Processor Building Blocks for Space Applications

John Holland, Northrop Grumman Corporation,
Linthicum, U.S.A., John.Holland@ngc.com

Max Corbin, Northrop Grumman Corporation,
Linthicum, U.S.A., Max.Corbin@ngc.com

Jeremy W. Horner, Northrop Grumman Corporation,
Linthicum, U.S.A., Jeremy.Horner@ngc.com

Eliot Glaser, Northrop Grumman Corporation,
Linthicum, U.S.A., Eliot.Glaser@ngc.com

Gary Petrosky, Northrop Grumman Corporation
Linthicum, U.S.A., Gary.Petrosky@ngc.com

***Abstract* — Designing Processors for space applications presents unique challenges. Challenging form factors, environmental requirements, survivability demands, thermal constraints, space-qualified material restrictions, and size, weight, and power goals all contribute to the complexity of the design trade space. Testing and screening of parts and components by commercial vendors is limited by market demands. This paper will address these problems, look at future trends, and explore open architecture solutions that maximize reuse of processor architectures and components and minimize one-time or non-recurring costs for digital front-end signal and data processing.**

Keywords — Front end processing; space; radiation; SERDES; LVDS; protocols; AXI4; portability; redundancy; digital beamforming

III. Introduction

Processors designed for space applications perform valuable functions for a space system. General purpose processors provide command and control of space vehicles and payloads. Front end digital signal processors provide the processing of sensor signals and the important task of data reduction to meet data storage limits of on-board data recorders and data bandwidth limits of satellite-to-ground communications links.

This paper discusses constraints on the design of processors for space applications, some of the options available for processor design, and future trends. Section II of this paper addresses the challenges of the space environment and the challenging form factors that are required. Section III discusses available components, physical interfaces, and interface protocols. Section IV looks at future trends and gaps in the component and knowledge base.

IV. Challenging Form Factors for the Space Environment

The space environment presents a difficult set of processing design challenges. This paper will divide these into four areas: radiation, thermal, packaging, and SWAP (size, weight and power). This section summarizes discussions in a previous paper [1].

A. Radiation

The two major radiation considerations are total dose radiation and single event effects (SEE). Total dose radiation is the total ionizing dose (TID) over the service life of the component or unit. The level of radiation exposure depends on the mission lifetime and profile, along with the orbital altitude and inclination. Radiation-qualified device capabilities range from radiation tolerant ($< 50 \times 10^3$ rads) to radiation hardened ($> 1 \times 10^5$ rads) [2].

Single event effects describe the effects of high-energy particles on an integrated circuit. These effects include single event upset (SEU), single event transient (SET), single event latch-up (SEL) and single event functional interrupt (SEFI). SEL includes both non-destructive and destructive latch-up. With decreasing integrated circuit feature size, TID tolerance increases but SEE rates also increase.

B. Thermal

The vacuum of the space environment limits the component cooling path to conduction. Conduction can be by heat sink or by liquid cooling. Heat radiators require a view of cold, deep space in order to radiate heat. Thermal cycles increase the stress on component interfaces such as solder joints so it is important to minimize the number and range of thermal cycles.

C. Packaging

Challenging form factors arise from limited and customized volumes often needed to fit into platforms and launch vehicles. Mechanical design must consider the thermal properties of materials as well as thermal compatibility between materials, especially when analyzing interconnections such as solder joints. Materials with mechanical interfaces must have compatible coefficients of thermal expansion (CTE).

D. SWAP

Size, weight, and power are critical to space processing design. Launch weight is expensive at greater than $10K-20K/kg (in FY00) [2]. The power load of the electronics determines the size and weight of the power system, which is approximately 25-30% of satellite weight.

978-1-4799-6233-4/14/$31.00 ©2014 IEEE

V. SPACE PROCESSING BUILDING BLOCKS

A. Components

Typical processing components such as memories, FPGAs, ASICs, and interfaces are generally available in space qualified devices, although usually in older technologies.

Memories for space processors require a trade between size and radiation susceptibility. Radiation hardened SRAM and EEPROM are available. Commercial DRAM and flash NVM are denser and faster but require SEE mitigation. Some SEE mitigation techniques are error detection and correction (EDAC), memory scrubbing, and power supply control to limit current or correct latch-up.

FPGAs are available in anti-fuse, flash-based, or RAM-based technologies. Anti-fuse FPGAs have the advantage of radiation hardness or tolerance and built-in SEE mitigation, but are one-time programmable. Flash-based FPGAs offer reprogrammability but have a low tolerance of TID. RAM-based FPGAs are reprogrammable, faster, and available with single event latch-up mitigation, but require SEU mitigation techniques for configuration memory and internal memory. RAM-based FPGAs also require a NVM to store the configuration which introduces additional SEU concerns.

Radiation-hardened fabrication processes are available for ASICs, but non-radiation-hardened processes are more numerous, less expensive, and have more advanced feature sizes. Radiation-hardened feature sizes are limited to 0.15 micron while non-radiation-hardened processes are available at 0.020 micron and below.

Device screening (or up-screening) is an option for devices fabricated using non-radiation-hardened processes. Screening can be for TID or from SEE tolerance. Figure 1 shows examples of SEE testing.

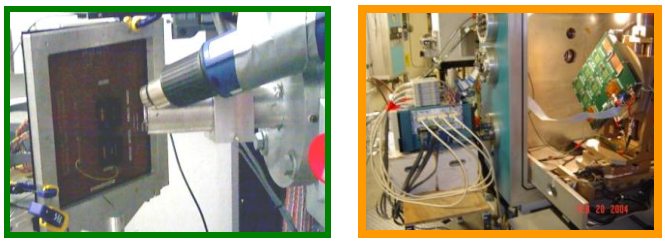

Fig. 1. Device screening, SEE testing

B. Interfaces

Space qualified devices support some of the popular interfaces standards such as SERDES and LVDS.

1) SERDES interfaces

The most efficient means of transmitting large amounts of data through the processor is by using high speed serializer/deserializer (SERDES) interfaces. The transmit (TX) functional block encodes data and embeds a clock prior to transmission over differential media in such a manner that the receiver (RX) block can reliably recover the clock and decode the data. The encoding scheme also allows for the insertion of special characters within the transmission, sometimes called K-characters, which provide additional information or help maintain transmission coherency and synchronization. Examples of these K-characters are idle, clock recovery, start of frame, and end of frame. These SERDES blocks are hard IP cores embedded in many of today's FPGAs and can achieve data rates higher than 3 Gbps for space-qualifed interfaces. Because of improved circuit board materials and routing techniques, complex processors can have hundreds of these SERDES channels providing high throughput rates.

2) LVDS interfaces

Low voltage differential signaling (LVDS) also is a popular communications method for today's space grade processors. Whether it involves communicating to high speed ADCs or DACs, DDR memories, or controlling processor functionality, LVDS transmission provides a low-power means for data transfer at speeds of 1+ Gbps rates.

C. Protocols for External, Internal and Intra-device Interfaces

1) External interface: Spacewire protocol

SpaceWire is a robust standard for data and communication interfaces that is well-suited to spaceflight applications. It is a high speed full duplex serial interface that uses a simple protocol, resulting in low resource and power requirements. SpaceWire has variable data rates and packet sizes. At its simplest level SpaceWire corresponds to the Physical & Data Link layers of the OSI model, but it includes capabilities for higher level protocols and routing.

A SpaceWire interface can implement protocols which correspond to the higher levels of the Open Systems Interconnection (OSI) model. Use of standard protocols reduces design time and increases interoperability with third party hardware and software. SpaceWire also includes the capability to incorporate user-defined protocols.

2) Internal interface: modified Fibrechannel protocol

A processor's size, flexibility, and processing power may require it to transmit large amounts of data through the processing blocks at a high sample rate or data bandwidth. A network of SERDES interconnects can transfer data from module to module. Space qualified SERDES channels run at a 3.125Gbps signaling rate with a data throughput rate of 2.5Gbps.

Tailoring of high-speed SERDES protocols can reduce the device resources and increase interface efficiency for a space application. An example of this is a reduced version of the Fibrechannel protocol standard. When channels are uni-directional and point-to-point, a simple protocol and architecture that minimizes FPGA resources is an efficient and logical approach. A subset of 8B/10B encoded control

character set can support byte alignment and clock correction as well as Start-of-Frame and End-of-Frame delimiters. Other protocol and device control options are cyclic redundancy checks (CRC), resets, power down controls, and equalization control. Optional status bits determine the quality of the channel. The user must ensure that with the encoding scheme, the special K- (or control) characters, and CRC that the data throughput does not exceed the channel bandwidth.

3) Intra-device interface: AXI4 protocol

The increasing resources available in ASIC and FPGA devices enable continued opportunities to improve the capabilities of systems, including space systems. However, compressing additional capability in a smaller area poses increasing challenges. Consideration should be given to how expanded requirements affect schedule, reusability, maintainability, testability, and deployment of future designs.

One approach to addressing these issues is to segregate design functions and limit inter-process communication to industry standard protocols. Additional design consideration should be given to making sub-functions as generic as possible. Simplified assembly of centralized applications follows as a natural course of such practices.

These concepts can be applied to processor design as shown in Figure 2. Challenges with maintainability and usability of an existing design can be addressed by reworking data and control pathways. The processing elements are isolated and wrapped in a packet architecture based on the ARM AXI4 streaming protocol. The routing capabilities of the protocol can be exploited to create a NoC (Network-on-Chip) system where each processing node has identical input and output buses. The final implementation greatly simplifies the overall design, standardizes the processing interfaces, and increases potential reusability as shown in Figure 2.

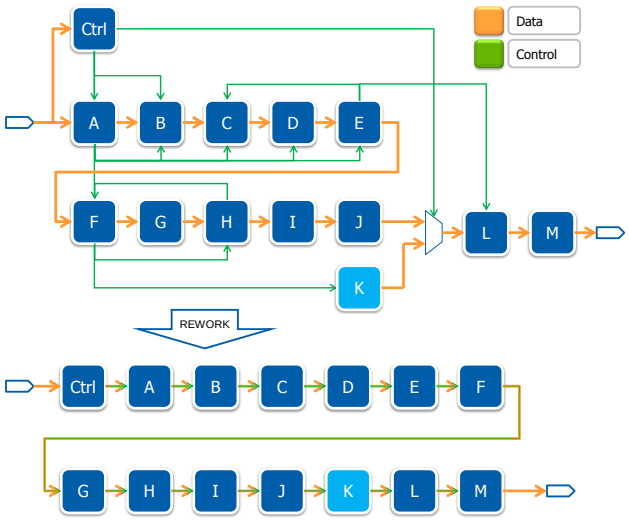

Fig. 2. AXI4 network conversion

A networked design based on standard protocols allows further simplification in deploying a design to multiple target platforms. This allows processing targeted to space applications to be developed on lower-cost commercial processor hardware. Only the input and output functional blocks need to be adapted for each migration, resulting in a significant reduction of overall effort. The use of the network also simplifies test bench debug and verification as only one protocol is in use.

D. Using Redundancy to Improve Reliability

Redundancy in a system allows the system to adapt to component, assembly or subsystem failures. In a block redundant system, redundant or duplicate components are in place but unused or unpowered unless a fault or failure is detected. The system controller commands a switch to the redundant component. Simple redundancy can be extended to a slice approach, with multiple copies of a component and the system controller choosing the M of N necessary components for system operation. M can be a variable that depends on system bandwidth requirements during a particular time period. Another approach keeps all N components powered on and in use, and all components processing data. The components then vote, with the M majority determining the processing result. Figure 3 shows a block diagram of these approaches.

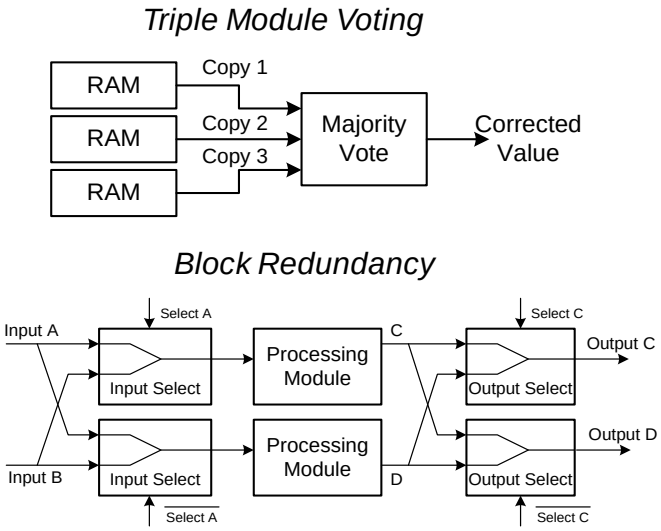

Fig. 3. Triple module and block redundancy

E. Example Architecture

This paper will use a digital beamforming architecture as the example architecture to examine the principles described in the previous sections.

Digital beamforming (Figure 4) forms one or more beams from sub-apertures of an antenna. In the figure the incoming wave front is received by each of the channels of the antenna, down-converted and A/D sampled. Digital beamforming uses time delays and/or frequency-specific phase shifts to steer the input channels so that each senses the same frequency and phase of the wave front.

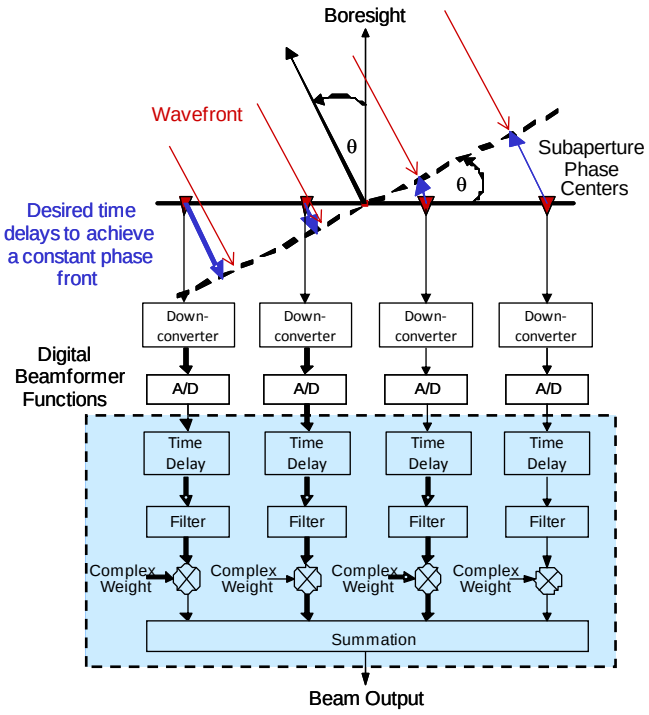

Fig. 4. Notional digital beamforming front end processing architecture

Key parameters for digital beamforming architectures are: input channels and sample rates, output beams and data bandwidth, as well as cost, size, weight, power, and environmental requirements.

F. Example Architecture Implementation

The building blocks described in previous sections can be used to implement the example architecture. Figure 5 shows an example implementation.

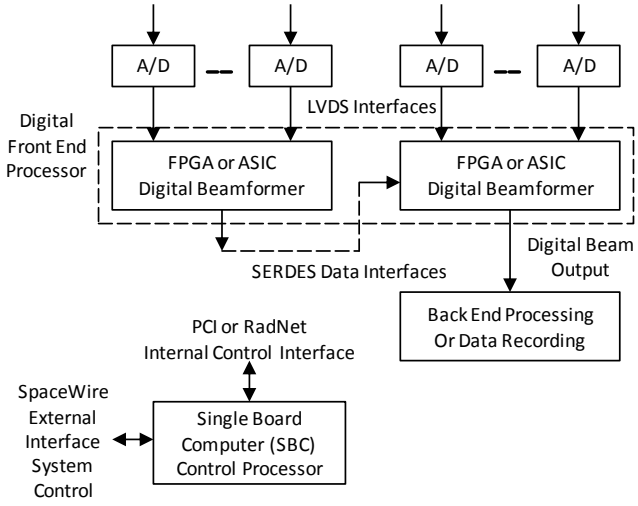

Fig 5. Example implementation of digital beamforming front end processing architecture

The A/D converters receive the sensors signals and transmit digital samples to the front end processor on LVDS interfaces. Digital front end processing is implemented in FPGAs or ASICs and includes time delays, filters and the digital beamforming. The digital beamformer is cascadable and scalable with high-speed SERDES data channels as the interface between processing devices and as the output interface to the back end processing. SpaceWire provides the external control interface to a single board computer (SBC). The SBC uses an internal backplane interface (CompactPCI is a possible choice) to distribute controls and beamforming weights to the processing elements.

G. Intellectual Property and Principles of Portability

1) Intellectual Property

Intellectual Property (IP) represents a cornerstone among the building blocks within a space system. The IP used to implement the system algorithms contributes significantly to the value provided to the customer and can be a key discriminator between the system and systems offered by competitors. IP comes in different sizes and can consist of a single operation (e.g. complex add) or an entire algorithm (e.g. digital beamformer).

The IP's value to the company is a function of the frequency of its use and the amount of effort that must be expended to adapt it for use in the target system. This value can be maximized by designing IP for reuse and by following principles of portability.

Each implementation option for a system offers a unique set of capabilities and limitations (shown in Table 1). While a previous paper [3] has discussed a process for achieving portability of algorithms across different hardware platforms, it is often possible to achieve portability of the IP implementation itself across different hardware platforms despite the diversity of those target platforms.

This is accomplished by following several key principles of portability during the design, implementation, and the System Integration & Test phases of IP development.

2) Principles of Portability: Design

a) Modular Architecture

Create a modular architecture for a design by identifying

	Custom: ASIC, Board	Custom: FPGA, Board	COTS: FPGA Board	COTS: Processor Board
I/O Flexibility	Highest	High	Medium	Low
Capability	Highest	Medium	Medium	Low
Processing Efficiency	Highest	Medium	Medium	Lowest
Programmability	None	Medium	Medium	High
Complexity	High	Medium	Medium	Low
NRE Cost	Highest	Medium	Low	Lowest
Power	Lowest	High	High	Medium

Most desirable
Desirable
Least desirable

Table 1: Hardware Platform Capabilities and Limitations

the major algorithms or functions that must be performed to accomplish the overall set of requirements and understanding how they will need to interact with one another. A processing module can then be created for each of these algorithms or functions using well-defined interfaces and interactions with other processing modules. Care should be taken to ensure that each processing module is not overly simple or overly complex.

b) Anticipate needs at peripheral interfaces: clock domain crossing, protocols

It can be expected that the primary focus of an effort to adapt IP for use in a target system will be on how to connect the system to the IP's peripheral interfaces. Many of the challenges and requirements that will be encountered can be and should be anticipated. For example, including dual-clock FIFOs on all peripheral interfaces and using well-defined protocols will help to minimize the amount effort required to host IP in a new system.

c) Use Standard Interfaces for Modularity/Testability

Wherever it is practical, industry standard interfaces should be used for external and internal interfaces in an IP design. This helps to increase the modularity and testability of a design.

d) Decentralize Control Logic

The control plane architecture must also be considered carefully. Where practical, control logic should be decentralized so that the design's modularity extends beyond the algorithm or function and includes the control logic for that algorithm or function. If the control plane logic is not decentralized, then the amount of effort to move the IP to a different platform can increase when the control plane architectures differ greatly.

3) Principles of Portability: Implementation

a) Use VHDL/Verilog; avoid device-specific tools

It should be expected that the IP will continue to be used for a decade or longer. In order to ensure that the design will have the required supportability, the design should be implemented using standard VHDL/Verilog. This will avoid many issues associated with tool versions or even Operating Systems becoming obsolete that could otherwise be required to support a design.

b) Use inferred logic; minimize device-specific libraries

The use of inferred logic is another practice which maximizes the portability of the code across platforms. There are some cases where device-specific components must be used, but each of these cases creates a barrier that must be overcome when retargeting the IP to a new device.

c) Use parameterization to write flexible code

Adding parameters that control bitwidths and other aspects of the design can dramatically increase the utility of an IP design without, necessarily, requiring a significant increase in development effort.

d) Avoid hard-coded numbers and indices.

In general, the use of hard-coded numbers and indices should be minimized. Instead, it can be very effective to "show your work" by using a combination of constants and calculations that, in turn, are based on other constants. This can, for example, show the bit-growth that takes place throughout the design.

4) Principles of Portability: Integration & Test

a) Develop and Prototype IP on inexpensive COTS platforms

The portability of the IP can now be leveraged to great advantage during the development of the space system. IP can be developed and prototyped on inexpensive COTS platforms (e.g. evaluation cards) before it is ported to space-qualified components.

An example progression of hardware platforms used to host the IP is as follows: inexpensive evaluation cards, Engineering Development Units (EDUs), and then to space-qualified parts.

This approach helps to ensure that the IP design has reached the desired level of maturity long before the space-qualified parts have become available. Furthermore, the accelerated availability of hardware platforms can enable early System Integration & Test activity in order to reduce integration risk.

b) Documentation: Interfaces, HW/SW interaction, sequence diagrams, etc.

Finally, it is important to thoroughly document the IP. This should include the internal and external interfaces, and details regarding how the IP interacts with other hardware and software. Whenever there is a complex interaction taking place, it can be effective to use sequence diagrams in order to communicate the order and dependencies of the interactions. Having thorough and effective documentation helps to minimize the effort necessary to adapt IP for use on a new target platform.

VI. Future Trends and Study

As processing blocks have become denser and more specialized, commercial availability has been decreasing. Trends in commercial many-core processors such as higher power, localized cooling, and less emphasis on maximizing processing efficiency make it difficult to transition advanced processing technology to space qualification. Options for future designs include:

1) Up-screen strategically selected devices for use in space applications. Smaller feature sizes of new technologies provide increased TID tolerance. The up-screening process and testing identifies SEE vulnerabilities, then SEE mitigation can be incorporated into the processor design.

2) Substantially increase the use of redundancy and SEE mitigation techniques. Redundancy at the subsystem, module, and component levels increases reliability. Techniques for instant resource switching without data loss would allow the use of less SEE-tolerant

technologies. Multi-core, multi-processor, and slice architectures can provide M of N processor redundancy.

3) Develop heterogeneous architectures that use ASICs where possible and existing space-qualified FPGAs where reprogrammability is a must. Robust processing architectures also allow the use of lower-cost, reprogrammable devices for early system development, then portability to an ASIC or space-qualified FPGA for the final flight design.

4) Reduce package sizes by relying more on high-speed serial (or SERDES) interfaces rather than large parallel buses. Space-qualified SERDES interfaces are available that can reduce interface widths, package pin counts and can offer redundant interfaces for increased reliability. If the overall package size can be dramatically reduced, then a higher device density can be achieved and therefore a greater number of programmable gates per board.

Challenges will increase with the future trends of smaller feature sizes and lower operating voltages:

1) SEE susceptibility will increase, requiring increased efforts on mitigation. Highly integrated Processor building blocks will be more susceptible to SEFI.

2) Existing DDR and Flash memory technologies are approaching density limits to be useful in space.

3) Advanced memory technologies look promising for use in space, but much development remains to achieve the required density [4]:

   - Phase change memory reliability can be negatively impacted by scaling reduction [5]
   - Magnetic RAM (MRAM) has demonstrated very high endurance, but scaling may be limited
   - Resistive memory has the potential to replace SRAM, DRAM, and Flash in space Applications [6]

4) FPGA and ASIC design tools already push the limits of available computer technology. Adding robust radiation mitigation features to designs only stress the tools further. Increased use of floor planning and compiled cores will be needed.

Opportunities exist where academic study can contribute to increasing the availability of options for future processing designs for space including:

1) Study redundant architectures that efficiently map to multi-core or multi-slice devices. A candidate architecture must isolate redundant paths from one another while providing sufficient routing resources to avoid routing congestion or I/O bottlenecks.

2) Explore the use of fault-tolerant configuration memory for FPGA fabrics. Limiting mitigation to the configuration memory reduces the number of layers/masks that would be impacted while still providing a meaningful increase in reliability.

3) Create an architecture consisting of an adaptive network of processing blocks which can detect and route around failures that occur during operation.

4) Develop models that predict reliability as a function of the average number of logic gates per redundancy stage. Determine whether it is more advantageous to perform TMR voting on intermediate products or on final products. This informs architectural development by providing a target size for each redundant block.

5) Assess the vulnerability of key technologies to different categories of radiation effects. Create models that show the number of failures per billion hours.

6) Study power distribution techniques that have the potential to mitigate latch-up. Device reliability can increase if latch-ups can be localized within a device rather than impacting the entire device and if redundancy strategies can mitigate the effect of these localized failures.

VII. CONCLUSION

Designing processing architectures for space presents unique challenges such as challenging form factors, environmental requirements, thermal limits, material restrictions, high development costs, and size, weight, power limits. These challenges limit component choices and extend design cycles. Appropriate risk mitigation, testing programs, portability strategies, and design reuse can expand component selection and make use of more of the processing technologies available for open system architectures.

ACKNOWLEDGMENT

The authors acknowledge the management of the Digital Technology department at Northrop Grumman Electronic Systems in Baltimore, MD for their support of the work for this paper.

REFERENCES

[1] J. Holland and E. Glaser, “Designing processing architectures for space applications,” High Performance Embedded Computing (HPEC) Workshop, September 2008.

[2] J. Wertz and W. Larson, Space Mission Analysis and Design, 3rd ed. Torrance, CA: Microcosm Press, 1999.

[3] J. Holland, J. Horner, R. Kuning, and D. Oeffinger, "Implementation of Digital Front End Processing Algorithms with Portability across Multiple Processing Platforms," High Performance Embedded Computing (HPEC) Workshop, September 2011.

[4] M. Marinella, “The Future of Memory,” IEEE Aerospace Conference, 2013

[5] A. Lacaita and D. Ielmini, “Reliability issues and scaling projections for phase change non volatile memories,” Electron Devices Meeting, 2007. IEDM 2007. IEEE International

[6] M. Marinella, S. Dalton, et. al., “Initial Assessment of the Effects of Radiation on the Electrical Characteristics of TaO_x Memristive Memories”, IEEE Transactions on Nuclear Science, Vol. 59, No. 6, December 2012