autonomous Computing

The IP app provides access to our image-processing suite, which includes several algorithms to perform a variety of functions. For future space-processing missions, it may become necessary for processing tasks to be completed autonomously. Basic exploratory functions have been added to CSPv1 to begin testing this domain of applications. The IP app has a set of algorithms for classifying images. These algorithms can allow CSP0 to autonomously make decisions about which images to keep, without user intervention. In a restricted downlink scenario, this app can determine if an image taken is unnecessary (e.g., an all-white image from cloud cover, or all-blue from just the sea), and can delete the image, saving storage capacity as well as

preventing this picture from wasting downlink bandwidth.

In-Situ Upload Capability

The CSP flight box has additional software features, which include software and firmware uploads. Flight software updates will primarily be made by uploading new cFE table and configuration files. cFE tables can be used to change the behavior of applications, or even to load new applications. As an example, an SC table can be uploaded that includes commands for cFE to start an uploaded cFE application, or stop an old version and load a new one from flash memory. For more drastic changes, such as a Linux kernel update, new boot images can be uploaded and stored in the partition in the region after the golden images as described previously. The new environment will contain instructions for U-Boot on booting the new image. If the U-Boot environment ever becomes corrupt, U-Boot will default to booting the golden image. Lastly, additional functionality on this mission includes file transfer between CSP0 and CSP1. The FTDPRECV and FTDPSEND apps can allow the transfer of large files or configurations between the two flight boards.

Partial Reconfiguration

The CSPv1 will be one of the first deployed space computers to include Partial Reconfiguration (PR) functionality. PR is the process of changing a specialized section of reconfigurable hardware during operational runtime. The CSPv1 allows multiple applications to be performed in the FPGA fabric without reconfiguring the entire device. PR can be used in space missions to reduce the total-area utilization of the fabric by switching out designs to reduce the vulnerable configuration area, employing fault-tolerant reconfigurable structures, and allowing new algorithms and applications to be uploaded after completion of the primary mission. PR can improve the performance of a device by allowing the user to include a suite of application designs to fit within a PR region, enabling a larger number of applications to be accelerated by hardware, rather than limited by a single static FPGA design. The CSPv1's corrective scrubbing and error logging are also available to PR design regions.

Space Middleware

The CSP explores new fault-tolerant approaches beyond pure hardware radiation-tolerance by extending its fault-mitigation considerations to flight software. In contrast to FPGA mitigation techniques discussed in previous sections, this experimental research takes a processor-centric perspective to assist in developing resilient applications on the processing system as the *Adaptive Dependable Distributed Aerospace Middleware* (ADDAM). The ADDAM research is motivated by a pursuit to provide a middleware platform of software services for fault-tolerant computing in harsh environments where execution errors are expected to be common in occurrence.

The means for accomplishing software resilience is through process redundancy: through a system of multiple processes operating in pursuit of a common application, the resilience is ameliorated while mitigating individual instances of execution failure. In order to recover from potential failures in processes over the application execution, the processes are developed with ADDAM through task division. Task division in the system is modeled after a traditional message-passing system and these tasks can be distinct for distributed processing, or replicated for increased redundancy.

Each process has a unique identifier, referenced globally in the network of processes for peer communication. The identifier is also used for correlating a process with its role of either the *coordinator* instance or *worker* instance, of which the same process can assume either role as needed. Worker failover is handled by task re-issue from the coordinator, coordinator failover is being developed through distributed election, and both types of failover are assisted with process restart through a cyclical processor monitor to prevent ADDAM process extinction through successive execution faults.

The latest prototype of ADDAM provides fault awareness to an app developer via an internal publish/subscribe messaging system for propagating events. The messaging system operates on events generated by discrete modules based on specific functionality. Currently, ADDAM generates events for *process discovery*, tracking peer connections and disconnections through heartbeats for the *health reactor*, which in turn generates events used for both the *task manager* as it dispatches workload divisions specified by the developer, and the *coordination manager* for determining process roles. Advanced fault-mitigation strategies and execution patterns can be developed to adapt behavior depending on mission parameters. Through this system, an extensible platform for generating fault awareness is available as another tool for incorporating fault-tolerant computing techniques onto a variety of space computers.

Device Virtualization and Dynamic Synthesis

The last secondary goal of the ISEM-CSP mission is to demonstrate an improved productivity tool set by generating FPGA designs through device virtualization

and dynamic synthesis. This research will allow future adopters of CSPv1 to have an easier effort in adapting FPGA designs to make use of the full SoC system. The performance and power advantages of FPGA hybrid computing system are well established, but have attendant challenges that have limited adoption of the technology. From the perspective of application designers, writing FPGA-accelerated code is a time-consuming process, complicated by low-level and relatively unfamiliar hardware-description languages (e.g. VHDL) typically used in design, and lengthy hardware-compilation times of tens of minutes to hours required even to make minor design changes [18]. The effectiveness of FPGA-accelerated cores is also limited by the efficiency of data transfer between the design cores and host software, which requires careful consideration of data-access patterns and work in kernel to optimize memory bandwidth.

From the perspective of system designers, FPGA acceleration poses additional challenges: how can multiple applications be supported efficiently using common and limited hardware resources (e.g. ultimately FPGA area); how can these systems be made resilient against changing applications and workloads; and how can system security be ensured when applications are encouraged to modify hardware, especially hardware with access to system memory and other privileged resources? These challenges are even more significant for space systems, where high launch costs can be better amortized by more flexible systems. Similarly, the cost of system failure due to errant hardware is significantly higher, with limited options for remediation.

Academic work on device virtualization and dynamic synthesis from high-level languages such as OpenCL [18] has shown significant promise to help address these challenges [17]. Device virtualization raises the fine-grained FPGA device (e.g. lookup table and register logic resources) up to the higher level of an application or domain by compiling to flexible high-level *overlays* rather than directly to the device.

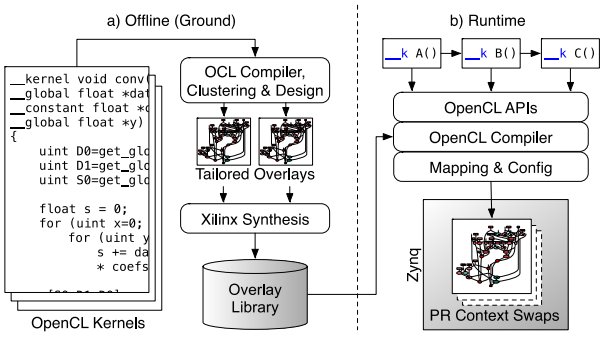

Figure 13: *CLIF* OpenCL Framework

CSPv1 integrates an implementation of OpenCL that uses this approach, called *CLIF* [17] [18], as illustrated in Figure 13. Applications using this framework are written against a C task and data API, with computational kernels specified in the OpenCL kernel language. Unlike other OpenCL implementations for FPGAs, applications package their kernels' source and rely on *CLIF*'s *runtime compiler* to handle device mapping. This mapping is performed using overlays from the system's overlay library, which can improve system flexibility in multiple scenarios:

- Hardware/software partitioning is deferred until runtime, where it may be informed by dynamic properties of the system (e.g., power, damaged regions, or the needs of other workloads).
- New applications or changes are added by small patches to application software, and hardware accelerated using support already in the overlay library or added through newly uploaded overlays.
- The system is free to introduce error mitigation or detection, or even optimizations, without requiring changes to application software (e.g., binding to fault-tolerant overlay instances).

This approach has other benefits for system design and security. High-level kernel descriptions permit the compiler to perform optimizations that can be infeasible for human designers. For example, previous work has shown that aggressive inter-kernel resource sharing using overlays can result in up to 70% lower area [18], with up to 250x faster kernel switches [17]. Since applications are implemented using the system's overlays rather than directly using FPGA resources, security policies can be enforced by restricting the capabilities provided by this overlay library. For example, in our implementation, accelerators have high-performance access to system memory through the Zynq coherency port. However, the addresses kernels can access over this interface are restricted by each overlay's memory controller to protect against faulting or malignant applications.

VIII. CONCLUSIONS

In this paper, we introduced the STP-H5 ISEM-CSP mission hardware and software configuration as well as the primary and secondary goals. The mission will serve as a TRL advancement and space validation for the CSPv1 board and its supporting software. During the mission, we will collect valuable radiation data and upset rates to the CSPv1 boards and gain insight to make improvements to the design. This mission is also a unique opportunity to test techniques and applications described in Section VII that have not been attempted

in space systems to date. The ISEM-CSP flight box is the first venture into exploring the capabilities of the CSPv1 flight board and the CSP concept in a real space environment, and will help advance studies for the next generation of space processors and prepare CSPv1 for future advanced missions.

Due to its capability to upload new flight hardware and software, the ISEM-CSP flight box will be a continuous development platform to upload new applications and software well after all of its objectives have been completed. This functionality provides an opportunity to test the effectiveness of these applications on an actual space platform without needing to wait for a new mission with a scheduled launch, thereby accelerating development and verification of new flight software.

Acknowledgments

This work was funded by the industry and government members of the NSF CHREC Center, which include many of the leading industries and agencies in the US space program, as well as the I/UCRC Program of the National Science Foundation under Grant Nos. EEC-0642422, IIP-1161022, and IIP-1265957. In addition to the authors, more than a dozen CHREC students at the University of Florida contributed heavily to the hardware and software development of CSPv1. Special thanks to Dylan Rudolph, Antony Gillette, Bryant Cockcroft, Sebastian Sabogal, Daniel Sabogal, Dorothy Wong, Brad Shea, Chris Morales, Andy Wilson, Ryan Zavoral, Rainer Ledesma, Travis Wise, Jay Wang, Joe Kimbrell, Dr. Herman Lam, and Dr. Ann Gordon-Ross for their contributions. We also would like to thank all the additional support received from NASA Goddard for the environmental testing and the mechanical box design, which includes Alessandro Geist, Keven Ballou, Dave Petrick, Mike Lin, Allison Willingham, Matt Colvin, Eric Gorman, Tracy Price, Curtis Dunsmore, and Katie Brown. We also gratefully acknowledge donations and support from Xilinx and Intersil that helped make this work possible, as well as support from the Department of Energy and Cisco for the Los Alamos and TRIUMF radiation tests, respectively.

References

1. Salas, A., Attai, W., Oyadomari, K., Priscal, C., Shimmin, R., Gazulla, O. and J. Wolfe, "Phonesat In-Flight Experience Results," Small Satellites Systems and Services Symp., Porto Petro, Majorca, Spain, May 26-30, 2014.
2. Allmen, J., and A. Petro, "Small Spacecraft Technology," Proc. of the 28th Annu. AIAA/USU Conf. on Small Satellites, Logan, UT, Aug. 3, 2014.
3. Yost. B, "EDSN – Edison Demonstration for SmallSat Networks Overview," CubeSat Summer Workshop, Logan, UT, Aug. 10-11, 2013.
4. LaBel , K. A., Johnston , A. H., Barth , J. L., Reed, R. A., and C. E. Barnes, "Emerging radiation hardness assurance issues: A NASA approach for space flight programs." IEEE Trans. Nucl. Sci., vol. 45, pp.2727 -2736, 1998.
5. Anthes, R. A. et al., Earth Science and Applications from Space: A Midterm Assessment of NASA's Implementation of the Decadal Survey. Washington, DC: Space Studies Board, National Research Council, 2007.
6. *Space Test Program Fact Sheet*, Kirtland AFB, Jan. 25, 2007. [Online]. Available: http://www.kirtland.af.mil/library/factsheets/factsheet.asp?id=6878
7. McLeroy, J., "Highlights of DoD Research on the ISS," ISS Research and Development Conf., June 26-27, 2012.
8. Petrick, D., "SpaceCube High Performance Hybrid Data Space Processing System," Information Science and Technology Colloq. Series, Greenbelt, MD, Nov 17, 2014.
9. Lin, M., Flatley, T., Geist, A., and D. Petrick, "NASA GSFC Development of the SpaceCube MINI," Proc. of the 25th Annu. AIAA/USU Conf. on Small Satellites, Logan, UT, Aug. 8-11, 2014.
10. Rudolph, D., Wilson, C., Stewart, J., Gauvin, P., George, A. D., Lam, H., Crum, G., Wirthlin, M., Wilson, A., and A. Stoddard, "CSP: A Multifaceted Hybrid Architecture for Space Computing," Proc. of the 28th Annu. AIAA/USU Conf. on Small Satellites, Logan, UT, Aug. 2-7, 2014.
11. Harward, N., Gardiner, M., and M. Wirthlin, "Neutron Radiation Test Results of the Linux Operating System Executing within the CHREC Space Processor (CSP)," in Proc. of Military and Aerospace Programmable Logic Devices (MAPLD), San Diego, CA, May 18-21, 2015.
12. Flatley, T., "SpaceCube - A Family of Reconfigurable Hybrid On-Board Science Data Processors," Int. Conf. on ReConFigurable Computing and FPGAs, Cancun, MX, Dec. 7-11, 2015.
13. *Coreflightexec*, NASA Goddard Space Flight Center, Dec. 12, 2012. [Online]. Available: http://sourceforge.net/projects/coreflightexec/

14. McComas, D., "NASA/GSFC's Flight Software Core Flight System," Flight Software Workshop, San Antonio, Tx, Nov. 7-9, 2012.

15. *Telescience Resource Kit*, NASA Marshall Space Flight Center, June. 2, 2012. [Online]. Available: http://trek.msfc.nasa.gov/

16. *Interoperable Remote Component*, NASA Goddard Space Flight Center, Feb. 27, 2015. [Online].Available: http://opensource.gsfc.nasa.gov/projects/IRC/index.php

17. Coole, J. and G. Stitt, "Adjustably Flexible, Low-Overhead Overlays for Runtime FPGA Compilation," The 22nd Annu. IEEE Int. Symp. on Field-Programmable Custom Computing Machines (FCCM), May 3-5, 2015.

18. Coole, J. and G. Stitt, "Fast and Flexible High-Level Synthesis from OpenCL using Reconfiguration Contexts," IEEE Micro, vol. 34, pp. 42-53, 2014.