Technology Validation: NMP ST8 Dependable Multiprocessor Project

John Samson, Gary Gardner, David Lupia
Honeywell Inc., Aerospace Systems
john.r.samson@honeywell.com

Minesh Patel, Paul Davis, Vikas Aggarwal
Tandel Systems LLC
mpatel@tandelsystems.com

Alan George
University of Florida
george@hcs.ufl.edu

Zbigniew Kalbarcyzk,
University of Illinois/Armored Computing, Inc.
kalbar@crhc.uiuc.edu

Rafi Some
Jet Propulsion Laboratory, California Institute of Technology
Raphael.Some@jpl.nasa.gov

Abstract—With the ever-increasing demand for higher bandwidth and processing capacity of today's space exploration, space science, and defense missions, the ability to efficiently apply Commercial-Off-The-Shelf (COTS) processors for on-board computing has become a critical need. In response to this need, NASA's New Millennium Program (NMP) commissioned the development of Dependable Multiprocessor (DM) technology for use in science and autonomy missions, but the technology is also applicable to a wide variety of DoD missions. The goal of the DM project is to provide spacecraft/payload processing capability 10x – 100x what is available today, enabling heretofore unrealizable levels of science and autonomy. DM technology is being developed as part of the NMP ST8 (Space Technology 8) project. The objective of this NMP ST8 effort is to combine high-performance, fault tolerant, COTS-based cluster processing and fault tolerant middleware in an architecture and software framework capable of supporting a wide variety of mission applications. Dependable Multiprocessor development is continuing as one of the four selected ST8 flight experiments planned to be flown in 2009. [1, ,2]

There are three key problems that need to be overcome in order to fly COTS in space: 1) an effective approach for handling SEUs (Single Event Upsets) in high performance cluster processors, 2) handling thermal issues associated with state-of-the-art COTS components, and 3) achieving high power efficiency (throughput per watt). DM technology solves all three problems. DM solves the SEU problem by combining cluster management software with SEU tolerance-enhancing software in a flexible, efficient, integrated DM middleware suite. DM solves the thermal issue by mining the ruggedized, conductive-cooled, COTS airborne embedded processing domain. DM solves the power efficiency problem by mining the high performance, low power mobile computing processing domain.

Recently, the DM project successfully passed several key NMP ST8 project milestones: the TRL5 (Technology Readiness Level 5) technology validation demonstration, the Experiment Preliminary Design Review, and the NASA Non Advocate Review. Passing the TRL5 milestone qualified the DM project for advancement to flight system development status. The ST8 Project passed its Preliminary Design and Confirmation Review and is now in its Implementation Phase. This paper describes the status of the project, the technology validation experiments and demonstrations achieved to date, the plans for the TRL6 technology validation effort, and the plans for the TRL7 flight validation.

TABLE OF CONTENTS

[1] Prepared for NSTC 2007. This paper is an update of a paper presented at the 2007 IEEE Aerospace Conference, Big Sky, MT, March 8, 2007. [1]
[2] The project formerly was known as the Environmentally-Adaptive Fault-Tolerant Computing (EAFTC) project.

1. INTRODUCTION

Many next-generation space missions will require onboard high-performance processing for science payloads as well as for autonomous data analysis and mission planning. Current space-qualified computing systems, built around radiation-hardened processors, cannot provide sufficient performance, i.e., throughput MOPS (Millions of Operations Per Second), or performance-density, e.g., MOPS per watt, to meet these requirements. In terrestrial laboratories, science data processing is performed on parallel processing cluster computers. Similarly, the complex models envisioned for future highly autonomous robotic systems also need high-performance, parallel or supercomputer architectures to meet near real-time requirements. A cluster computer comprises a set of single board computers, interconnected by a high speed switched network, running a file-oriented multi-threading operating system and a "middleware" which controls and coordinates parallel processing applications. A typical system might consist of 10 to 20 Motorola G4-based single board computers, interconnected via a Gigabit Ethernet, running the LINUX operating system and an MPI middleware. The parallel processing applications are typically written in a version of FORTRAN, C or C++ and are supported by parallel math libraries such as ScaLAPACK or PLAPACK (Parallel Linear Algebra PACKage). In the most advanced architectures, Field Programmable Gate Arrays (FPGAs) are used to implement the algorithms directly in hardware. FPGAs allow configuring of hardware "on the fly" and provide the most power and time efficient implementations of mathematical routines.

Over the past few generations, COTS computer components have become extremely resistant to the debilitating effects of radiation. Many commercial parts can withstand many 10s of kilorads of Total Ionizing Dose (TID) and are immune to catastrophic Single Event Latchup (SEL). The primary issue preventing the deployment of a COTS-based spaceborne cluster computer is their continued susceptibility to Single Event Upsets or SEUs, (a.k.a. soft errors). SEUs however, unlike TID and SEL, entailing only a bit flip from 1 to 0 or 0 to 1, do not cause permanent damage. Further, in the latest generation of computer electronics, SOI CMOS (Silicon on Insulator Complementary Metal Oxide Semiconductor) has proven to be approximately an order of magnitude less susceptible to SEU than previous bulk CMOS. If we can withstand a few errors per day per processor, without unduly impacting system dependability, it would be possible to fly essentially commercial cluster computers. Not only would this provide mission enabling performance and performance-density levels, but it would significantly lower the cost of development as standard laboratory science codes could be easily ported to these systems without the expensive and error prone process normally associated with moving complex codes from the lab to a new platform.

The Honeywell Dependable Multiprocessor experiment will validate the technological concept, the architecture, the fault tolerance techniques and the associated performance, reliability, and availability models behind this technology. Supplementing ground-based testing, the in-space validation will test those aspects of the technology which cannot be effectively exercised on the ground. This includes the ability to withstand concurrent omni-directional, multi-species, multi-energy, and extremely high energy radiation while meeting required reliability and availability levels. The experiment will also provide the data required to calibrate the associated models and to allow scaling of the models to radiation and computing environments well beyond the ST8 environment.

2. TECHNOLOGY OVERVIEW

With the ever-increasing demand for higher bandwidth and processing capacity of today's space exploration, space science, and defense missions, the ability to efficiently apply COTS processors for on-board computing has become a critical need. In response to this need, NASA's NMP commissioned the development of DM technology for use in science and autonomy missions, but the technology is also applicable to a wide variety of DoD missions. DM technology is a COTS-based, power-efficient, high-performance, highly dependable, fault-tolerant cluster computer.

There are three key problems that need to be overcome in order to fly COTS in space: 1) handling SEUs in high performance cluster processors, 2) handling thermal issues associated with state-of-the-art COTS components, and 3) achieving high power efficiency (throughput per watt). DM technology solves all three problems. DM solves the SEU problem by combining cluster management software with SEU tolerance-enhancing software in a flexible, efficient, integrated DM middleware suite. DM solves the thermal issue by mining the ruggedized, conductive-cooled, COTS airborne embedded processing domain. DM solves the power efficiency problem by mining the high performance, low power mobile computing processing domain.

While current COTS high performance processors are exhibiting adequate TID performance to meet the requirements of the natural space radiation environment, SEUs caused by heavy ions and solar flares are, and will remain, a problem. Traditional approaches to mitigate the SEU problem involve fixed redundancy schemes such as Self Checking Pairs (SCP) or Triple Modular Redundancy (TMR). While effective in mitigating the effects of SEUs, use of these techniques comes at a high price, 100% overhead for SCP, and 200% overhead for TMR. This is particularly vexing in the broad range of applications where such a level of protection is not needed. In such cases, it would be beneficial to be able to convert that unneeded overhead into useful mission processing capability. The idea behind DM is to be able to configure the processing system

to maximize the processing capability available to the mission

To satisfy this need, the DM concept has been demonstrated and is currently being developed further as one of the flight experiments for NASA's NMP ST8 project. The objective of this NMP ST8 effort is to combine high performance, fault tolerant, COTS-based cluster processing with replication services, Algorithm-Based Fault Tolerance (ABFT), and fault tolerant middleware in an architecture and software framework capable of supporting a wide variety of mission applications.

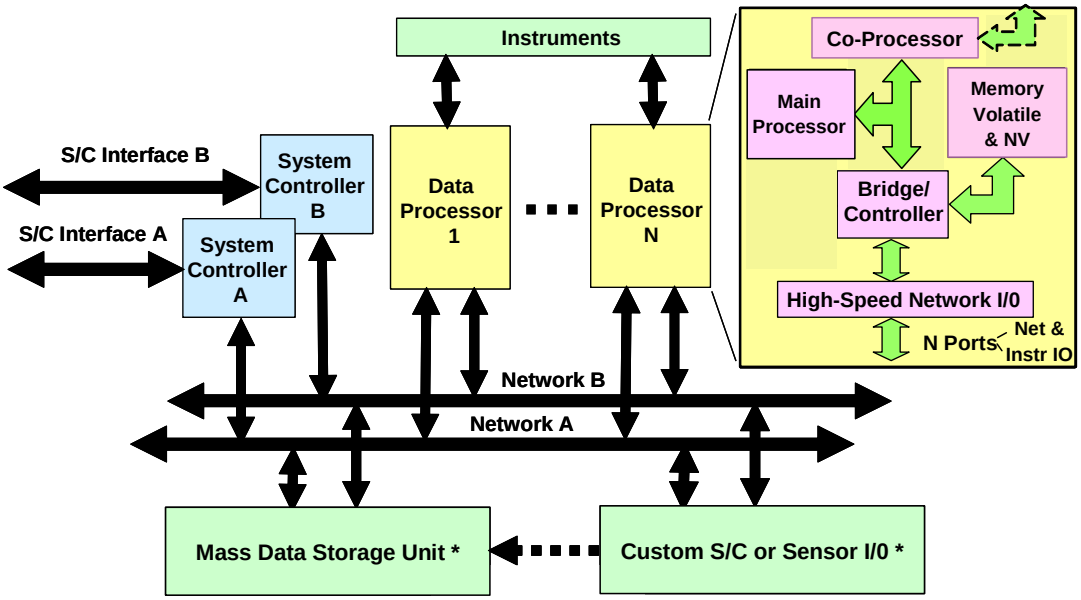
The objective of the ST8 DM technology advance is to demonstrate a high-performance, COTS-based processing cluster that can operate in a natural space environment providing high-throughput, low power, scalability, and fully programmable processing achieving high throughout density (> 300 MOPS/watt), technology independent system software that manages the cluster of COTS processing elements, technology independent system software that enhances radiation upset tolerance, high system availability (> 0.995), and low system unreliability (< 0.005) in terms of the probability of delivering undetected erroneous or untimely data.

DM technology comprises four key elements:

- An architecture and methodology which enables the use of COTS-based, high-performance, scalable, multi-computer systems in a space environment, incorporating reconfigurable and high performance algorithmic co-processors, supporting parallel/distributed processing for science codes, and accommodating future COTS parts/standards through upgrades.
- Application software development and runtime environments that are familiar to science application developers and facilitate porting of applications from the laboratory to the spacecraft payload data processor.
- An autonomous and adaptive controller for fault tolerance configuration, responsive to environment, application criticality, and system mode, that maintains required dependability and availability while optimizing resource utilization and system efficiency.
- A methodology and tools which allow the prediction of the system's behavior in the space environment, including: predictions of availability, dependability, fault rates/types, and system level performance.

The DM hardware architecture is depicted in Figure 1. The basic architecture consists of a redundant radiation-hardened system controller which acts as the highly reliable controller for a parallel processing cluster of COTS-based, high-performance, data processing (DP) nodes, a redundant network interconnect, and a redundant spacecraft interface. The system can be augmented with mission-specific elements, including mass storage, custom interfaces, and radiation sensors as required.

The DM software architecture framework is depicted in Figure 2. Figure 2 shows the two types of processing nodes: the first type, the reliable system controller, which can operate through any foreseeable environment without upsetting, for control functions, and the second type, a high performance, COTS-based, cluster processing node. A high



*** Examples: Other mission-specific functions**

Figure 1 – Dependable Multiprocessor Hardware Architecture

level API (Application Interface) and a high level SAL (System Abstraction Layer) provide both application independence and platform independence, while allowing the particular mission applications and platforms to take advantage of fault tolerance services and reliable messaging offered by the generic fault tolerant middleware layer. The function of the DM system software is two-fold: 1) to support cluster operation for scalable high performance systems, and 2) to provide a system environment that enhances SEU tolerance through software fault tolerance techniques.

In order to support scalable cluster processing, DM system software encompasses: system initialization/re-initialization including discovery/membership, self-test, the establishment of communication, and the establishment of system resource tables; basic job management services including loading/unloading, starting/stopping, pausing/resuming, transition handling, and dynamic maintenance of job and resource tables; basic job execution services including job scheduling (periodic scheduling, frame-based gang scheduling, a-periodic scheduling, triggered scheduling, continuous scheduling, and single executions) and job synchronization/coordination (application-based, process-based, task-based, event-based, and data-based); basic communication services including reliable messaging and user level APIs; and basic resource management services including effecting established mission policies and application execution modes, keeping track of resource status (busy/active/halted nodes, busy/active halted jobs and processes) and dynamic maintenance of resource tables.

As mentioned previously, one of the main problems flying COTS in space is the occurrence of SEUs. Physically, SEUs affect the hardware, but manifest themselves in software as errors. The types of soft errors encountered in applications and middleware include: data errors, control flow errors, hangs and crashes, OS exceptions induced by the applications, and communication errors and time-outs. The types of soft errors encountered in the OS include: kernel PANICs causing an OS hang or crash, OS exceptions from hardware detection mechanisms, and communication protocol errors. In the application and middleware domains, soft errors can be detected by Operating System (OS) exception captures, by replication, by heartbeat and thread monitoring, by hang and crash timers, by exit handlers, by message traffic monitors, and by message error checking. Application data errors can be detected with spatial and temporal replication (SCP and TMR) and with ABFT techniques. Soft errors in the OS can be detected with exception handlers, heartbeat monitors, microprocessor and bridge chip exception capture, and communication error checking.

Recovery approaches range from periodic check-pointing and roll back, roll forward, and communication retry to soft or hard resets to system re-boot if the effects of the soft error are severe enough to warrant it. TMR voting schemes and some ABFT techniques support immediate recovery and continued operation though the error.

In order to support enhanced SEU-tolerant performance in the DM, the traditional resource management services have been augmented with fault tolerant modes of operation

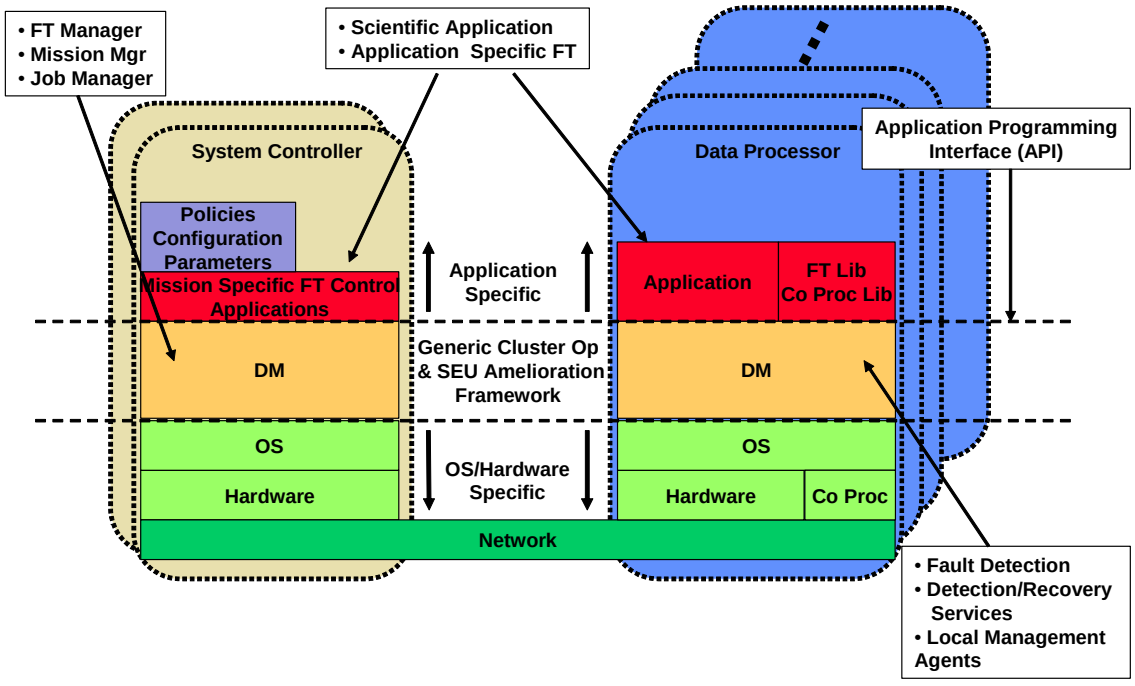


Figure 2 – Dependable Multiprocessor Software Architecture

including hardware (spatial) and software (temporal) redundancy, rapid detection and recovery from soft errors, rapid detection and recovery from hard faults, and fault/error management services including fault/error logging, fault/error handling diagnostics, management of resource health status, management of application/process status, and management of redundancy and sparing. The flexible, efficient, and cost-effective integration of user-selectable cluster management and SEU tolerance enhancement software to achieve high reliability and high availability in a wide variety of missions and environments is the key benefit of DM technology.

Implementation of this functionality is embodied in the major components of the DM middleware which are: the High Availability Middleware (HAM), the System Services (SS), the Application Services (AS), and MPI Services, the Fault Tolerant Embedded Message Passing Interface (FEMPI). System Services encompass the Job Manager (JM), the Job Manager Agents (JMAs), the Mission Manager (MM), the Fault Tolerance Manager (FTM), and the Mass Data Storage Manager (MDSM). Application Services encompass Co-processor, e.g., FPGA, Services (FCPS), Replication Services (RS), and Algorithm-Based Fault Tolerance (ABFT). The Job Manager, the Job Manger Agents, the Fault Tolerance Manager, the Mission Manger, and the High Availability Middleware form the software control services for the DM system. More information about these DM middleware functions can be found in Reference [1].

As part of the TRL5 technology validation demonstration, a set of software fault injection campaigns were run, during which the DM system was subjected to thousands of fault injections emulating SEUs. Coverage, detection and recovery latency, throughput, overhead, and fault tolerance performance data were recorded and fed into predictive Reliability, Availability, and Performance Models, demonstrating the effectiveness of the DM system in different environments. In addition to the software fault injection campaigns, thirty-three (33) experiments were run with representative science applications executing with various modes of fault tolerant operation. Preliminary radiation testing of key COTS components selected for the flight experiment showed that these components exhibited no catastrophic latch-up and a sufficient number of SEUs to support the flight validation experiment.

A top-level overview of the DM software architecture is provided in Figure 2. A key feature of this software architecture is the incorporation of a set of generic fault tolerant middleware techniques implemented in a software framework that is independent of and transparent to the specific-mission application, and independent of and transparent to the underlying platform (HW and Operating System). This independence and transparency is achieved through well-defined, high-level, application interfaces, an API (Application Programming Interface) to support mission-specific application needs, and a SAL (System Abstraction Layer) which isolates the remainder of the software system from the underlying platform, simplifying the porting of this software system to other platforms and allowing the generic fault tolerance middleware services to be available to future mission applications on future onboard processing platforms. More information on the Dependable Multiprocessor and related technologies can be found in references [1] – [18].

TECHNOLOGY BENEFITS

The success of DM technology offers many benefits to future users: 10x – 100x more delivered computational throughput in space than currently available enabling heretofore unrealizable levels of science data and autonomy processing; faster, more efficient application software development via robust, COTS-derived fault tolerant cluster processing, the ability to port applications directly from the laboratory to the space environment encompassing MPI-based middleware and compatibility with standard cluster processing software including existing parallel processing libraries; minimization of non-recurring development time and costs for future mission; highly efficient, flexible, and portable SW fault tolerance approaches applicable to space and other harsh environments; and direct portability to future advances in hardware and software technology.

One of the goals of the DM project is to provide spacecraft/payload processing capability 10x – 100x what is available today. Figure 3 depicts the potential benefits of DM technology applied to the IOMI (Indian Ocean Meteorological Instrument) project which was performed in conjunction with the NMP EO3 GIFTS (Geosynchronous Imaging Fourier Transform Spectrometer) effort. A comparison of performance and performance density for a 1K complex FFT (Fast Fourier Transform) benchmark is provided for today's technology shown above the dotted line and Dependable Multiprocessor technology shown below the dotted line. The FFT example was chosen because it is a familiar benchmark and is a function found in many science applications. One of the key elements of the DM implementations is the high Reliability and high Availability provided by the DM fault tolerant middleware and supporting fault tolerance techniques such as Replicated Services and ABFT.

NMP EO3 Geosynchronous Imaging Fourier Transform Spectrometer Technology Indian Ocean Meteorological Instrument (IOMI) - NRL

Radiation Tolerant 750 PPC SBC		**133 MHz ~ 266 MFLOPS ~ 1.2 kg**	**1K Complex FFT in ~ 448 μsec ~ 13 MFLOPS/watt**
Radiation Hardened Vector Processor		**DSP24 @ 50 MHz ~ 1000 MFLOPS ~ 1.0 kg**	**1K Complex FFT in ~ 52 μsec ~45 MFLOPS/watt**

NMP ST8 Dependable Multiprocessor Technology

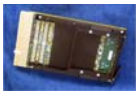
7447a PPC SBC with AltiVec		**800 MHz ~ 5200 MFLOPS ~ 0.6 kg**	**1K Complex FFT in ~ 9.8 μsec ~ 520 MFLOPS/watt**

Figure 3 – Dependable Multiprocessor Technology Benefit Example: Comparison of NMP ST8 Dependable Multiprocessing Technology and Technology That Would Be Flying Today on NMP EO3

3. TECHNOLOGY VALIDATION PLAN

The overall DM technology validation plan is depicted in Figure 4, starting with the TRL4 validation at the end of Phase A, the Concept Formulation Phase. This is followed by the TRL5 validation at the end of Phase B, the Formulation Refinement Phase, the TRL6 validation at the end of the Phase C, the Implementation Phase, and culminates with the TRL7 validation in the Flight Experiment Operations Phase. Each new validation level is characterized by increasing system fidelity and integration. One of the key elements of TRL5 milestone is the development and validation of models which can be used to predict the performance, reliability, and availability of the DM in the ST8 flight experiment and in future NASA missions. The TRL6 and TRL7 experiments will refine and validate the models and the parameters used in the models. After a successful TRL4 demonstration at the end of Phase A, which proved the underlying environmental monitoring and reconfiguration capabilities of the DM system, NASA requested that the TRL5 effort focus more on high-performance, fault-tolerant cluster processing with fault-tolerant MPI (Message Passing Interface) capability to provide a software development environment that is familiar to NASA science application developers.

Models

One of the key DM project deliverables is the set of models which can be used to predict DM performance in future NASA missions in different radiation environments, in different orbits, and with technology upgrades, and descriptions of how to use them. The objective of the TRL5, TRL6, and TRL7 technology validation experiments is to validate the models and the parameters used in the models. There are five (5) basic models: the Canonical Fault Model, the Radiation Effects and Hardware SEU Susceptibility Model, the Availability Model, the Reliability Model, and the Performance Model. Figure 5 depicts the DM modeling flow and the inputs and outputs of each model. The Canonical Fault Model identifies the faults which are used as the basis for the Hardware SEU Susceptibility Model and against which the fault tolerance performance of the DM will be evaluated. The Radiation Effects Model takes into account the DM system architecture, the DM hardware architecture, the mission orbit, the mission epoch or time frame, and the radiation characterization of the components. The Radiation Effects Model outputs the expected particle fluxes, energies, and component SEEs (Single Event Effects) for the given orbit. The Hardware SEU Susceptibility Model outputs the fault rates for each fault type in the Canonical Fault Model. The outputs of the Hardware SEU Susceptibility Model are combined with the detection coverage for each fault/error type in the Canonical Fault Model, the recovery coverage for each fault/error type in the Canonical Fault Model, the detection and recovery latencies for each fault/error type in the Canonical Fault Model, the probability that a particular fault affects the application, the number of expected mode changes for the mission, and the time to effect the mode change to predict the Availability and Reliability of the DM in the particular mission application. The Performance Model takes into account the mission application, the peak throughput of the CPUs in the high- performance data

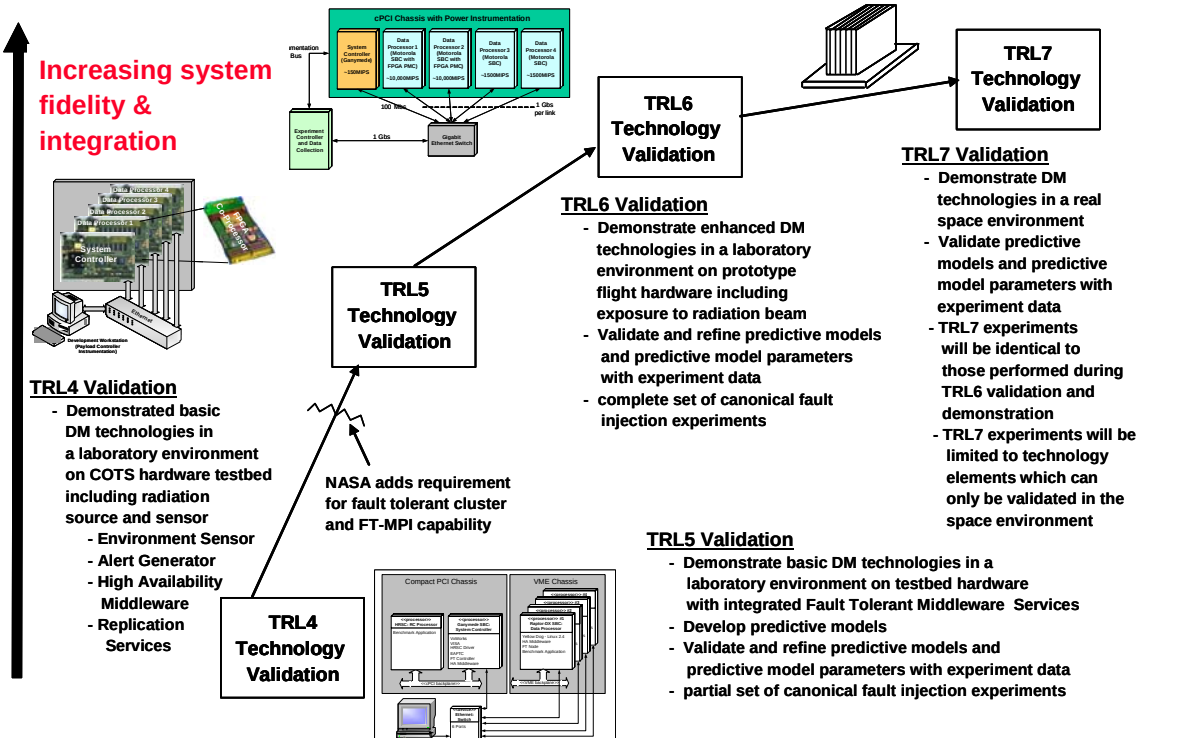


Figure 4 – Dependable Multiprocessor Technology Validation Plan

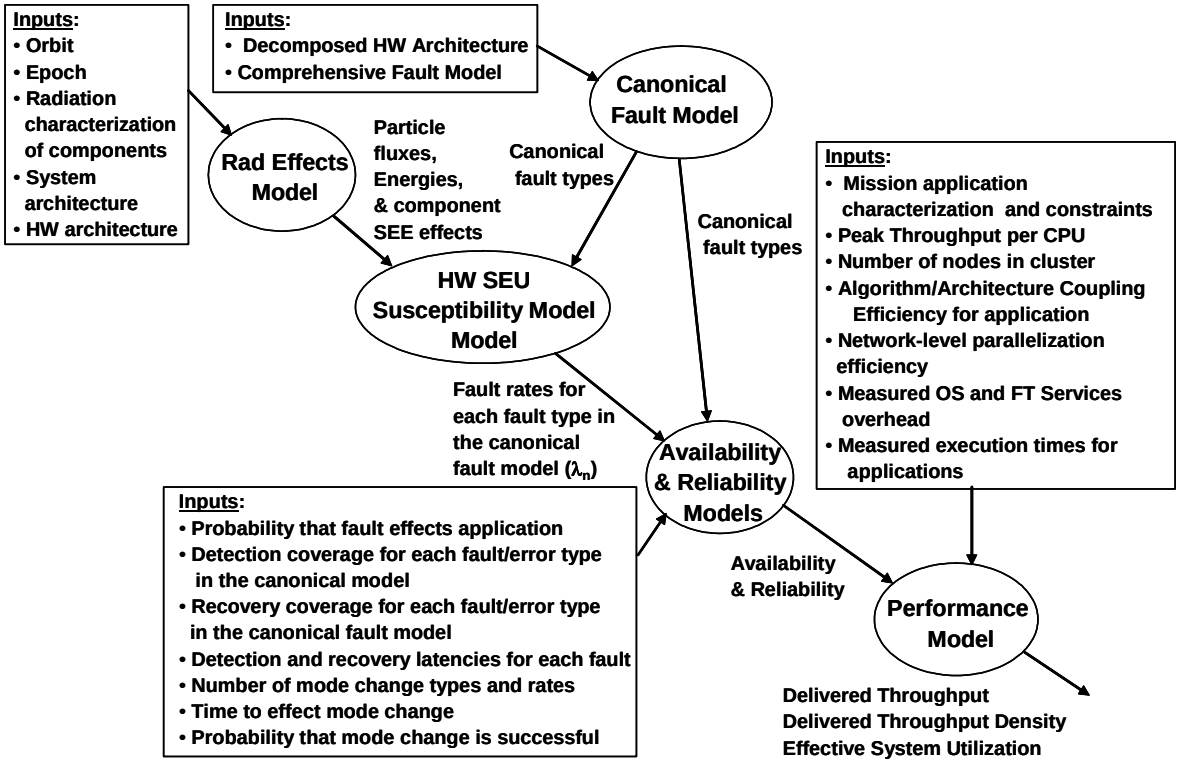


Figure 5 – Dependable Multiprocessing Model Flow

processing nodes, the algorithm/architecture coupling efficiency for the application, the number of nodes in the cluster, the network-level parallelization efficiency, the measured OS and FT services overhead, and the measured execution times for the applications to determine the delivered throughput (MOPS), the delivered throughput density (MOPS/watt), and the effective system utilization for the mission.

4. TRL4 VALIDATION

At the TRL4 TMA (Technology Maturity Assessment), which was conducted at end of the Concept Formulation Phase, the basic environmentally-adaptive technologies were demonstrate on COTS testbed hardware including a radiation source and sensor. This demonstration comprised the functionality of the environment sensor, the environment alert generator, high availability middleware, and high-level replication services, e.g., SCP (Self-Checking Pair) and TMR (Triple Modular Redundancy). The DM system demonstrated the capability to switch from simplex operation to SCP and TMR operation and back as the radiation level was varied.

5. TRL5 VALIDATION

The DM project successfully passed several key NMP ST8 project milestones: the TRL5 (Technology Readiness Level 5) technology validation demonstration, the Experiment Preliminary Design Review, and the NASA Non Advocate Review. Passing the TRL5 milestone qualified the DM project for advancement to flight system development status. All of the required DM software functionality was implemented and demonstrated as part of the TRL5 validation. The higher level DM middleware components including the JM, MM, JMA, FTM, FEMPI, MDSM, RS, and FCPS were designed, developed, and demonstrated by Honeywell and its University of Florida teammate during the project's Formulation Phase. These prototyped DM components are currently being transitioned to Honeywell Category 4 Flight Software.

TRL5 Overview

During the TRL5 effort, a set fault injection campaigns using the NFTAPE (Networked Fault Tolerance and Performance Evaluation) tool to validate DM technology were conducted. With NFTAPE, thousands of faults were inject into the instrumented DM TRL5 testbed, allowing us to perform fault-to-system error profiling (mapping) of the DM testbed system. The instrumented system allowed us to collect error detection and recovery coverage and latency statistics on DM system response to fault injections. These experiment results were used to populate parameters in the Availability, Reliability, and Performance Models and to demonstrate the ability of the models to predict Availability, Reliability, and Performance in different environments. Thirty-three (33) mission "application" segments based on 2DFFT, LUD (Lower-Upper Decomposition) for matrix inverse operations, and Matrix Multiply (MM) algorithms were used to exercise and demonstrate all of the fault tolerance capabilities of the DM system. Scalability to large cluster networks of 20-32 nodes was also demonstrated along with the portability of DM middleware software between PPC (Power PC) and Pentium-based processing systems.

In addition to validating the basic DM functionality, the TRL5 experiments measured the parameters needed to validate the Performance, Reliability, and Availability models. These parameters include: the detection and recovery coverage for each fault/error type in the Canonical Fault Model, the detection and recovery latencies for each fault/error type in the Canonical Fault Model, the probability that a particular fault affects the application, the number of expected mode changes for the mission, the time to effect the mode changes, the peak throughput of the CPUs in the high-performance data processing nodes, the algorithm/architecture coupling efficiency for the application, the network-level parallelization efficiency, the OS and Fault-Tolerance services overhead, and the measured execution times for the applications, with and without FPGA Co-Processor acceleration, to determine the system reliability, the system availability, the effective delivered throughput (MOPS), the effective delivered throughput density (MOPS/watt), and the effective system utilization for the mission.

TRL5 Testbed System

The COTS testbed hardware used in TRL5 is depicted in Figure 6. The TRL5 system consists of four (4) high-performance COTS data processing nodes, two (2) COTS processors to emulate the redundant radiation hardened system controllers, and redundant Gigabit Ethernet switches. Three of the data processing nodes had FPGA coprocessor accelerators. One of the data processor nodes was used to emulate a payload mass data storage element. The clock rate of the controller nodes were reduced to match the performance of the radiation-hardened controllers in the flight system. LINUX OS was used on all nodes.

Figure 6 also shows the location of the key software elements including the NFTAPE Controller which resided on the Host Computer, which also emulated the operation of the spacecraft computer, and the NFTAPE Process Managers which resided on each node. The NFTAPE Process Manager is the means for physically injecting faults in to the testbed system. The SENA board provided power control to each of the testbed nodes. This allowed the power to the nodes to be cycled or reset under spacecraft host computer control as might be done in a real space system. Finally, the testbed provided access to the outside world. This capability was useful to allow our ST8 DM teammates to run experiments remotely on the testbed hardware at Honeywell. The SENA board allowed the testbed to be reset and rebooted remotely.

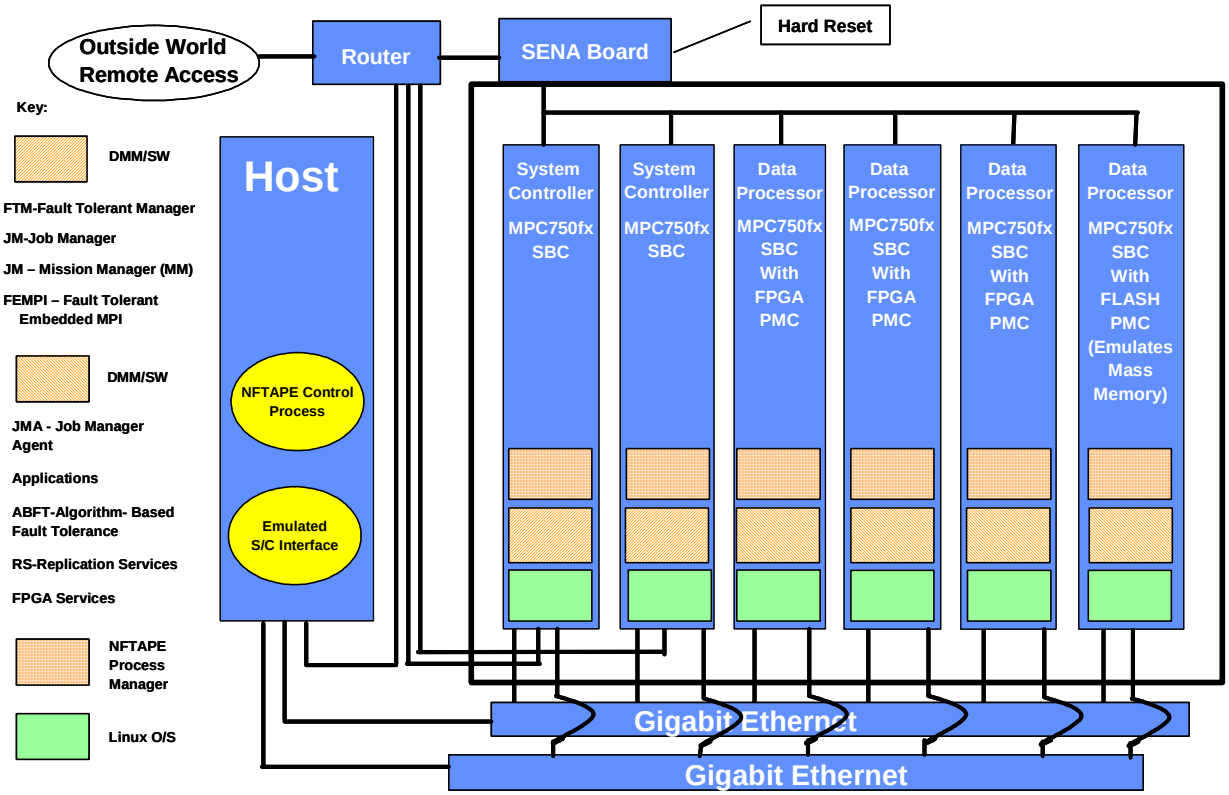


Figure 6 – TRL5 Testbed System

A top-level overview of DM software architecture stack and the partitioning and mapping onto the TRL5 system hardware is depicted in Figure 7. The DM software architecture includes the middleware layers which provide fault tolerance for the cluster and a thin isolation layer which makes porting between platforms a minimal and straightforward process. DM fault tolerant middleware includes COTS High Availability Middleware (HAM), and the DM Job Management Services (JMS), Fault Tolerance Management Services (FTMS), Fault-tolerant Embedded Message Passing Interface (FEMPI), FPGA Co-Processor Services (FCPS), Replication Services (RS), and Checkpoint and Rollback (CR) functions. The Job Management Services function consists of the Job Manager (JM) which executes on the system controller node and the Job Manager Agents (JMAs) which execute on the high-performance data processing nodes. Correspondingly, the Fault Tolerance Management Services function consists of the Fault Tolerance Manager (FTM) which executes on the system controller. In the interest of efficiency and to avoid unneeded redundancy, the Fault Tolerance Management Agents (FTMAs) which execute on the high-performance data processing nodes were absorbed into the Job Management Agents (JMA) function. The COTS HAM functions include the basic cluster management services, availability management services, replicated data base services, and data messaging services including reliable communications. The DM fault tolerant middleware components execute on top of the LINUX OS (Operating System).

Software Fault Injection Experiments

A key element of the TRL5 effort was the testing and profiling of the DM system via software fault injection (SWFI). The software fault injection experiments were performed using NFTAPE, the Networked Fault Tolerance and Performance Evaluation tool developed by the University of Illinois and Armored Computing Inc. NFTAPE supported fault injections into both kernel space and application space, into memories, system registers, the stack, the heap, data, processes, and code. The tool was used to access specific targets and perform random injections. It was used to perform non-breakpoint injections into system registers, data breakpoint injections into data including the stack and the heap, and instruction breakpoint injections into executing code. More information about the fault injection experiment can be found in Reference [1].

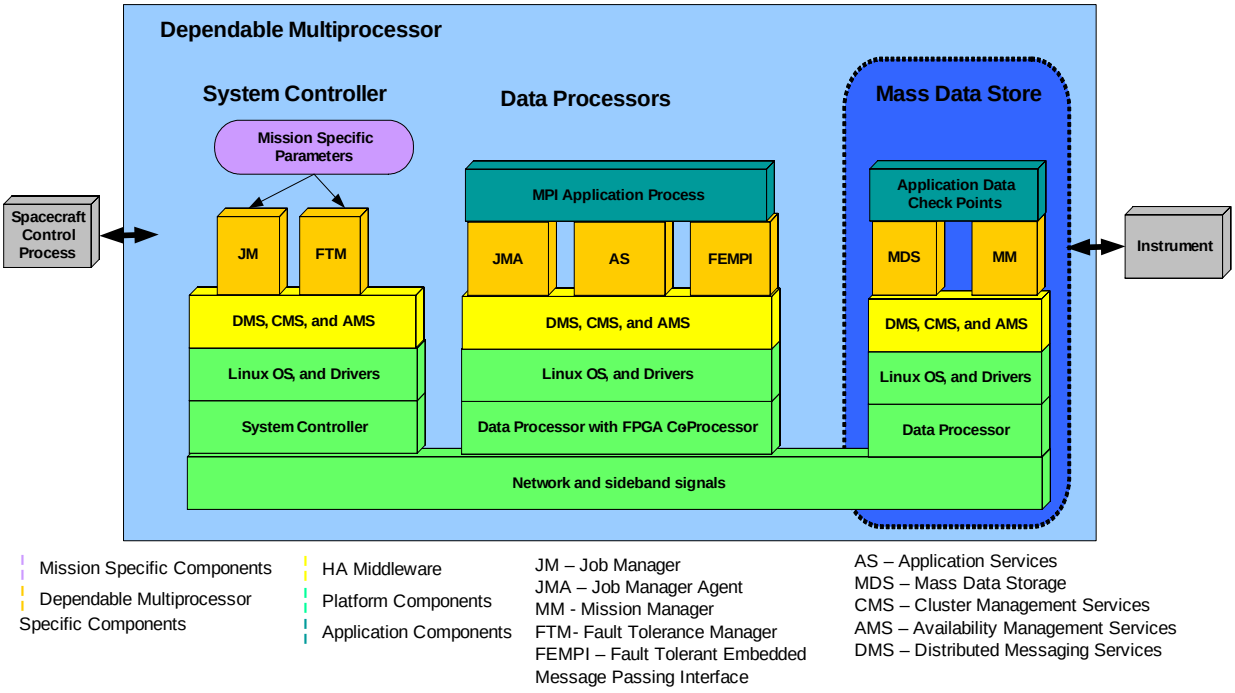


Figure 7 – Dependable Multiprocessor Middleware Components, Partitioning, and Mapping on TRL5 System

TRL 5 Demonstrations

In addition to the fault injection campaigns, a number of specific demonstrations were generated to show the breadth of DM technology capability. A summary list of the TRL5 demonstrations is shown in Table 1. Three (3) self-contained, high workload benchmark applications, the 2DFFT, the LUD, and matrix multiply applications developed for TRL5, were used assess the fault tolerance performance of the DM experiment payload in realistic application loading conditions. These application were implemented with various levels of fault tolerance protection, e.g., simplex (no fault tolerance protection), replication (physical and temporal SCP & TMR), and ABFT, to assess DM system fault tolerance performance.

6. TRL6 Validation

Following the NASA ST8 Project Confirmation Review, the DM project moved into the Implementation Phase which includes TRL6 technology validation and the fabrication and ground testing of the TRL7 flight system. The TRL6 technology validation is being performed in two stages. In the first stage, DM technology development and validation is continuing on the TRL5 testbed system. In the second stage, the DM system software will be ported to the TRL7 flight system on which all TRL6 validation and demonstration experiments will be repeated. The DM project adopted the "Test Like You Fly" philosophy. All proposed flight experiments will be run on the TRL6 systems. This includes the emulation of flight experiment payload commands and the emulation of the down-linking of DM payload health and experiment results. For the TRL6 testbed validation, the COTS-emulated System Controller will be replaced by a Honeywell Galaxy SBC, a prototype of the Rad Hard System Controller which will be used in flight. In the second stage, the DM flight system will be taken to a radiation beam test facility where one of the high-performance data processing nodes will be exposed to a particle beam for system-level tests.

7. TRL 7 Flight Validation Experiment

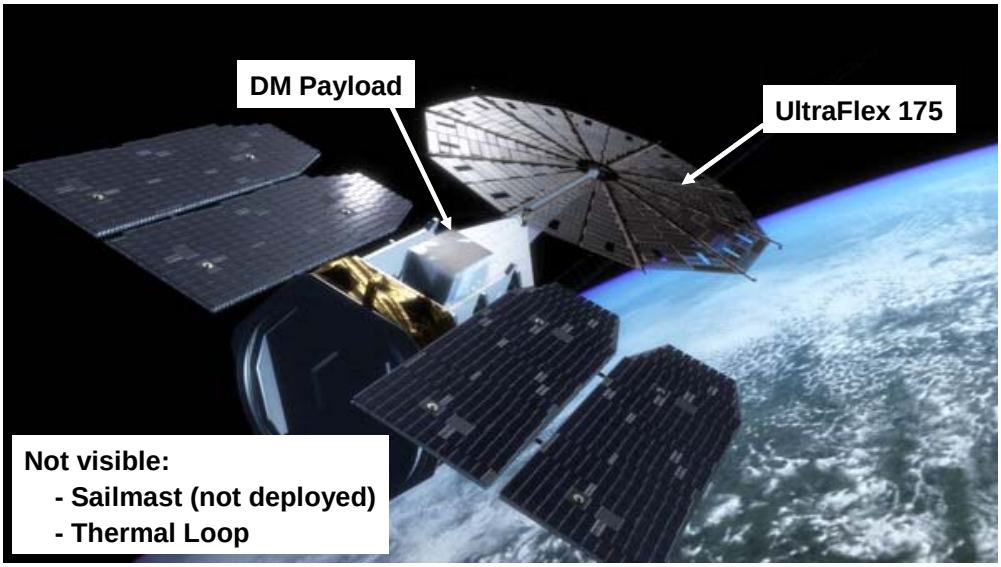
An artist's conception of the NMP Carrier spacecraft for ST8 showing the Dependable Multiprocessor payload is depicted in Figure 8. The DM (DM) experiment will share the spacecraft bus with three (3) other ST8 experiments: 1) the UltraFlex 175 deployable solar array experiment, 2) the TL (Thermal Loop) heat pipe experiment, and 3) the deployable Sailmast experiment. Orbital Sciences Corporation (OSC) is under contract to provide the ST8 spacecraft bus.

Table 1 TRL5 Demonstrations

Demo	Mission	Processing Type	Timing	Fault Detect Mode	Recovery Mode	Fault Coverage	Criteria
1	LUD	Serial	none	none	Restart (CP N/A)	AJHC	3.2
2	LUD	Serial	none	3TR	Vote and RF	AHJC+DE	3.2
3	LUD	Serial	RTD	3SR	Vote and RB	AHJC+DE	3.2
4	LUD	Serial	none	ABFT	ED1	AHJC+DE	3.2
5	LUD	Serial	none	ABFT+3TR	ED1+Vote and RF	AHJC+DE	3.2
6	LUD	Serial	RTD	ABFT+3SR	ED1+Vote and RB	AHJC+DE	3.2
7	FTMM	Serial	RTD	ABFT	EC or Restart	AHJC+DE	3.2
8	2DFFT	Serial	none	none	Restart (CP N/A)	AJHC	3.2
9	2DFFT	Serial	none	3TR	Vote and RF	AHJC+DE	3.2
10	2DFFT	Serial	RTD	3SR	Vote and RB	AHJC+DE	3.2
11	LUD-P	Parallel	none	none	RB without CP	AHJC	3.1, 4.4
12	LUD-P	Parallel	none	3TR	Vote and RF	AHJC+DE	3.1, 4.4
13	LUD-P	Parallel	RTD	3SR	Vote and RB	AHJC+DE	3.1, 4.4
14	LUD-P	Parallel	none	ABFT	ED2	AHJC+DE	4.4
15	LUD-P	Parallel	none	ABFT+3TR	ED2+Vote and RF	AHJC+DE	4.4
16	LUD-P	Parallel	none	ABFT+3SR	ED2+Vote and RB	AHJC+DE	4.4
17	2DFFT-P	Parallel	none	Rebuild	RCP	AHJC+DE	3.1, 4.4
18	2DFFT-P	Parallel	none	3TR	Vote and RF	AHJC+DE	3.1, 4.4
19	2DFFT-P	Parallel	none	3SR	Vote and RB	AHJC+DE	3.1, 4.4
20	LUD,2DFFT,LUD,2DFFT	Sequentially Serial	RTD	ABFT,2SR,3SR,3TR	RF,RB,RB,RF	AHJC+DE	3.3
21	LUD,2DFFT+2DFFT,LUD	Sequentially Distributed	RTD	ABFT,2SR+2TR,2SR	RF,RB+RF,RB	AHJC+DE	3.3
22	2DFFT+LUD-P	Distributed Serial/Parallel	none	2SR+2TR	RCP + Vote and RF	AHJC+DE	3.4, 4.2
23	2DFFT-P+LUD	Distributed Serial/Parallel	none	2TR+2SR	RCP + Vote and RF	AHJC+DE	3.4, 4.2
24	LUD,LUD	Serial	RTD	Env. Adaptable SR	Vote and RB	AHJC+DE	N/A
25	Kmeans	Parallel	none	none	Abort	none	N/A
26	LUD+System Diagnostic	Serial	RTD	Frame Scheduling	RB+Abort	Mission	3.5
27	2DFFT (Chain of 4)	Serial	RTD	3SR,2TR,3SR,2TR	Vote and RB	AHJC+DE	4.7
28	2DFFT-FPGA	Serial	none	none	RB without CP	none	5
29	2DFFT-FPGA	Serial	none	HW TMR	Vote and RB	FPGA FT	6
30	2DFFT-FPGA	Serial	none	Threaded Replication	Vote and RB	FPGA FT	6
31	2DFFT-FPGA	Master/Slave Distributed	none	HW TMR	Vote and RF	FPGA FT	6
32	LUD-P	Parallel	none	3TR	Vote and RF	Network failover	4.7
33	LUD	Serial	none	Processor Signals	Abort	AHJC+DE+CE	3.2, 4.6

CP N/A = CP not avalilable in Serial 2DFFT and LUD
AJHC = Application and JMA Hang and Crash
FBS = Frame-based scheduling
RTD = Real-time Deadline
TR = Temporal Replication
SR = Spatial Replication
CP = Check Point
RCP = Restart from CP
RF = Roll Forward
EC = Error Correct
CE = Control Error
RB = Roll Back
ED1 = Error Detect and 1 Restart
DE = Data Error
ED2 = Error Detect and Abort

The DM experiment configuration is depicted in Figure 9. The Space Segment comprises the spacecraft bus and the DM experiment payload. The Ground Segment comprises the ST8 mission ground facility, which consists of two elements, the USN (Universal Space Network) which will provide the communication link between the ground and the spacecraft, and the MOC (Mission Operations Center), which also will be provided by OSC, and the Experiment Control facility at Honeywell. Experiment command requests will be forwarded to the spacecraft through the Mission Operations Center. Experiment telemetry and data received from the spacecraft will be transmitted over an Internet link to Honeywell where data reduction and analysis will be performed.



ST8 Orbit: **- sun-synchronous**
- 955 km x 450 km @ 98.2^{o} inclination
- selected to maximize DM data collection while minimizing stress on the spacecraft bus and other experiments

Figure 8 – Artist's Conception ST8 Spacecraft and DM Payload

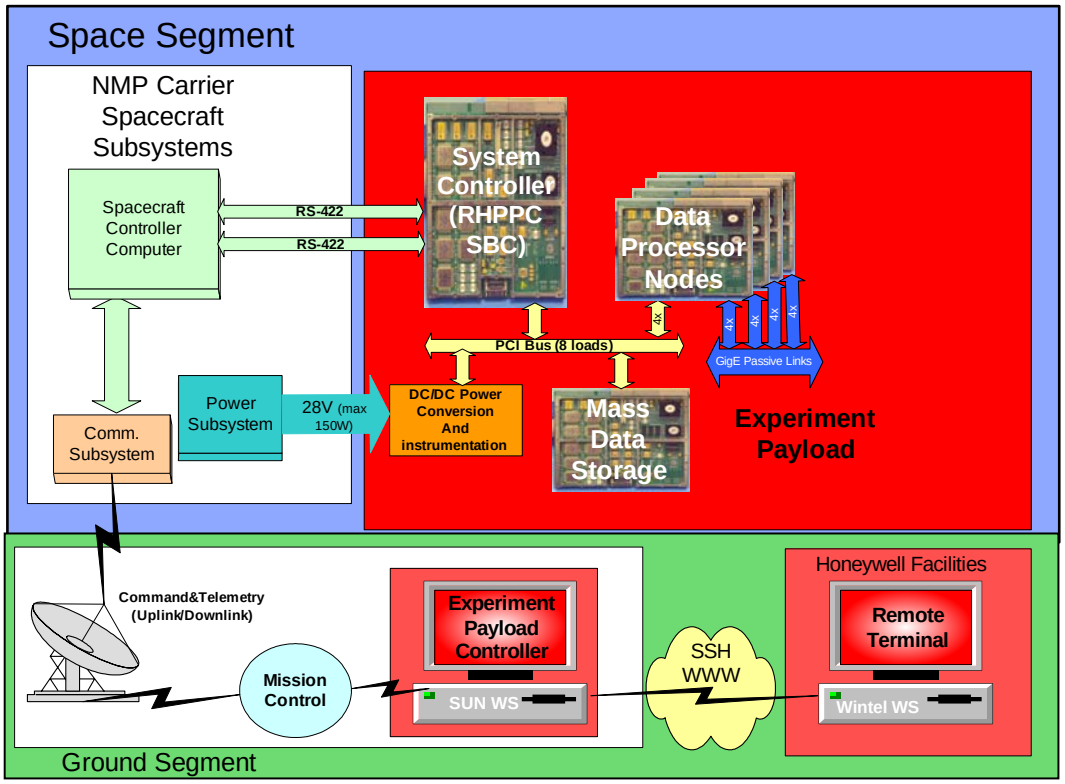


Figure 9 – Dependable Multiprocessor Experiment Configuration

The objectives of the DM flight experiment are three-fold:

1) to expose a COTS-based, high-performance processing cluster to the real space radiation environment,
2) to correlate the radiation performance of the COTS components with the environment, and
3) to assess the radiation performance of the COTS components and the DM system response in order to validate the predictive Reliability, Availability, and Performance models for the ST8 flight experiment and for future NASA missions.

The highly-inclined (98.2^{O}), elliptical ST8 mission orbit with a planned apogee of 955 km and a perigee of 450 km was selected to maximize the data collection capability for the DM experiment while minimizing stress on the spacecraft bus and the other experiments.

Except for some power-up and initialization testing, whenever the DM payload is powered on, DM operation is planned to be a free running experiment, correlating the environment and detected events with spacecraft ephemeris, and monitoring and reporting DM response. The DM experiment is planned to be run continuously for at least four of the six month ST-8 mission to maximize the amount of data collected.

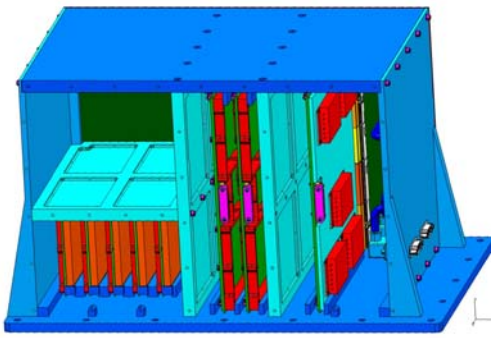
The DM flight experiment will encompass measurement of component and system parameters that can only be validated in a real space environment. Primarily, these are the component fault/error rates dues to radiation, and the system response data needed to validate the accuracy of the predictive fault/error model. The spacecraft ephemeris will be used to correlate the radiation performance of the COTS components with the orbit location. Other technology validation data, including cluster performance, error detection and recovery latencies, Operating System overhead, and fault tolerant middleware overhead, will be collected in the TRL5 and TRL6 ground-based technology validation experiments. The latter parameters do not need to be re-validated in space because they are not expected to change from the values measured during the ground-based experiments.

The DM Flight Experiment hardware is shown in Figure 10.

Figure 11 is a high level depiction of normal DM payload flight operation. Any time the DM payload is powered off, prior to being turned on, the spacecraft will turn on the warm-up heaters to ensure the DM system is at a safe temperature to power up the COTS components. After the DM payload reaches its start-up temperature, the spacecraft will turn on power to the DM payload. Upon the application of power, the DM payload will execute its normal power-on initialization sequence, starting with the System Controller and continuing through the Mass Data Storage element and the four COTS data processors. Once the DM network is established, the remaining DM System Software, primarily the DM fault tolerant middleware, will be started.

Because on-orbit operation time is so critical to the DM experiment, as soon as the DM Middleware is up and operating, the DM experiment will immediately start collecting and reporting experiment data. After power on testing, the DM Payload will turn on the spacecraft interface software followed by the DM middleware (DMM) software, the environment data collection software, and the experiment application software. Again, because on-orbit operation time is so critical to the DM experiment, the start of experiment data collection will not wait for a command to do so. The DM payload is being designed to be an autonomous, self-contained experiment. The DM System Software, the environment data collection software, and application experiment software will run continuously until commanded to turn off or until power is removed from the DM payload. In the absence of commands from the spacecraft or from the ground, the application experiment software will be set up to run pre-defined and pre-loaded sequences of experiments. In essence, the DM payload is a free-running experiment designed to collect and report experiment data as long as the DM payload is powered up.

When powered up, the DM will periodically cycle through multiple mission applications, it will capture data from SEU events as they occur, it will continuously output summary experiment data in regularly-scheduled SOH (State-of-Health) messages, and it will output additional "experiment data" associated with the sensed radiation environment, the application events, and SEU events in the periodic Experiment Data Telemetry messages. The System Controller will collect all experiment data, buffer it, and format it for transmission to the spacecraft Mission Interface Unit (MIU). In normal operation, whenever the DM payload is powered on, the spacecraft MIU will poll the DM payload once every four seconds for a State-Of-Health (SOH) message and will also poll the DM payload once every four seconds for Experiment Data Telemetry messages. Upon receipt of one of these polling messages, the DM payload will respond with a message in the requisite format. These messages will be collected by the spacecraft MIU and down-linked to the ground when the spacecraft is in view of one of the ground stations. Upon receipt on the ground, the experiment SOH and telemetry messages will be combined with spacecraft state and ephemeris information and transmitted to Honeywell for reduction and analysis.



- 1 RHPPC SBC System Controller node
- 4 COTS DP nodes
- 1 Mass Storage node
- Gigabit Ethernet interconnect
- cPCI
- ST8 S/C interface
- Utility board
- Power Supply

DM Flight Hardware

- **Dimensions**
  - **10.6 x 12.2 x 18.0 in. (26.9 x 30.9 x 45.7 cm)**
- **Weight (Mass)**
  - **~ 42 lbs (19 kg)**
- **Power**
  - **~ 100 W**

Figure 10 - DM Flight Experiment Hardware

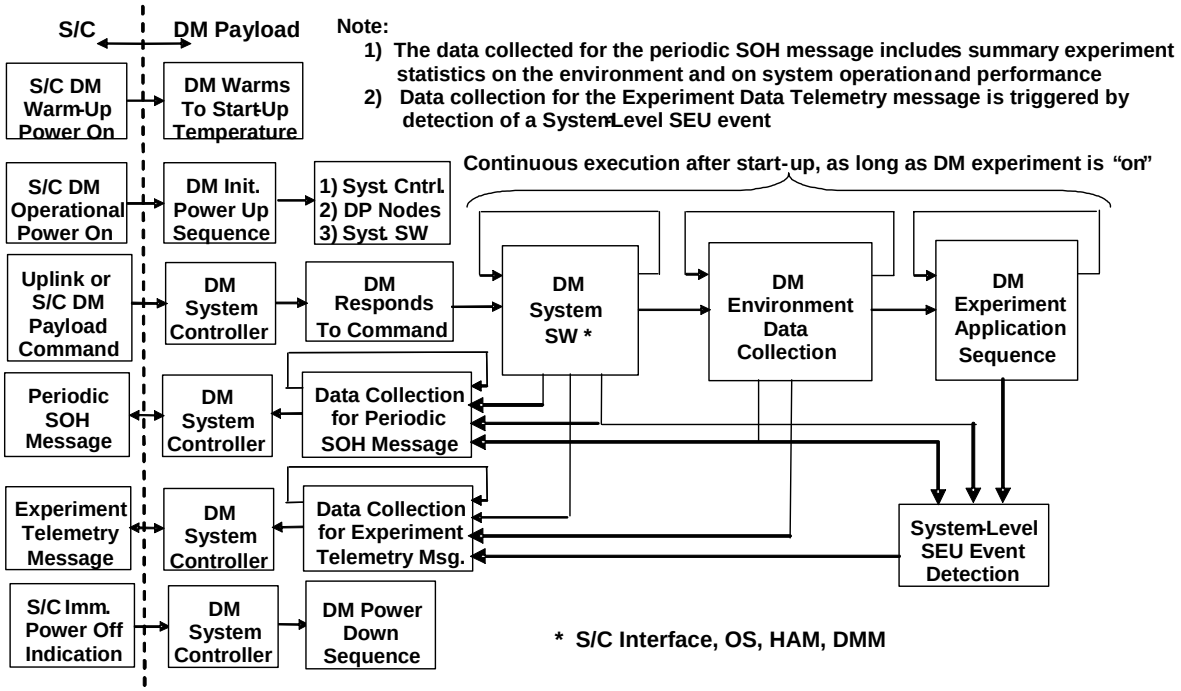


Figure 11 - Overview of DM Payload Flight Experiment Operation

There is no real science mission instrument in the DM flight experiment. Synthetic science application data will be processed continuously. The processed output will be compared with known correct output, i.e., "golden standards," to determine if an error occurred which was not detected by the DM system. The data collected will be stored and down-linked when the spacecraft is in view of one of the ground stations.

8. CURRENT STATUS

The current DM technology readiness and experiment development status and future plans are shown in Figure 12. The TRL5 technology validation and Experiment Preliminary Design Review have been completed. Preliminary radiation testing of the key microprocessor components showed they exhibited no catastrophic latch-up and sufficient number of upsets to conduct a flight validation experiment. Figure 12 also shows the currently scheduled dates for the TRL6 technology validation, the Experiment Critical Design Review, and the projected ST8 launch and operational mission period. As mentioned previously, the ST8 Project passed its Preliminary Design and Confirmation Review and is now in its Implementation Phase.

The DM project currently is preparing for the Experiment Critical Design Review (E-CDR). The design of the DM flight experiment hardware has been completed. The backplane with cPCI and Ethernet interconnects is in fabrication. The spacecraft interface software, which handles all of the command and telemetry communications

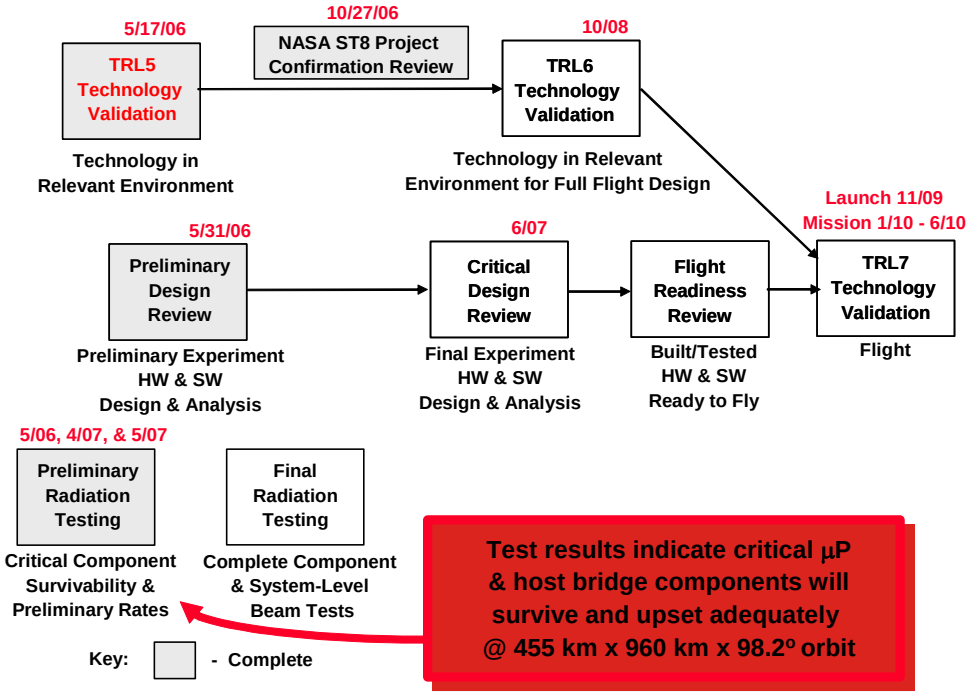


Figure 12 - DM Technology Readiness & Experiment Development Status and Future Plans

with the spacecraft MIU, has been designed and currently is being prototyped and tested. This effort includes the experiment data collection and formatting for the SOH and Experiment Data Telemetry messages.

The DM project is also working with NASA Goddard Space Flight Center to port its Evolvable Synthetic Neural System software to the DM TRL6 testbed systems and eventually to the DM flight experiment system where it will take advantage of the DM fault tolerance features including the DM middleware. The Evolvable Synthetic Neural System software has been ported to the DM TRL6 testbed where it will be subjected to the same fault injection experiments as the other science applications to demonstrate the benefits of DM technology. DM technology combined with the Goddard application is applicable to applications such as the CEV (Crew Exploration Vehicle) docking computers.

9. Summary And Conclusion

The goal of the Dependable Multiprocessor project is to provide spacecraft/payload processing capability 10x – 100x what is available today, enabling heretofore unrealizable science and autonomy. Dependable Multiprocessor technology is a key enabler for future NASA science missions including increased autonomy for remote exploration, landing support, and lunar or Martian surface rovers. Over the past few generations, COTS computer components have become more resistant to the debilitating effects of radiation. Many commercial parts can withstand many 10s of kilorads of Total Ionizing Dose (TID) and are immune to catastrophic Single Event Latchup (SEL). The primary issue preventing the deployment of a COTS-based space-borne cluster computer is their continued susceptibility to Single Event Upsets or SEUs, which cause only soft, transient errors, not permanent hardware failures. Further, the latest generation of computer electronics, SOI CMOS, has proven to be approximately an order of magnitude less susceptible to SEU than previous bulk CMOS. If DM technology allows a system to withstand a few errors per day per processor, without unduly impacting system dependability, it will be possible to fly, essentially commercial, cluster computers. Not only would this provide mission enabling performance and performance density levels, but it would significantly lower the cost of development, as standard laboratory science codes could be easily ported to these systems without the expensive and error prone process normally associated with moving complex codes from the lab to a new flight platform.

Migrating high-performance COTS processing to space is not a new idea. A key element of the DM project, which distinguishes it from previous attempts to migrate COTS to space, is that the NASA ST8 project is also providing the "ride." NASA has already issued contracts for the spacecraft and the ground facilities. Issuance of the contracts for the launch vehicle and launch support is pending. The DM experiment only needs to get through the remaining NASA and NMP gates to realize the goal of flying COTS high-performance computing in space.

While DM technology is currently being develop by NASA, primarily to support NASA science and autonomy missions including future lander and rover applications, the technology is also applicable to a wide range of DoD missions including UAVs (Unattended Airborne Vehicles, USVs (Unattended Surface Vehicles), UUVs (Unattended or Un-tethered Underwater Vehicles), Stratolites, and ORS (Operationally Responsive Space).

Finally, it should be pointed out that DM technology is not totally new technology. The ST8 DM project is the only the current incarnation of the long-held desire to fly COTS in space. Many DM system concepts are based on related technologies developed and demonstrated over past three decades on several DARPA (Defense Advanced Research Projects Agency), NASA, and DoD programs such as Space Touchstone [8], Remote Exploration and Experimentation (REE) [10, 12, 13, 14], Improved Space Computer Project/Improved Space Architecture Concept (ISCP/ISAC), Advanced Onboard Signal Processor (AOSP) [17], Advanced Architecture for Onboard Processing (AAOP), ARGOS [9], and others. All of these programs had the goal of putting high performance processing in space. None of these predecessor projects made it to space. The current NMP ST8 DM project offers the best opportunity ever to demonstrate high performance COTS processing in space.

REFERENCES

[1] Samson, John, Jr., et. al., "Technology Validation: NMP ST8 Dependable Multiprocessor Project II," *Proceedings of the 2007 IEEE Aerospace Conference*, Big Sky, MT, March 3-10, 2007.

[2] Samson, John, Jr., et. al., "High Performance Dependable Multiprocessor II," *Proceedings of the 2007 IEEE Aerospace Conference*, Big Sky, MT, March 3-10, 2007.

[3] Samson, Jr. John R., et. al.,"NMP ST8 High Performance Dependable Multiprocessor," 10th High Performance Embedded Computing Workshop, M.I.T. Lincoln Laboratory, September 20, 2006.

[4] Samson, John, J. Ramos, M. Patel, A, George, and R. Some, "Technology Validation: NMP ST8 Dependable Multiprocessor Project," *Proceedings of the 2006 IEEE Aerospace Conference*, Big Sky, MT, March 4-11, 2006.

[5] Ramos, Jeremy, J. Samson, M. Patel, A, George, and R. Some, "High Performance, Dependable Multi-processor," *Proceedings of the 2006 IEEE Aerospace Conference*, Big Sky, MT, March 4-11, 2006.

[6] Samson, Jr. John R., J. Ramos, A. George, M. Patel, and R. Some, "Environmentally-Adaptive Fault Tolerant Computing (EAFTC)," 9th High Performance Embedded Computing Workshop, M.I.T. Lincoln Laboratory, September 22, 2005.

[7] Ramos, Jeremy, and D. Brenner, "Environmentally-Adaptive Fault Tolerant Computing (EAFTC): An Enabling Technology for COTS based Space Computing ," *Proceedings of the 2004 IEEE Aerospace Conference*, Big Sky, MT, March 2004.

[8] Samson, Jr. John R., "Migrating High Performance Computing to Space," 7th High Performance Embedded Computing Workshop, M.I.T. Lincoln Laboratory, September 22, 2003.

[9] Samson, Jr., John R., "Space Touchstone Experimental Program (STEP) – Final Report 002AD," January 15, 1996.

[10] Lovellette, Michael, and K. Wood, "Strategies for Fault-Tolerant, Space-Based Computing: Lessons Learned for the ARGOS Testbed," *Proceedings of the 2002 Aerospace Conference,"* Big Sky, MT, March 9-16, 2002.

[11] Samson, John R., and C. Markiewicz, "Adaptive Resource Management (ARM) Middleware and System Architecture – the Path for Using COTS in Space," *Proceedings of the 2000 IEEE Aerospace Conference*, Big Sky, MT, March 8-15, 2000.

[12] Samson, Jr., John R., L. Dela Torre, J. Ring, and T. Stottlar, "A Comparison of Algorithm-Based Fault Tolerance and Traditional Redundant Self-Checking for SEU Mitigation," *Proceedings of the 20th Digital Avionics Systems Conference*, Daytona Beach, Florida, 18 October 2001.

[13] Karapetian, Arbi, R. Some, and J. Behan, "Radiation Fault Modeling and Fault Rate Estimation for a COTS Based Space-borne Computer," *Proceedings of the 2002 IEEE Aerospace Conference*, Big Sky, MT, March 2002.

[14] Some, Raphael, W. Kim, G. Khanoyan, and L. Callum, "Fault Injection Experiment Results in Space Borne Parallel Application Programs," *Proceedings of the 2002 IEEE Aerospace Conference*, Big Sky, MT, March 9-16, 2002.

[15] Some, Raphael, J. Behan, G. Khanoyan, L. Callum, and A. Agrawal, "Fault-Tolerant Systems Design Estimating Cache Contents and Usage," *Proceedings of the 2002 IEEE Aerospace Conference*, Big Sky, MT, March 2002.

[16] Samson, Jr., John, "SEUs from a System Perspective," Single Event Upsets in Future Computing Systems Workshop, Pasadena, CA, May 20, 2003.

[17] Prado, Ed, J. R. Samson, Jr., and D. Spina. "The COTS Conundrum," *Proceedings of the 2000 IEEE Aerospace Conference*, Big Sky, MT, March 9-15, 2003.

[18] Samson, Jr. John R., "The Advanced Onboard Signal Processor - A Validated Concept," DARPA 9th Strategic Space Symposium, Monterey, CA. October 1983.

ACKNOWLEDGEMENTS

The authors would like to thank the following people and organizations for their contributions to the Dependable Multiprocessor effort: Sherry Akins, Dr. Mathew Clark, and Lee Hoffmann of Honeywell Aerospace, Defense & Space; a team of researchers in the High-performance Computing and Simulation (HCS) Research Laboratory at University of Florida led by Dr. Alan George. Members of the team at UF include: Dr. Ian Troxel, Dr. Raj Subrmaniyan, John Curreri, Mike Fisher, Grzegorz Cieslewski, Adam Jacobs, and James Greco; Brian Heigl, Paul Arons, Gavin Kavanaugh, and Mike Nitso, from GoAhead Software, Inc. Other members of the team are Dr. Ravishankar Iyer, Weining Guk, and Tam Pham from the University of Illinois and Armored Computing Inc, and the ST8 DM Technology Review Board, Jack Stocky (JPL), Dan Katz (LSU/JPL), Coy Kouba (NASA JSC), Lee Mendoza (Aerospace Corp.), Roger Lee (JPL), and Dave Rennels (UCLA/JPL).

The research described in this paper was carried out under the auspices of the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration. The Dependable Multiprocessor effort is funded under NASA NMP ST-8 contract NMO-710209.

BIOGRAPHIES

John R. Samson, Jr. *is a Principal Engineering Fellow with Honeywell Aerospace in Clearwater, Florida. He received his Bachelor of Science degree from the Illinois Institute of Technology, his Master of Science degree and the Degree of Electrical Engineer from the Massachusetts Institute of Technology, and his Ph.D. in Engineering Science with Specialization in Computer Science from the University of South Florida. During his 36-year career, John has worked at M.I.T. Lincoln Laboratory, Raytheon Company Equipment Division, and Honeywell Space Systems. His work has encompassed multiple facets of ground, airborne and space-based surveillance system applications. He has spent most of his career developing onboard processors, onboard processing architectures, and onboard processing systems for real-time and mission critical applications. He was Principal Investigator for a pioneering study investigating the feasibility of migrating high-performance COTS processing to space, a predecessor of the work described in this paper. Dr. Samson is the Principal Investigator for the Dependable Multiprocessor project. He is an Associate Fellow of the AIAA and a Senior Member of the IEEE.*

Minesh I. Patel is a systems and software architect and consultant with Tandel Systems in Clearwater, Florida. He received his BSEE and BSCpE in electrical and computer engineering and his MSCpE and Ph.D. in Computer Science and Engineering from the University of South Florida. His research and technical interests include software and system fault tolerance, artificial intelligence and machine learning, embedded and real-time systems and high-performance, parallel and distributed computing. Dr. Patel is Lead Software Architect for the Dependable Multiprocessor project.

Alan D. George *is Professor of Electrical and Computer Engineering at the University of Florida, where he serves as Director and Founder of the High-performance Computing and Simulation (HCS) Research Laboratory. He received the B.S. degree in Computer Science and the M.S. in Electrical and Computer Engineering from the University of Central Florida, and the Ph.D. in Computer Science from the Florida State University. Dr. George's research interests focus on high-performance architectures, networks, services, and systems for parallel, reconfigurable, distributed, and fault-tolerant computing. He is a senior member of IEEE and SCS, and can be reached by e-mail at george@hcs.ufl.edu. Dr. George is the Principal Investigator for the Dependable Multiprocessor software research and development effort at the University of Florida.*

Raphael Some *is a program technologist at JPL for the New Millennium Program. He has served as Contract Technical Manager and Leader of the Technology Review Board for the Dependable Multiprocessor project. Prior to his involvement with the NMP ST8 project, Mr. Some was the Chief Engineer for the Remote Exploration and Experimentation Project at the Jet Propulsion Laboratory. Previously, at JPL, he formulated and managed the Smart Sensor project. His experience prior to JPL includes the development of fault tolerant space based supercomputers as well as a variety of avionics and signal processing systems for both commercial and military applications. He holds a BSEE from Rutgers University.*

Zbigniew Kalbarczyk *is a research professor at the Center for Reliable and High-Performance Computing in the Coordinated Science Laboratory of the University of Illinois at Urbana-Champaign. His research interests are in the area of reliable and secure networked systems. Currently he is a lead researcher on the project to develop high availability and security software infrastructure and configurable hardware architecture to provide low-overhead application-aware error detection, masking of security vulnerabilities, and recovery. His research involves also development of automated techniques for validation and benchmarking of dependable and secure computing systems. Kalbarczyk received a Ph.D. in Computer Science from the Bulgarian Academy of Sciences. He is a member of the IEEE, the IEEE Computer Society, and IFIP Working Group 10.4 on Dependable Computing and Fault Tolerance.*

Vikas Aggarwal *is a systems engineer with Tandel Systems in Clearwater, Florida. He received his B.Tech. degree in Electronics and Communications Engineering from G.G.S. Indraprastha University, Delhi, India. He received his MS degree in Electrical and Computer Engineering from University of Florida. His research interests include reconfigurable and embedded computing and system fault-tolerance, high-performance, parallel and distributed computing.*

David Lupia *is a Senior Systems Engineer at Honeywell Aerospace, Defense and Space Systems in Clearwater, Florida. He received his Bachelor of Science degree in Electrical Engineering from Ohio University. From 1993-1995, he was a Stocker Research Fellow in Ohio University's Master's Program. He has over 10 years of experience in the design and development of space and military electronics systems, with his core expertise in Digital Signal Processing, Communication Systems, and Reconfigurable Computing. His primary background is in modeling and simulation. He was responsible for model development and fault injection testing during the NMP ST8 DM project TRL5 effort.*

Gary Gardner *is a Program Manager with Honeywell Aerospace, Defense and Space Systems in Clearwater, Florida. He received his Bachelor of Science degree from the University of Kansas, and his Master of Science degree. in Computer Engineering from the University of South Florida. He has more than 30 years experience developing onboard processing systems for space applications. This experience includes work on SSMEC (Space Shuttle Main Engine Controller). Recently, he was the Product Development Manager for the MuSICA (Multiconfigurable Spacecraft Interface Assembly) and Technology Development Manager for several radiation hardened component developments including RHrFPGA (Radiation Hardened reconfigurable FPGA) and MIPS support chip sets. He is currently the Program Manager for the NMP ST8 Dependable Multiprocessor Project.*

Paul B. Davis *is a Principal Systems Engineer at Tandel Systems in Clearwater, Florida. He earned a Bachelor of Science degree in Electrical Engineering from the University of Tennessee graduating Magna Cum Laude. In 1993, Mr. Davis received an Outstanding Engineer Award in for his exceptional work on the Follow-on Early Warning System as part of the Strategic Defense Initiative program. His research and technical interests include embedded real-time systems, high-performance computing systems and operating systems. He was responsible for the TRL5 DM testbed system including integration of the NFTAPE fault injection tool and for testing and demonstration of the DM system software. He also conducted studies and trades of techniques to make commercial operating systems more robust for use in the space environment.*