

# A Framework to Analyze Processor Architectures for Next-Generation On-Board Space Computing

Tyler M. Lovelly, Donavon Bryan, Kevin Cheng, Rachel Kreynin, Alan D. George, Ann Gordon-Ross  
NSF Center for High-Performance Reconfigurable Computing (CHREC)  
Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA  
{lovelly,donavon,cheng,kreynin,george,ann}@chrec.org

Gabriel Mounce  
Air Force Research Laboratory (AFRL), Space Vehicles Directorate  
Kirtland Air Force Base, Albuquerque, NM, USA  
gabriel.mounce2@kirtland.af.mil

**Abstract**—Due to harsh and inaccessible operating environments, space computing presents many unique challenges with respect to stringent power, reliability, and programmability constraints that limit on-board processing performance and mission capabilities. However, the increasing need for real-time sensor and autonomous processing, coupled with limited communication bandwidth with ground stations, are increasing the demand for high-performance, on-board computing for next-generation space missions. Since currently available radiation-hardened space processors cannot satisfy this growing demand, research into various processor architectures is required to ensure that potential new space processors are based on architectures that will best meet the computing needs of space missions. To enable this research, we present a novel framework to analyze potential processor architectures for space computing. By using this framework to analyze a wide range of existing radiation-hardened and emerging commercial processors, tradeoffs between potential space computing architectures can be determined and considered when designing new space processors or when selecting commercial architectures for radiation hardening and use in space missions. We demonstrate the ability of the framework to generate data for various architectures in terms of performance and power, and analyze this data for initial insights into the effects of processor architectures on space mission capabilities. The framework provides a foundation for the analysis of a broad and diverse set of processor architectures for potential use in next-generation, on-board space computing.

lengthy, complex, and costly process, and since space mission design typically requires lengthy development cycles, there is a large technological gap between commercial and space processors that results in limited and outdated processor selections for space missions.

While current space processors increasingly lag behind the capabilities of emerging commercial processors, computing requirements for space missions are becoming more demanding. Furthermore, improving sensor technology and increasing mission data rates, problem sizes, and data types are increasing the demand for communication bandwidth to ground stations. Due to limited bandwidth and long transmission latencies, remote transmission of real-time operating decisions or new software/hardware reconfigurations becomes impractical for space missions. High-performance, on-board computing can alleviate these challenges and address the unique computing needs of space missions by processing data prior to transmission to ground stations and making real-time operating decisions autonomously.

To address the continually increasing demand for high-performance, on-board space computing, new processor architectures must be analyzed for potential new space processors. Current rad-hard space processors are typically based on commercial processors with architectures that were not explicitly designed for the unique challenges of space computing. To ensure that new space processors are based upon architectures that are most suitable for next-generation space missions, tradeoffs in architectural characteristics should be determined and considered when designing a space processor or when selecting a commercial architecture for radiation hardening and use in space missions. However, this analysis presents several challenges, since both the space-computing domain and the set of available processors are broad and diverse, with many possible applications and processor architectures to evaluate. To address these challenges, Figure 1 conceptualizes the proposed framework that enables the analysis of potential processor architectures for space computing.

To study and characterize the broad space-computing domain, we perform an expansive study to determine common and critical space mission computing requirements that considers key applications, data types, problem sizes, and other relevant algorithmic details. With this information, we establish a set of computational dwarfs and compose a taxonomy that broadly defines and classifies the space-computing domain. From this taxonomy, we establish a benchmark suite that consists of key computations that broadly represent space mission requirements, and thus simplifies the space-computing domain to a manageable set of computations.

## TABLE OF CONTENTS

1	INTRODUCTION .....	1
2	BACKGROUND AND RELATED WORK .....	2
3	SPACE-COMPUTING TAXONOMY .....	3
4	DEVICE METRICS ANALYSIS .....	4
5	DEVICE BENCHMARKING ANALYSIS .....	6
6	CONCLUSIONS AND FUTURE RESEARCH .....	8
	ACKNOWLEDGMENTS .....	8
	REFERENCES .....	8
	BIOGRAPHY .....	9

## 1. INTRODUCTION

Most currently available radiation-hardened (rad-hard) space processors, such as the BAE Systems RAD750 and the Xilinx Virtex-5QV, are the result of commercial processor architectures being selected for radiation hardening and use in space missions [1-2]. Since creating rad-hard space processors is a

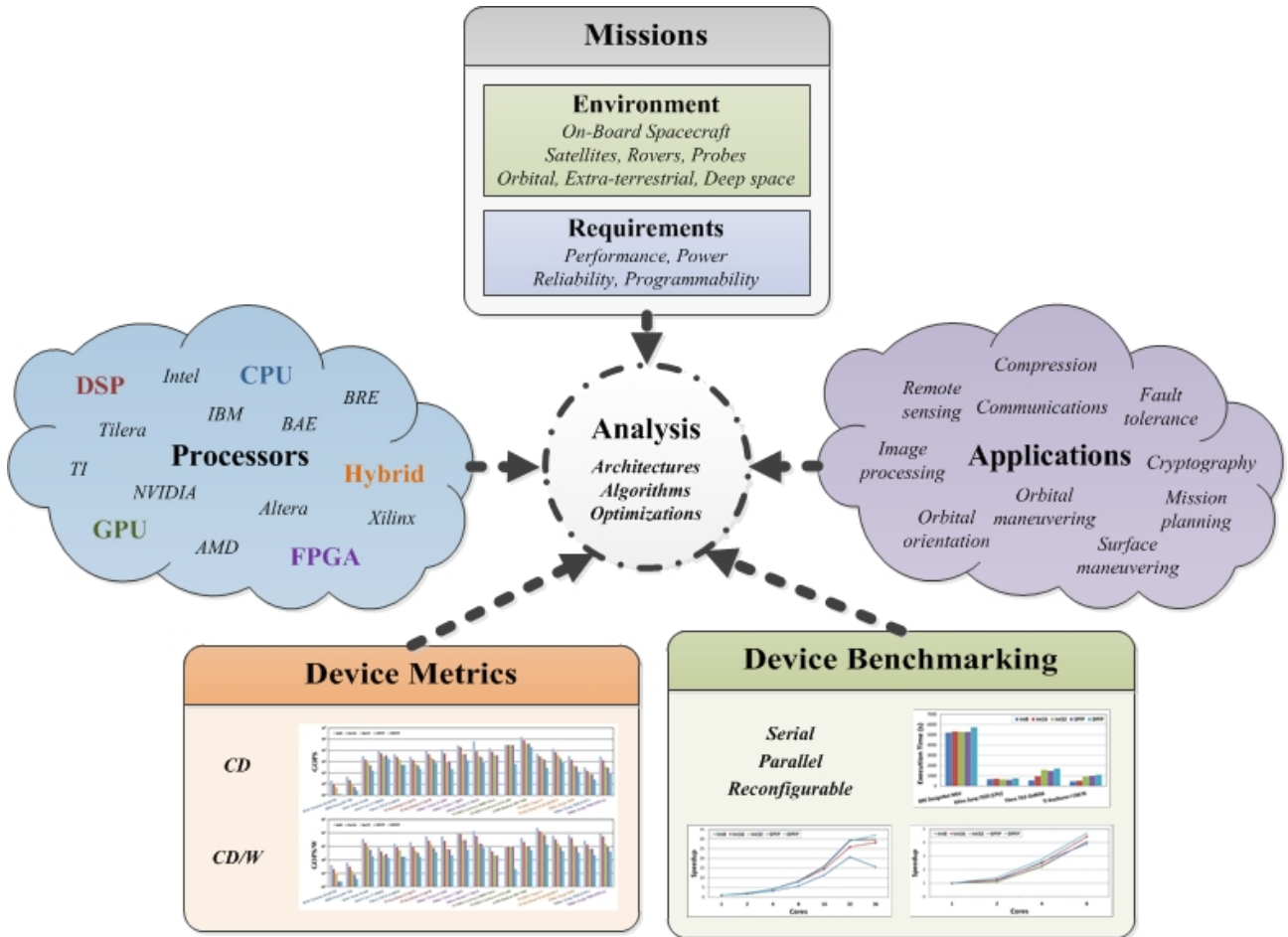


Figure 1: Framework concept diagram

To identify and characterize the numerous and diverse set of potential processor architectures for space computing, we leverage a suite of device metrics that provide a theoretical basis for the study of architectural capabilities. Facilitated by device metrics, we conduct initial quantitative analysis and objective comparison of many diverse processor architectures, from categories such as multi-core and many-core central processing units (CPUs), digital signal processors (DSPs), field-programmable gate arrays (FPGAs), graphics processing units (GPUs), and hybrid configurations of these architectures. Device metrics analysis provides insights into which architectures are most suitable for space computing in terms of performance and power. We then target the most suitable architectures for further analysis with device benchmarking by developing a space-computing benchmark suite and testing the performance capabilities of the targeted architectures. We first test benchmarks in serial operation, then further develop and test for parallelization across processor cores and reconfigurable fabrics. The framework enables analysis of potential processor architectures for space computing based on theoretical capabilities and the performance of key computations required for space missions. While this research focuses on space computing, our methodologies can be adapted and applied to processor architecture analysis for any computing domain.

The remainder of this paper is structured as follows. Section 2 describes background and related work. In Section 3, we present computational dwarfs and a novel taxonomy for

space computing that includes key computations selected to establish a space-computing benchmark suite. In Section 4, we present device metrics analysis over a broad and diverse set of architectures for potential use as space processors. In Section 5, we conduct device benchmarking analysis on architectures that show potential for use in space computing. Finally, Section 6 discusses conclusions and future research directions.

## 2. BACKGROUND AND RELATED WORK

The framework leverages established concepts in the analysis of algorithms and processor architectures, and applies these concepts to the space-computing domain. These concepts include the creation of taxonomies for various computing domains based upon computational dwarfs, device metrics analysis as an initial comparison of the capabilities of a broad set of architectures, and device benchmarking for further architecture analysis based upon the performance of key computations.

### Computational Dwarfs

The University of California at Berkeley (UCB) introduced the computational dwarf concept for designing and analyzing computing models and architectures. Asanovic et al. [3] defined a computational dwarf as “an algorithmic method that captures a pattern of computation and communication.” Figure 2 lists these UCB dwarfs, which were defined at high

levels of abstraction to encompass all computational methods used in modern computing. The UCB dwarfs can be used to characterize applications by determining the application’s key computations and classifying the application under the appropriate dwarf. For example, computations such as matrix multiplication and QR decomposition are both classified as linear algebra dwarfs, while Fast Fourier Transform (FFT) and wavelet transform are both classified as spectral methods dwarfs. Abstracting applications as computational dwarfs enables analysis of key computational patterns across a wide range of applications, independent of the actual software and hardware implementation details. For any computing domain, computational dwarfs can be identified and used to create taxonomies that broadly define and classify the key computational patterns within that domain. This concept has been demonstrated in various computing domains, such as symbolic computation [4], cloud computing [5], and workload characterization [6]. The framework leverages these concepts to establish computational dwarfs and a taxonomy for space computing.

- |   |  |
|---|--|
| <ul style="list-style-type: none"> <li>• <i>Dense linear algebra</i></li> <li>• <i>Sparse linear algebra</i></li> <li>• <i>Spectral methods</i></li> <li>• <i>N-body methods</i></li> <li>• <i>Structured grids</i></li> <li>• <i>Unstructured grids</i></li> <li>• <i>MapReduce</i></li> </ul> | <ul style="list-style-type: none"> <li>• <i>Combinational logic</i></li> <li>• <i>Graph traversal</i></li> <li>• <i>Dynamic programming</i></li> <li>• <i>Backtrack and branch-and-bound</i></li> <li>• <i>Graphical models</i></li> <li>• <i>Finite state machines</i></li> </ul> |
|---|--|

**Figure 2:** UCB dwarfs [3]

#### Device Metrics

We leverage an in-house developed set of device metrics for quantitative analysis of processor architectures in terms of performance and power [7-9]. Device metrics provide a theoretical basis for processor capability analysis and can be calculated based solely on architectural characteristics, allowing for the study of a larger and broader set of architectures than is practical with device benchmarking. Device metrics enable the objective comparison of disparate architectures, from categories such as multi-core and many-core CPU, DSP, FPGA, GPU, and hybrid configurations. In this research, we employ device metrics to analyze and compare a broad set of processor architectures as an initial step to determine the architectures’ potential for space computing.

Since space processors must operate with limited power consumption, the framework focuses on the computational density (CD) and CD per Watt (CD/W) device metrics. CD and CD/W data provide an initial analysis of architectural capabilities in terms of performance and power. CD evaluates a processor’s raw performance capabilities in terms of addition and multiplication operations per second, and is calculated separately for each data type considered. The framework evaluates various data types, including several integer precisions and both single-precision and double-precision floating-point (SPFP and DPFP, respectively). Richardson et al. [7] defined CD as:

$$CD = f \times \sum_{i=1}^n \frac{N_i}{CPI_i} \quad (1)$$

where  $N$  is the number of execution units or the number of operations that can be issued simultaneously,  $CPI$  is the average number of cycles per instruction, and  $f$  is the operating frequency. This calculation accounts for all  $n$

types of execution units that can support operations with the evaluated data type. To ensure that CD is memory-sustainable, the realistic ability of memory to provide data for each parallel operation is considered. Richardson et al. [7] defined CD/W as CD divided by power, which measures how much performance is achieved for each Watt dissipated.

Calculating CD and CD/W is more complex for FPGAs [7-8] as compared to traditional processors, and requires use of vendor tools. First, arithmetic cores for the evaluated operations and data types are generated for the evaluated FPGA, and a linear-programming algorithm is applied to FPGA resource utilization data, which is used to determine optimal packing of cores onto the reconfigurable fabric. Then, CD is calculated by multiplying the maximum possible number of cores by the limiting core frequency, and CD/W is calculated by using vendor tools for power estimation to determine static and dynamic power for the device given the packing configuration.

#### Device Benchmarking

Whereas device metrics serve as a valuable first step in the analysis of processor architectures, more thorough analysis can be conducted with device benchmarking. Developing and testing device benchmarks on the processors provides insights into the performance capabilities of the evaluated architectures. Although device benchmarking analysis requires greater hardware costs and development efforts than device metrics analysis, the resulting insights are specific to key computations for the evaluated computing domain. Further analysis becomes possible as processors approach theoretical capabilities and the effects of architectural characteristics on performance can be carefully studied. Thus, device benchmarking provides a methodology to compare performance tradeoffs for various architectures, algorithms, and optimizations under consideration.

### 3. SPACE-COMPUTING TAXONOMY

To establish a set of computational dwarfs for space computing, we conduct a comprehensive study of space applications, data types, problem sizes, and other key algorithmic details based upon space mission needs. Requirements for on-board space computing are rapidly increasing due to advancements in remote sensors and data acquisition, including common radar and laser applications and operations for merging sensor data, which impose intensive computational demands. Image processing is commonly required, such as imaging across frequency spectrums and in noisy environments, in addition to resolution enhancement, stereo vision, and detection and tracking of features across image frames [10-13].

Guidance, navigation, and control applications are key to space missions, and require intensive computing for real-time autonomous operations, which includes horizon and star tracking, and determination and control algorithms for spacecraft attitude and orbit [14-15]. Autonomous maneuvering is required in orbital missions for proximity operations, such as relative motion control for rendezvous and docking and on-orbit assembly [16-18]. Surface missions require autonomous maneuvering to study foreign environments and to safely and efficiently land on and navigate unfamiliar terrain [19-22]. Autonomous mission planning consists of intelligent scheduling and abstract modeling of spacecraft control operations and profiling of on-board science experiments [23-25].

Communication capabilities with ground stations or other remote systems are also vital for space missions, and include software-defined radio and packet switching operations [26]. Since sensor data cannot always be processed on-board, and communication bandwidth to ground stations is limited, data compression can reduce communication requirements to ensure that critical sensor data is retrieved and analyzed [27]. Due to the unreliability of remote communication systems and hazards posed by the harsh space environment, fault tolerance is critical for space missions. Data reliability can be strengthened with periodic memory scrubbing and channel coding of data transmissions [28]. Encryption techniques are considered since cryptography may be necessary to protect sensitive mission information [29]. While mission security may require specific, classified encryption algorithms, computationally-similar unclassified algorithms are also of significance for less-sensitive or shorter-duration missions.

Given these requirements, Table 1 depicts the space-computing taxonomy, which is composed of broad, high-level computational dwarfs and corresponding applications. This taxonomy provides a comprehensive assessment of common and critical requirements for on-board space computing. Table 2 depicts the space-computing benchmark suite. These benchmarks represent key computations required by the corresponding dwarfs in Table 1. Since it is impractical to exhaustively consider every possible space application or algorithm, this taxonomy and benchmark suite provide a broad representation of key computations required for space missions. These computations can be characterized with the more abstracted UCB dwarfs, with most benchmarks classified under the linear algebra, spectral methods, and combinatorial logic UCB dwarfs. By developing this benchmark suite and conducting device benchmarking, architectures can be targeted and analyzed for potential use as space processors and the processors' performance capabilities can be analyzed for key computations either for specific applications or broadly across the space-computing domain.

#### 4. DEVICE METRICS ANALYSIS

Device metrics provide a method for the initial analysis and comparison of a broad set of processor architectures without the increased hardware costs and development efforts required for more exhaustive device benchmarking. Device metrics provide a quantitative and objective comparison of similar or disparate architectures, including comparison of existing rad-hard technology with emerging commercial processors. Architectures can be analyzed for various data types used in space computing, and hybrid architectures can be studied as individual constituent components or in a combined/hybrid fashion. This analysis provides insights into which architectures are most suitable for space computing in terms of performance and power and should be further evaluated with device benchmarking.

Figures 3 and 4 show initial CD and CD/W data, reported in billions (giga) of operations per second (GOPS) and GOPS per Watt (GOPS/W), respectively, for various integer and floating-point data precisions on a broad range of architectures. Table 3 shows corresponding raw data and architectural categorizations. Using this data, we study various architectural tradeoffs to gain insights into specific architectural considerations for space computing in terms of performance and power. Evaluated architectural categories include multi-core and many-core CPUs, DSPs, FPGAs, GPUs, and hybrid configurations.

**Table 1:** Space-computing taxonomy

Dwarf	Application areas
<i>Remote sensing</i>	<i>Synthetic-aperture radar Light detection and ranging Beamforming Sensor fusion</i>
<i>Image processing</i>	<i>Hyper/multi-spectral imaging Overhead persistent infrared Super resolution imaging Stereo vision Feature detection &amp; tracking</i>
<i>Orbital orientation</i>	<i>Horizon &amp; star tracking Attitude determination &amp; control Orbit determination &amp; control</i>
<i>Orbital maneuvering</i>	<i>Relative motion control Rapid trajectory generation On-orbit assembly</i>
<i>Surface maneuvering</i>	<i>Autonomous landing Hazard detection &amp; avoidance Terrain classification &amp; mapping Path optimization</i>
<i>Mission planning</i>	<i>Intelligent scheduling Model checking Experiment profiling</i>
<i>Communications</i>	<i>Software-defined radio Packet switching</i>
<i>Compression</i>	<i>Image &amp; video compression Hyper/multi-spectral compression</i>
<i>Fault tolerance</i>	<i>Memory scrubbing Channel coding</i>
<i>Cryptography</i>	<i>NSA Type-1 certified encryption Unclassified encryption</i>

**Table 2:** Space-computing benchmark suite

Benchmark	Dwarfs
<i>Matrix multiplication</i>	<i>Remote sensing Image processing Orbital orientation</i>
<i>Matrix transpose</i>	<i>Remote sensing</i>
<i>Convolution</i>	<i>Remote sensing Image processing Orbital orientation</i>
<i>FFT</i>	<i>Remote sensing Communications</i>
<i>QR decomposition</i>	<i>Image processing</i>
<i>Wavelet transform</i>	<i>Image processing Compression</i>
<i>TCP/IP operations</i>	<i>Communications</i>
<i>Error correction coding</i>	<i>Fault tolerance</i>
<i>Rijndael AES</i>	<i>Cryptography</i>
<i>Singular value decomposition</i>	<i>Orbital orientation</i>
<i>Lambert's problem</i>	<i>Orbital orientation Orbital maneuvering</i>
<i>Graph search</i>	<i>Orbital orientation Mission planning</i>
<i>Artificial potential function</i>	<i>Orbital maneuvering</i>
<i>Newton's method</i>	<i>Orbital maneuvering Surface maneuvering</i>
<i>Kalman filtering</i>	<i>Orbital maneuvering Surface maneuvering</i>

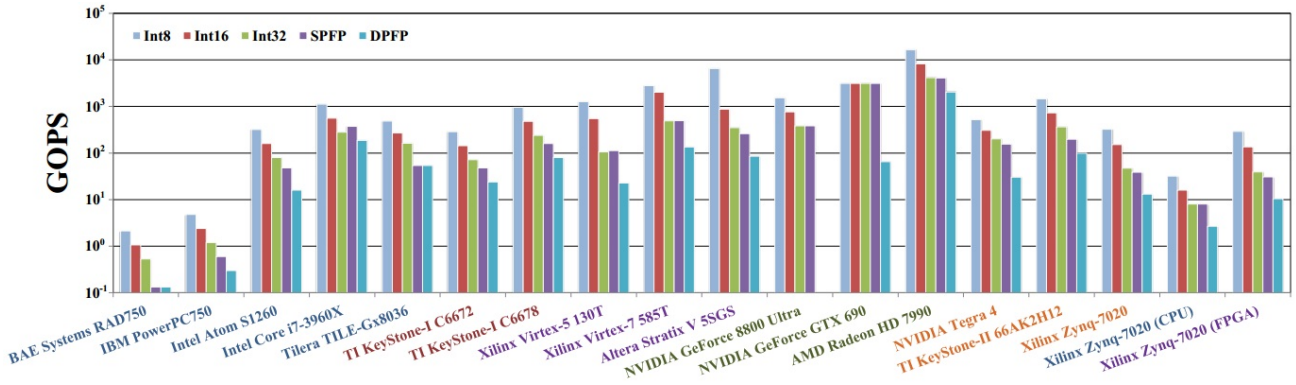


Figure 3: CD comparison for broad set of architectures

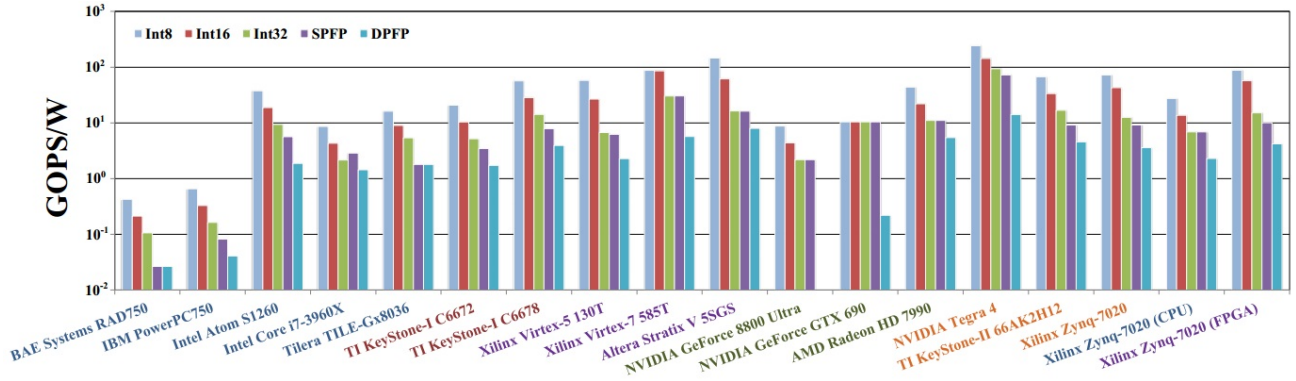


Figure 4: CD/W comparison for broad set of architectures

Table 3: CD and CD/W raw data

Device		CD (GOPS)					W	CD/W (GOPS/W)				
		Int8	Int16	Int32	SPFP	DPFP		Int8	Int16	Int32	SPFP	DPFP
CPU	BAE Systems RAD750	2.13	1.06	0.53	0.13	0.13	5.00	0.43	0.21	0.11	0.03	0.03
	IBM PowerPC750	4.80	2.40	1.20	0.60	0.30	7.30	0.66	0.33	0.16	0.08	0.04
	Intel Atom S1260	320.00	160.00	80.00	48.00	16.00	8.50	37.65	18.82	9.41	5.65	1.88
	Intel Core i7-3960X	1123.20	561.60	280.80	374.40	187.20	130.00	8.64	4.32	2.16	2.88	1.44
	Tiler TILE-Gx8036	486.00	270.00	162.00	54.00	54.00	30.00	16.20	9.00	5.40	1.80	1.80
DSP	TI KeyStone-I C6672	288.00	144.00	72.00	48.00	24.00	13.86	20.78	10.39	5.19	3.46	1.73
	TI KeyStone-I C6678	960.00	480.00	240.00	160.00	80.00	20.30	56.75	28.37	14.19	7.88	3.94
	Xilinx Virtex-5 130T	1271.22	550.08	106.09	112.87	22.75	21.92	57.99	26.77	6.71	6.25	2.29
FPGA	Xilinx Virtex-7 585T	2798.66	2028.80	490.80	495.70	135.30	31.93	87.65	85.80	30.40	30.50	5.70
	Altera Stratix V 5SGS	6526.95	877.50	351.00	261.00	85.69	44.74	145.90	61.71	16.38	16.26	7.99
	NVIDIA GeForce 8800 Ultra	1536.00	768.00	384.00	384.00	0.00	175.00	8.78	4.39	2.19	2.19	0.00
GPU	NVIDIA GeForce GTX 690	3130.39	3130.39	3130.39	3130.39	65.22	300.00	10.43	10.43	10.43	10.43	0.22
	AMD Radeon HD 7990	16448.00	8256.00	4160.00	4096.00	2048.00	375.00	43.86	22.02	11.09	11.09	5.46
	NVIDIA Tegra 4	520.64	307.84	201.44	155.84	30.40	2.15	242.16	143.18	93.69	72.48	14.14
Hybrid	TI KeyStone-II 66AK2H12	1459.20	729.60	364.80	198.40	99.20	21.69	67.28	33.64	16.82	9.15	4.57
	Xilinx Zynq-7020	324.15	152.00	47.38	38.78	13.14	4.48	72.35	42.82	12.57	9.17	3.59
	Xilinx Zynq-7020 (CPU)	32.02	16.01	8.04	8.04	2.68	1.17	27.44	13.72	6.89	6.89	2.30
	Xilinx Zynq-7020 (FPGA)	292.13	135.99	39.34	30.74	10.46	4.48	88.26	57.14	15.13	10.06	4.21

The results show that the prevalent BAE Systems RAD750 space processor and the corresponding commercial processor, the IBM PowerPC750, are becoming obsolete by several orders of magnitude when compared to the emerging processors in all major architectural categories studied. While architectural variations of the RAD750 exist, CD and CD/W data is based on a frequency of 133 MHz and a power dissipation of 5W [1].

Device metrics can be used not only to objectively compare very disparate architectures, but also architectures with similarities, such as belonging to the same architectural category, vendor, or processor family. When analyzing commercial architectures for potential use in space missions, we first compare several CPUs from the same architectural category.

Results show that the Intel Atom S1260 lacks the performance capabilities of the Intel Core i7-3960X and the Tiler TILE-Gx8036, but achieves higher CD/W for all evaluated data types, revealing the low-power advantage of the Atom S1260 architecture. The Tiler TILE-Gx8036 fails to match the performance capabilities of the Core i7-3960X, but does achieve higher CD/W for most data types. We also compare commercial DSPs from the same architectural category, vendor, and processor family. Results show that the octal-core TI KeyStone-I C6678 achieves both higher CD and CD/W than the dual-core KeyStone-I C6672. When analyzing FPGAs, we evaluate the Xilinx Virtex-5 130T FPGA, which is the commercial counterpart of the Virtex-5QV space processor, and find that greater CD and CD/W can be achieved with the more advanced Virtex-7 and Altera Stratix V architectures.

The results for GPUs reveal not only the high power requirements of emerging commercial GPUs, but the effects of data types and precisions on CD. The NVIDIA GeForce 8800 Ultra contains floating-point units capable of only SPFP operations, which is insufficient for high-precision applications that require DPFP processing. While the NVIDIA GeForce GTX 690 does contain DPFP units, CUDA cores are used to support both SPFP and 32-bit integer operations, with smaller integer precisions automatically converted up to 32-bit values. Therefore, applications can leverage precision levels up to 32-bit integer or SPFP without decreasing CD. With the AMD Radeon HD 7990, CD data follows a more predictable pattern and decreases with higher precisions.

Device metrics also provide the capability to analyze hybrid architectures, such as the NVIDIA Tegra 4, which combines a GPU with a quad-core CPU, or the TI KeyStone-II, which contains both an octal-core DSP and quad-core CPU. With hybrid architectures, CD and CD/W are calculated first for all constituent architectures. CD values are then combined to give the hybrid CD, which is then divided by the total power dissipation to obtain the hybrid CD/W. Therefore, hybrid architectures can be analyzed with constituent architectures in isolation or in a combined fashion as is demonstrated with the Xilinx Zynq-7020, which contains both a dual-core ARM Cortex-A9 CPU and an Artix-7 FPGA fabric. Computational metrics data for the Zynq-7020 shows that the FPGA fabric provides most of the performance capability and achieves better CD/W than the CPU for all data types studied.

## 5. DEVICE BENCHMARKING ANALYSIS

Although analysis with device metrics provides a valuable initial step for architecture analysis, more thorough analysis with device benchmarking can provide further insights into architectures that show potential for use as space processors. We develop several benchmarks in the space-computing benchmark suite and test these benchmarks on the targeted architectures. The resulting timing data provides insights into performance capabilities for key computations either for specific applications or broadly across the space-computing domain. We evaluate device benchmarking on a variety of processors to analyze the effects of different architectures, algorithms, and optimizations on performance. The benchmarks developed include a triple-loop matrix multiplication of two  $n \times n$  matrices, a double-loop matrix transpose of an  $n \times n$  matrix, and a quad-loop flipped-kernel convolution of an  $n \times n$  matrix. We first conduct benchmarking with serial operations to compare existing rad-hard technology with emerging commercial processors. We then adapt the benchmarks to parallel operation to test the parallelizability of computations across processor cores. Finally, we develop a benchmarking strategy to test the acceleration of computations by mapping to reconfigurable fabrics. Although initial benchmarking data provides insights into potential architectures for future space processors, extensive optimization of algorithms and architectures is required to more closely reflect the full capabilities of the evaluated processors.

To analyze existing rad-hard technology, we conduct device benchmarking on the BRE DesignNet MSV, an existing rad-hard space processor of close architectural similarity to the more prevalent BAE Systems RAD750. The MSV is based on the commercial IBM PowerPC750FX single-core architecture, which is analogous to the IBM PowerPC750 found in the RAD750, with some minor additional features. To analyze the various architectural categories of emerging commercial

processors, we conduct device benchmarking on the Xilinx Zynq-7020, which is a hybrid CPU/FPGA architecture, the Tiler TILE-Gx8036, which is a many-core CPU, and the TI KeyStone-I C6678, which is a multi-core DSP.

We develop benchmarks in C and VHDL for various data types and problem sizes and verify correct operation using known test patterns. We then conduct benchmarking with randomized data using processor-specific design tools and environments, including a single-board computer platform with the WindRiver VxWorks OS and design tools for the BRE DesignNet MSV, a Digilent ZedBoard with the Xilinx OS and Xilinx ISE design tools for the Xilinx Zynq-7020, a TILEmpower-Gx platform with CentOS for the Tiler TILE-Gx8036, and an EVM board with the MCSDK SYS/BIOS and TI CCS design tools for the TI KeyStone-I C6678.

### Serial Benchmarking

Figures 5 and 7 show serial benchmarking results, which demonstrate the relatively, and expected, inferior performance of existing rad-hard technology when compared to emerging commercial processors for all evaluated data types, even when commercial multi-core and many-core architectures are limited to single-core operation. While the rad-hard processor shows the worst performance for all benchmarks tested, the best performing architecture varies depending upon the benchmark and data type, which shows that these aspects should be considered when analyzing architectures for future space processors.

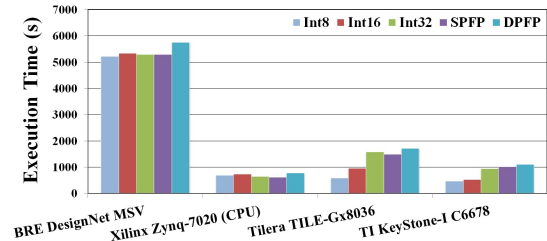


Figure 5: Serial matrix multiplication,  $n = 2048$

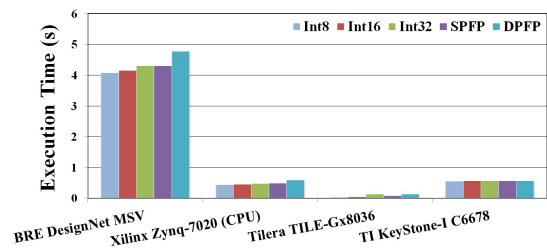


Figure 6: Serial matrix transpose,  $n = 2048$

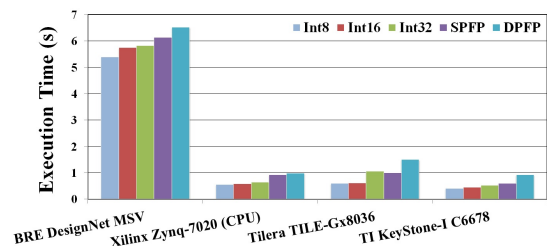
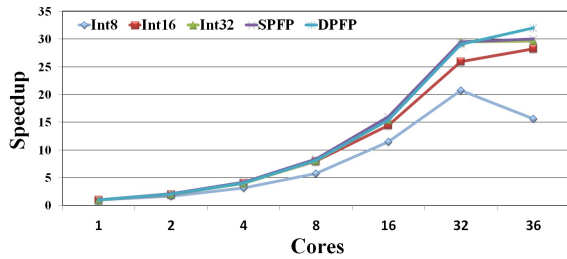
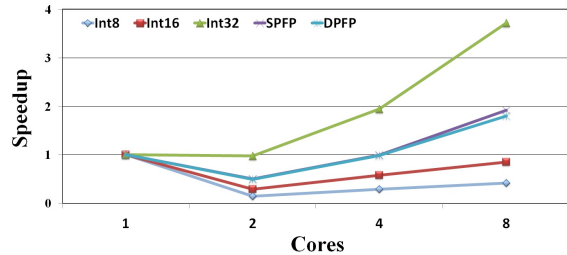


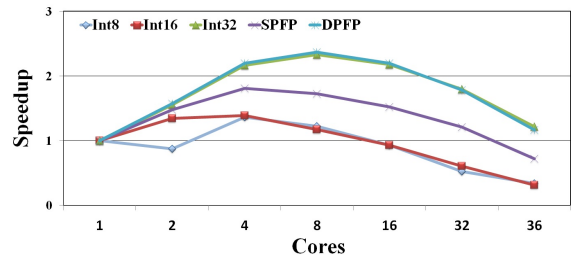
Figure 7: Serial convolution,  $3 \times 3$  Sobel kernel,  $n = 2048$



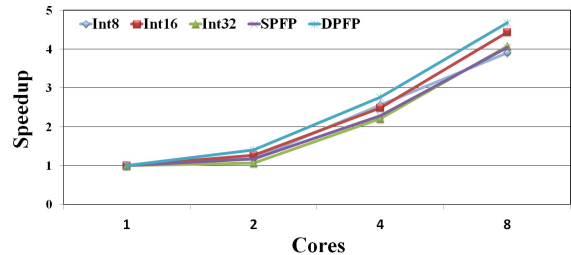
**Figure 8:** Parallel matrix multiplication  
Tiler TILE-Gx8036,  $n = 2048$



**Figure 9:** Parallel matrix multiplication  
TI KeyStone-I C6678,  $n = 2048$



**Figure 10:** Parallel matrix transpose  
Tiler TILE-Gx8036,  $n = 2048$



**Figure 11:** Parallel matrix transpose  
TI KeyStone-I C6678,  $n = 2048$

### Parallel Benchmarking

Since multi-core, many-core, and hybrid architectures are of increasing prevalence in emerging commercial processors, we further develop our benchmarks with an OpenMP shared-memory parallelization strategy. We then test performance on the Tiler TILE-Gx8036 and the TI KeyStone-I C6678 to analyze the parallelizability of computations across processor cores.

Figures 8 and 9 show parallelized matrix multiplication results for the TILE-Gx8036 and KeyStone-I C6678, respectively. For both architectures, speedup is achieved as computations are distributed across processor cores. The TILE-Gx8036's speedup increases as the number of cores increases up to an eventual tipping point in performance gains caused by the communication overhead associated with the parallelization strategy. When increasing cores on the KeyStone-I C6678, a performance penalty is incurred that prevents speedup for lower data precisions without increased optimization. For higher precisions, speedup occurs when the number of cores is high enough for the benefits of parallelization to overtake any communication overhead. Unlike the TILE-Gx8036, the KeyStone-I C6678 does not contain enough cores to reach a tipping point in performance gains for the evaluated benchmark. However, for most parallelized applications, a performance limit exists where additional parallelization across cores will stop significantly increasing speedup and can begin degrading performance. Using application details such as data types and problem sizes, the optimal number of parallel cores can be determined for a specified architecture. Achieved speedup and optimal number of parallel cores vary with differing architectures and parallelization strategies, as greater development and optimization efforts may be required for some processors to reach performance levels near theoretical capabilities.

Figures 10 and 11 show parallelized matrix transpose results for the TILE-Gx8036 and KeyStone-I C6678, respectively. For both processors, speedup is achieved with increasing number of processor cores until an eventual tipping point

in performance gains. While the TILE-Gx8036 achieves greater speedup for matrix multiplication than the KeyStone-I C6678, the opposite is true for matrix transpose. This outcome demonstrates that speedup for targeted processors may vary between algorithms of different computational characteristics, allowing decisions about architectures for potential new space processors to be adjusted based upon specific application requirements.

### Reconfigurable Benchmarking

Since customized hardware acceleration of compute-intensive algorithms has the potential to alleviate performance bottlenecks for space missions, FPGA benchmarking is used to determine the amenability of space applications to reconfigurable fabrics. We evaluate FPGA acceleration by designing a reconfigurable convolution benchmark in VHDL and test this benchmark on the Artix-7 FPGA fabric of the Xilinx Zynq-7020. The convolution datapath is designed to perform arithmetic operations in parallel. Throughput is increased by dividing the datapath into pipeline stages, which reduces the propagation delay incurred per clock cycle, thus increasing achievable operating frequencies.

**Table 4:** Reconfigurable convolution  
Xilinx Zynq-7020 (FPGA),  $3 \times 3$  Sobel kernel,  $n = 2048$

Data precision	Resource utilization			Performance		
	FF	LUT	DSP	cycles	frequency	time (ms)
Int8	160 (1%)	32 (1%)	17 (7%)	466033	500 MHz	0.9321
Int16	320 (1%)	43 (1%)	17 (7%)	466033	500 MHz	0.9321
Int32	928 (1%)	210 (1%)	35 (15%)	466033	500 MHz	0.9321
SPFP	4010 (3%)	2735 (5%)	43 (19%)	466033	300 MHz	1.5534
DPFP	12706 (11%)	8315 (15%)	123 (55%)	466033	300 MHz	1.5534

Table 4 shows reconfigurable benchmarking results, which are analyzed both in terms of performance and FPGA resources utilized for the convolution datapath. The design requires a relatively small percentage of flip-flop (FF), look-up table (LUT), and hard-wired multiplier (DSP) resources available on the FPGA fabric. Throughput is further increased by generating a phase-locked loop to increase operat-

ing frequencies. Performance is then calculated by scaling the total number of cycles required to finish computation by the operating frequency. Results demonstrate that FPGA acceleration can be achieved without requiring significant resource utilization, enabling additional FPGA-based computation on remaining resources as needed.

## 6. CONCLUSIONS AND FUTURE RESEARCH

As the need for high-performance, on-board space computing is continually increasing, we develop a novel framework to analyze the capabilities of processor architectures to meet future space mission requirements. The framework addresses the challenges presented when considering both a broad domain of computing and a broad set of potential architectures. We evaluate the framework's ability to gain initial insights into a variety of processor architectures for potential space computing, and we evaluate a broad range of common and critical space applications to identify computational dwarfs and establish a novel space-computing taxonomy. With a basis in this taxonomy, key computations are selected for a space-computing benchmark suite that broadly represents the computational needs of future space missions. Device metrics analysis is demonstrated as an initial step to quantitatively and objectively compare a broad set of processor architectures for potential use in space computing. Using a space-computing benchmark suite, device benchmarking is conducted for more thorough and targeted analysis of processor architectures. Serial benchmarks are developed and tested on both existing rad-hard technology and emerging commercial architectures with potential for use as space processors. Benchmark parallelizability is tested across processor cores and reconfigurable fabrics. Results confirm that existing rad-hard technology is greatly outperformed by emerging commercial processors, and that space-computing benchmarks often demonstrate parallelizability, with multi-core, many-core, and reconfigurable architectures enabling performance speedup. The framework thus creates a research foundation for the quantitative analysis of processor architectures for use in next-generation, on-board space computing.

Future research will extend the framework for further analysis of tradeoffs in architectural characteristics for space computing with a more complete set of emerging processors, including new multi-core, many-core, reconfigurable, and hybrid architectures. The space-computing benchmark suite will be further developed, and benchmarking methodologies will be improved by leveraging existing optimized libraries and creating optimized benchmarking environments for architectures under study.

## ACKNOWLEDGMENTS

This work was supported in part by the IUCRC Program of the National Science Foundation under Grant Nos. EEC-0642422 and IIP-1161022.

## REFERENCES

- [1] A. Cady, "RAD750 System Flight Computer," BAE Systems, Feb 13 2002
- [2] Xilinx, Inc., "Radiation-Hardened, Space-Grade Virtex-5QV Family Overview," DS192 (v1.3), Mar 8 2012
- [3] K. Asanovic, R. Bodik, B. C. Catanzaro, J. J. Gebis, P. Husbands, K. Keutzer, D. A. Patterson, W. L. Plishker,

- J. Shalf, S. W. Williams and K. A. Yelick, "The Landscape of Parallel Computing Research: A View from Berkeley," Technical Report No. UCB/EECS-2006-183, University of California, Berkeley, Dec 18 2006
- [4] E. L. Kaltofen, "The 'Seven Dwarfs' of Symbolic Computation," Numerical and Symbolic Scientific Computing, Texts and Monographs in Symbolic Computation, vol.1, pp.95-104, 2012
- [5] S. C. Phillips, V. Engen, and J. Papay, "Snow White Clouds and the Seven Dwarfs," IEEE Third International Conference on Cloud Computing Technology and Science (CloudCom), pp.738,745, Nov 29 - Dec 1 2011
- [6] K. Manakul, P. Siripongwutikorn, S. See, and T. Achalakul, "Modeling Dwarfs for Workload Characterization," IEEE 18th International Conference on Parallel and Distributed Systems (ICPADS), pp.776,781, Dec 17-19 2012
- [7] J. Richardson, S. Fingulin, D. Raghunathan, C. Massie, A. George, and H. Lam, "Comparative Analysis of HPC and Accelerator Devices: Computation, Memory, I/O, and Power," Proc. of High-Performance Reconfigurable Computing Technology and Applications Workshop (HPRCTA), at SC'10, New Orleans, LA, Nov 14 2010
- [8] J. Williams, C. Massie, A. D. George, J. Richardson, K. Gosrani, and H. Lam, "Characterization of Fixed and Reconfigurable Multi-Core Devices for Application Acceleration," ACM Transactions on Reconfigurable Technology and Systems (TRETs), vol.3, no.4, pp.19:1-19:29, Nov 2010
- [9] J. Williams, A. D. George, J. Richardson, K. Gosrani, and S. Suresh, "Computational Density of Fixed and Reconfigurable Multi-Core Devices for Application Acceleration," Proc. of Reconfigurable Systems Summer Institute (RSSI), Urbana, IL, Jul 7-10 2008
- [10] A. Plaza, D. Qian, C. Yang-Lang, and R. L. King, "High Performance Computing for Hyperspectral Remote Sensing," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol.4, no.3, pp.528,544, Sep 2011
- [11] W. Visser, K. Havelund, G. Brat, S. Park, and F. Lerda, "Model Checking Programs," ACM Journal of Automated Software Engineering, vol.10, no.2, pp.203-232, Apr 2003
- [12] S. B. Goldberg and L. Matthies, "Stereo and IMU assisted visual odometry on an OMAP3530 for small robots," IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp.169,176, Jun 20-25 2011
- [13] C. Chang, H. Ren, and S. Chiang, "Real-time processing algorithms for target detection and classification in hyperspectral imagery," IEEE Transactions on Geoscience and Remote Sensing, vol.39, no.4, pp.760,768, Apr 2001
- [14] J. A. Tappe, "Development of Star Tracker System for Accurate Estimation OF Spacecraft Attitude," Masters Thesis, Naval Postgraduate School, 2009
- [15] G. A. Boyarko, M. Romano, and O. A. Yakimenko, "Time-optimal reorientation of a spacecraft using a direct optimization method based on inverse dynamics," IEEE Aerospace Conference, pp.1,13, March 6-13 2010
- [16] G. A. Boyarko, O. A. Yakimenko, and M. Romano, "Real-Time 6DoF Guidance For of Spacecraft Proxim-



ity Maneuvering and Close Approach with a Tumbling Object,” AIAA/AAS Astrodynamics Specialist Conference, Toronto, Ontario, Canada, August 2-5 2010

- [17] I. Lopez and C. R. McInnes, “Autonomous rendezvous using artificial potential function guidance,” AIAA Journal of Guidance, Control, and Dynamics, vol.18, no.2, Mar - Apr 1995
- [18] A. Badawy and C. McInnes, “On-orbit assembly using superquadric potential fields,” AIAA Journal of Guidance, Control and Dynamics, vol.31, no.1, Jan - Feb 2008
- [19] C. D. Epp and T. B. Smith, “Autonomous Precision Landing and Hazard Detection and Avoidance Technology (ALHAT),” IEEE Aerospace Conference, pp.1,7, March 3-10 2007
- [20] T. Estlin, B. Bornstein, D. Gaines, D. R. Thompson, R. Castano, R. C. Anderson, C. de Granville, M. Burl, M. Judd, and S. Chien, “AEGIS Automated Science Targeting for the MER Opportunity Rover,” ACM Transactions on Intelligent Systems and Technology (TIST), vol.3, no.50, New York, NY, USA, May 2012
- [21] S. B. Goldberg, M. W. Maimone, and L. Matthies, “Stereo vision and rover navigation software for planetary exploration,” IEEE Aerospace Conference, vol.5, pp.5-2025,5-2036, Mar 9-16 2002
- [22] M. Bajracharya, J. Ma, A. Howard, and L. Matthies, “Real-time 3D Stereo Mapping in Complex Dynamic Environments,” International Conference on Robotics and Automation - Semantic Mapping, Perception, and Exploration (SPME) Workshop, May 15 2012
- [23] R. Knight and S. Hu, “Compressed Large-scale Activity Scheduling and Planning (CLASP),” Proc. of the Sixth International Workshop in Planning and Scheduling for Space (IWSS), Pasadena, California, Jul 19-21 2009
- [24] A. Weiss, I. Kolmanovsky, M. Baldwin, and R. S. Erwin, “Model Predictive Control of three dimensional spacecraft relative motion,” American Control Conference (ACC), pp.173,178, Jun 27-29 2012
- [25] T. Wongpiromsarn, U. Topcu, and R. M. Murray, “Receding Horizon Temporal Logic Planning,” IEEE Transactions on Automatic Control, vol.57, no.11, pp.2817,2830, Nov 2012
- [26] M. N. O. Sadiku and C. M. Akujuobi, “Software-defined radio: a brief overview,” IEEE Potentials, vol.23, no.4, pp.14,15, Oct - Nov 2004
- [27] Consultative Committee for Space Data Systems (CCSDS), “Image Data Compression,” Blue Book, Issue 1, Nov 2005
- [28] Consultative Committee for Space Data Systems (CCSDS), “TM Synchronization and Channel Coding,” Blue Book, Issue 2, Aug 2011
- [29] Consultative Committee for Space Data Systems (CCSDS), “CCSDS Cryptographic Algorithms,” Blue Book, Issue 21, Nov 2012

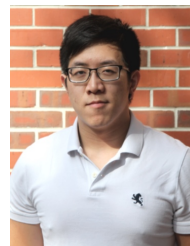
## BIOGRAPHY



**Tyler M. Lovelly** is a Ph.D. student in Electrical & Computer Engineering at the University of Florida, where he received his M.S. and B.S. degrees. He is a graduate researcher and group leader at the NSF Center for High-Performance Reconfigurable Computing (CHREC). His professional experience includes AFRL Space Vehicles at Kirtland Air Force Base and United Space Alliance at NASA Kennedy Space Center. His research interests include advanced space computing, computer architecture, high-performance computing, reconfigurable systems, fault tolerance, robotics, and machine intelligence.



**Donavon Bryan** is an M.S. student in Electrical & Computer Engineering at the University of Florida, where he received his B.S. degree. He is a graduate researcher at the NSF Center for High-Performance Reconfigurable Computing (CHREC), working on device metrics and benchmarking for space processor characterizations. His professional experience includes two internships at NASA Goddard Space Flight Center. His research interests include advanced space computing, reconfigurable systems, high-performance computing, and fault tolerance.



**Kevin Cheng** is an M.S. student in Electrical & Computer Engineering at the University of Florida, where he received his B.S. degree. He is a graduate researcher at the NSF Center for High-Performance Reconfigurable Computing (CHREC), working on device metrics studies for fixed and reconfigurable architectures. His research interests include computer architecture, reconfigurable systems, fault tolerance, and multi-core computing.



**Rachel Kreynin** is a B.S. student in Electrical Engineering at the University of Florida. She is an undergraduate researcher at the NSF Center for High-Performance Reconfigurable Computing (CHREC), working on device benchmarking for reconfigurable architectures. Her professional experience includes AFRL Space Vehicles at Kirtland Air Force Base. Her research interests include advanced space computing, computer architecture, and reconfigurable systems.



**Alan D. George** is Professor of ECE at the University of Florida, where he founded and directs the NSF Center for High-Performance Reconfigurable Computing (CHREC). He received the B.S. degree in CS and the M.S. in ECE from the University of Central Florida, and the Ph.D. in CS from the Florida State University. His research interests

focus upon high-performance architectures, networks, systems, services, and applications for reconfigurable, parallel, distributed, and fault-tolerant computing. Dr. George is a Fellow of the IEEE.



**Ann Gordon-Ross (M'00)** received her B.S and Ph.D. degrees in Computer Science and Engineering from the University of California, Riverside (USA) in 2000 and 2007, respectively. She is currently an Associate Professor of Electrical and Computer Engineering at the University of Florida (USA) and is a member of the NSF Center for High-Performance Reconfigurable Computing (CHREC) at the University of Florida. She is also the faculty advisor for the Women in Electrical and Computer Engineering (WECE) and the Phi Sigma Rho National Society for Women in Engineering and Engineering Technology. She received her CAREER award from the National Science Foundation in 2010 and Best Paper awards at the Great Lakes

Symposium on VLSI (GLSVLSI) in 2010 and the IARIA International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM) in 2010. Her research interests include embedded systems, computer architecture, low-power design, reconfigurable computing, dynamic optimizations, hardware design, real-time systems, and multi-core platforms.



**Gabriel Mounce** is the Deputy Chief for the Space Electronic Technology Program of the Air Force Research Laboratory's Space Vehicles Directorate. As such, Mr. Mounce directs research activities focused on increasing the reliability, survivability, and performance of space electronics used in the U.S. Air Force and other federal agency space systems. Mr. Mounce received his B.S. in Electrical Engineering from New Mexico State University and his M.S. in Electrical Engineering from the Air Force Institute of Technology.