



Jet Propulsion Laboratory
California Institute of Technology

High Performance Spaceflight Computing (HPSC) Next Steps at NASA and AFRL

presented by

Richard J. Doyle

Manger, Information and Data Science Program

Jet Propulsion Laboratory, California Institute of Technology

presented at

University of California, Riverside

Dept. of Computer Science & Engineering

April 12, 2013

Air Force Research Laboratory:

NASA Ames Research Center:

NASA Goddard Space Flight Center:

Jet Propulsion Laboratory:

NASA Johnson Space Center:

Keith Avery, Gabriel Mounce

Michael Lowry

Michael Johnson, Wesley Powell

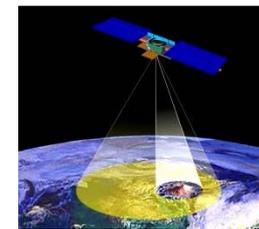
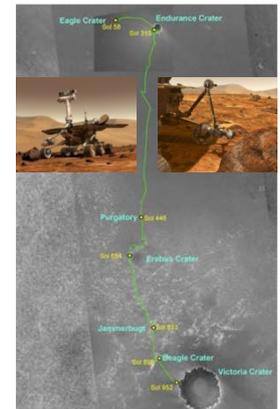
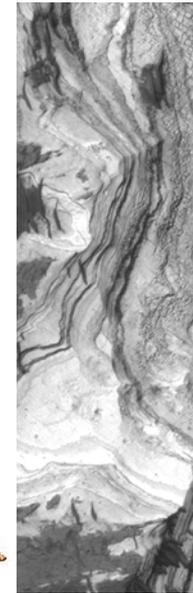
Larry Bergman, Raphael Some, William Whitaker

Montgomery Goforth

The NASA Need

There are important mission scenarios that today cannot be accomplished robustly and cost-effectively

- Space-based computing has not kept up with the needs of current and future NASA missions
- Government and industry are developing high-performance space-qualifiable processors
- But there is a need for NASA leadership and investment
 - Due to unique requirements and architectural features needed for the energy efficiency and fault tolerance challenges of Deep Space, Earth-Observing, and Human Spaceflight missions



HPSC Formulation Task

Assignment, Results

The Assignment	The Results
Identify relevant NASA use cases What are the paradigm-shifting NASA space-based applications that will drive next generation flight computing?	Developed 9 human spaceflight (HEOMD) and 10 science mission (SMD) use cases for future flight computing, spanning critical mission functions, high data rate instruments, and autonomy utilizing model-based reasoning techniques
Derive requirements What are the future onboard computing requirements?	100X performance increase, low power (down to 7W) with scaling, support for a range of fault tolerance, common programming languages, avoidance of additional V&V effort, interoperable with co-processors
Perform a gap analysis How/where do commercial and defense industry developments in computing fall short of NASA's unique requirements and architectural needs?	No existing or emerging spaceflight processors possess all necessary performance, power efficiency, reliability, and programmability attributes
Trade architectures against the defined Key Performance Parameters (KPPs) Which computing architecture will make the most difference?	Rad-hard general-purpose multi-core best addresses the future flight computing requirements and presents the most affordable gap against the KPPs
Make a recommendation How can NASA best invest limited resources to meet the future needs of its space systems?	Competed/directed program plan for rad-hard general-purpose multi-core, with solutions for power/energy, fault tolerance and other NASA requirements, leveraging other agency and industry investments

NASA Applications

for High Performance Spaceflight Computing

HEOMD Use Cases

1. Cloud Services
2. Advanced Vehicle Health Management
3. Crew Knowledge Augmentation Systems
4. Improved Displays and Controls
5. Augmented Reality for recognition and cataloging
6. Tele-Presence
7. Autonomous & Tele-Robotic Construction
8. Automated Guidance, Navigation, and Control (GNC)
9. Human Movement Assist

SMD Use Cases

1. Extreme Terrain Landing
2. Proximity Operations / Formation Flying
3. Fast Traverse
4. New Surface Mobility Methods
5. Imaging Spectrometers
6. Radar
7. Low Latency Products for Disaster Response
8. Space Weather
9. Autonomous Mission Planning
10. Immersive Environments for Science Ops / Outreach

High value and mission critical applications identified by NASA scientists and engineers

Science Mission Applications

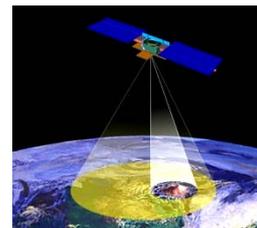
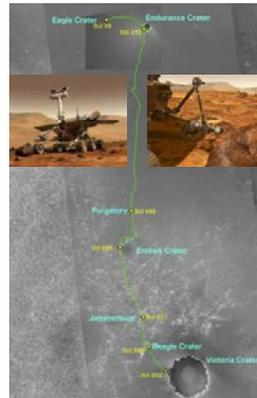
for High Performance Spaceflight Computing

SMD Use Cases

1. **Extreme Terrain Landing**
2. Proximity Operations / Formation Flying
3. **Fast Traverse**
4. New Surface Mobility Methods
5. Imaging Spectrometers
6. Radar
7. Low Latency Products for Disaster Response
8. Space Weather
9. **Science Event Detection and Response**
10. Immersive Environments for Science Ops / Outreach

Benefits to Missions

- **Extreme Terrain Landing**
 - Enables reliable and safe landing in hazardous terrain for TRN and HDA algorithms within a single computer – benchmarked by Mars Program as requiring six (6) dedicated RAD750s
- **Fast Traverse**
 - Remove computation as a limiting factor to mobility – drive 10X faster or more, safely
- **Science Event Detection and Response**
 - Increase capture rate for dynamic, transient events from ~10% to ~75%, with <5% false positives, for increased and more timely science return



**No longer need to size science / mission scope
to flight computing capability**

NASA Flight Computing Requirements

as derived from the NASA use cases

Computation Category	Mission Need	Objective of Computation	Flight Architecture Attribute	Processor Type and Requirements
Vision-based Algorithms with Real-Time Requirements	<ul style="list-style-type: none"> • Terrain Relative Navigation (TRN) • Hazard Avoidance • Entry, Descent & Landing (EDL) • Pinpoint Landing 	<ul style="list-style-type: none"> • Conduct safe proximity operations around primitive bodies • Land safely and accurately • Achieve robust results within available timeframe as input to control decisions 	<ul style="list-style-type: none"> • Severe fault tolerance and real-time requirements • Fail-operational • High peak power needs 	<ul style="list-style-type: none"> • Hard real time / mission critical • Continuous digital signal processing (DSP) + sequential control processing (fault protection) • High I/O rate • Irregular memory use • General-purpose (GP) processor (10's – 100's GFLOPS) + high I/O rate, augmented by co-processor(s)
Model-Based Reasoning Techniques for Autonomy	<ul style="list-style-type: none"> • Mission planning, scheduling & resource management • Fault management in uncertain environments 	<ul style="list-style-type: none"> • Contingency planning to mitigate execution failures • Detect, diagnose and recover from faults 	<ul style="list-style-type: none"> • High computational complexity • Graceful degradation • Memory usage (data movement) impacts energy management 	<ul style="list-style-type: none"> • Soft real time / critical • Heuristic search, data base operations, Bayesian inference • Extreme intensive & irregular memory use (multi-GB/s) • > 1GOPS GP processor arrays with low latency interconnect
High Rate Instrument Data Processing	High resolution sensors, e.g., SAR, Hyper-spectral	<ul style="list-style-type: none"> • Downlink images and products rather than raw data • Opportunistic science 	<ul style="list-style-type: none"> • Distributed, dedicated processors at sensors • Less stringent fault tolerance 	<ul style="list-style-type: none"> • Soft real time • DSP/Vector processing with 10-100's GOPS (high data flow) • GP array (10-100's GFLOPS) required for feature ID / triage

Computing Architectures

Candidates evaluated under the HPSC task

- General-purpose multi-core
 - Rad-hardened
 - COTS
- DSP multi-core
 - Rad-hardened
 - COTS (Eliminated early in study as not viable due to fault tolerance issues)
- Reconfigurable computing
 - Rad-hardened
 - COTS (Eliminated early in study as not viable due to radiation and fault tolerance issues)
- Graphics processing units (GPU)
 - Rad-hardened
 - COTS (Eliminated early in study as not viable due to radiation tolerance and power dissipation issues)
- *Criteria for choosing architecture(s):*
 - Gap to be closed within available budget
 - Already on or could be placed on a path to space qualification
 - TRL 6 maturity achievable in 3 years or less

Key Performance Parameters

Application-referenced KPPs

- Computational performance
- Radiation and fault tolerance
- Power and energy management
- Software verification and validation

Architecture-referenced KPPs

- Software verification and validation (this is the single cross-over KPP)
- Programmability and flight software applicability
- Interoperability
- Extensibility and evolveability

Additional KPPs

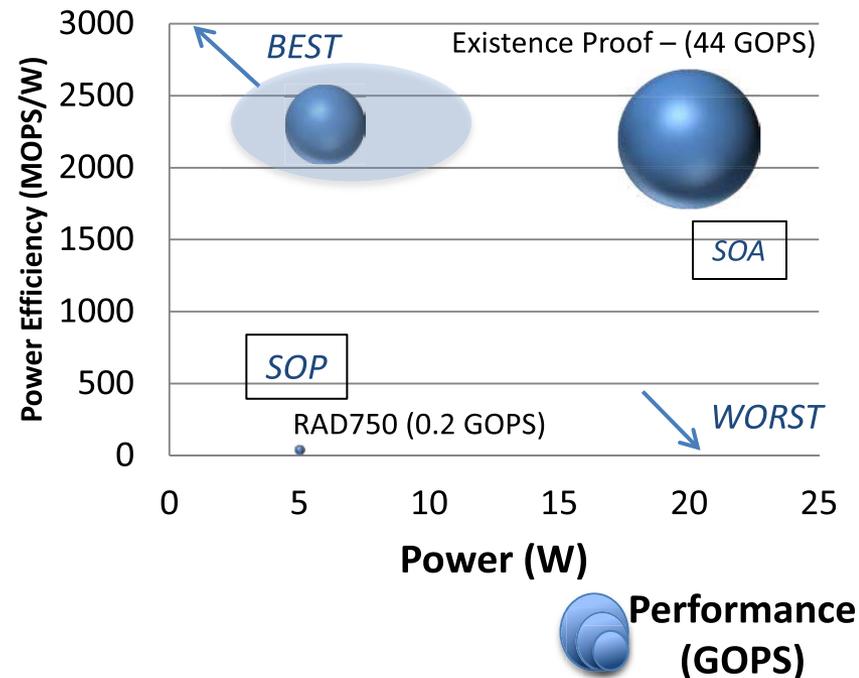
- Non-recurring cost
- Recurring cost
- Cross-cutting applicability across the NASA mission set

The Bottom Line

and How This Is Game Changing

Rad-hard General Purpose Multicore

- **Best overall fit to application requirements** – provides both general purpose and some DSP capability as well as interoperability with co-processors (DSP, Reconfigurable)
- Natively receptive to **Power Scaling** at core-level granularity
 - **Power Dissipation** issues to address fit within available investment resource envelope
- Natively receptive to thread-based **Fault Tolerance** approaches
 - Fault detection/correction/isolation
 - Ability to segregate failed cores from the pool of available cores in support of graceful degradation
- Scalable to order of magnitude increase in the number of cores
 - Combined with **Power Scaling** allows the increased cores to consume power only as thread-load dictates

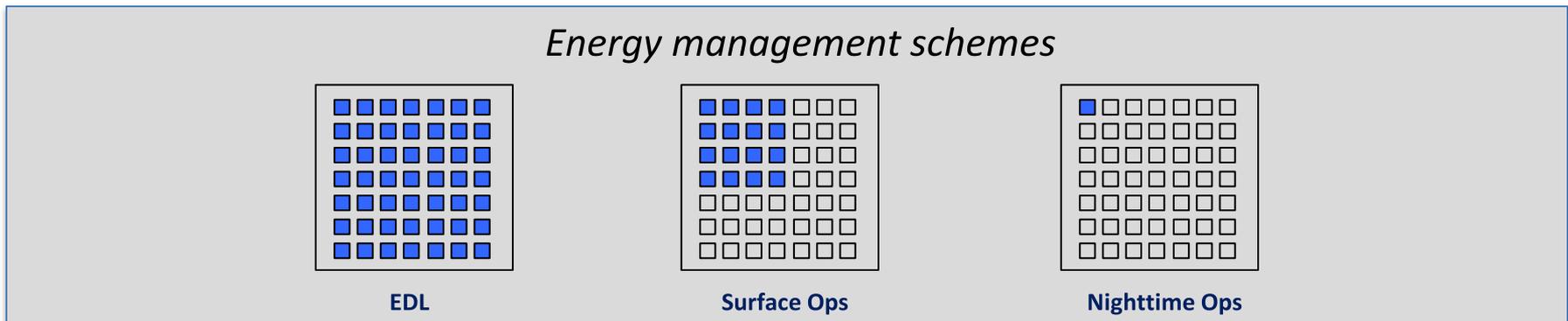
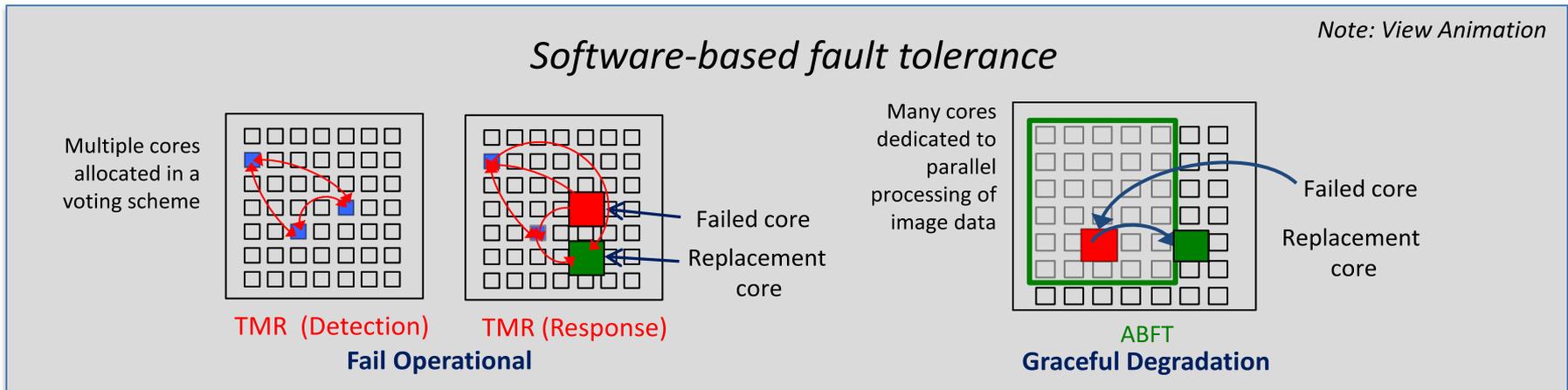


Computational Performance, Efficiency, and Scalability of Multi-core redefines general-purpose computing for space systems

The Benefit

Fault Tolerance and Energy Management

- Beyond the performance advantages, multicore architectures support system-level scalability, flexibility and efficiency



Scoring of the Architectures

vs. the KPPs

Key Performance Parameter (KPP)	Rad-hard General Purpose Multicore	Rad-hard DSP Multicore	Rad-hard Reconfigurable Computing	COTS-based Multicore	Rad-hard Graphics Processing Units
Cross-cutting Potential across NASA Missions	4.1	2.8	2.9	2.3	2.0
Computational Performance	5	2	4	5	5
Fault Tolerance	4	4	4	2	1
Power Dissipation	3	2	2	1	1
Power Scaling	5	3	5	1	2
Radiation Tolerance	4	4	4	2	3
Programmability and FSW Applicability	5	3	3	4	5
Flight Software V&V	4	3	3	4	5
Non-recurring cost	5	4	5	4	5
Recurring cost	4	3	2	5	2
Interoperability	4	3	2	5	4
Extensibility and Evolveability	5	4	4	4	3
Totals	52.1	37.8	40.9	39.3	38.0
# KPP scores above mean	12/12	4/12	7/12	6/12	5/12

Ranking of the Architectures

The Runners-Up

Rad-hard DSP multicore

- Specialized processing provides insufficient support for **Cross-cutting Applicability**, possibly requiring a general-purpose co-processor.
- Modular architecture may be amenable to a RHBD approach but with impact on **Non-Recurring Costs**.

Rad-hard reconfigurable computing

- Lack of tools and poor testability present difficulties in FSW design (**Programmability**) and **V&V**.
- The **Non-Recurring Cost** to develop the underlying hardware is significant; without large investment presents large **Recurring Costs** for any new or mission-specific functionality.
- **Power Dissipation** is significant as the underlying hardware re-programmability fabric is power inefficient.

COTS-based multicore

- High **Power Dissipation** and poor **Power Scaling** capability in pursuit of high-performance is a persistent problem within the COTS class.
- A lack of **Radiation Tolerance** leads to complex and power hungry redundancy solutions for mission-critical applications.
- Because of the availability of **Non-Recurring Cost** leveraging, innovative solutions in this class bear watching.

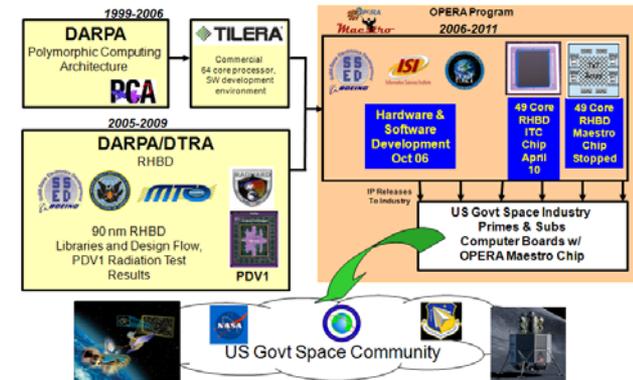
Rad-hard graphics processing units

- Despite suitability for certain image processing applications, the challenge to bring **Power Dissipation, Power Scaling, Fault Tolerance, Non-Recurring Cost, and Recurring Cost** in-line make this solution inappropriate for a flight computer with **Cross-cutting Applicability**.

HPSC Formulation Task Recommendations

Investment Focus and Approach

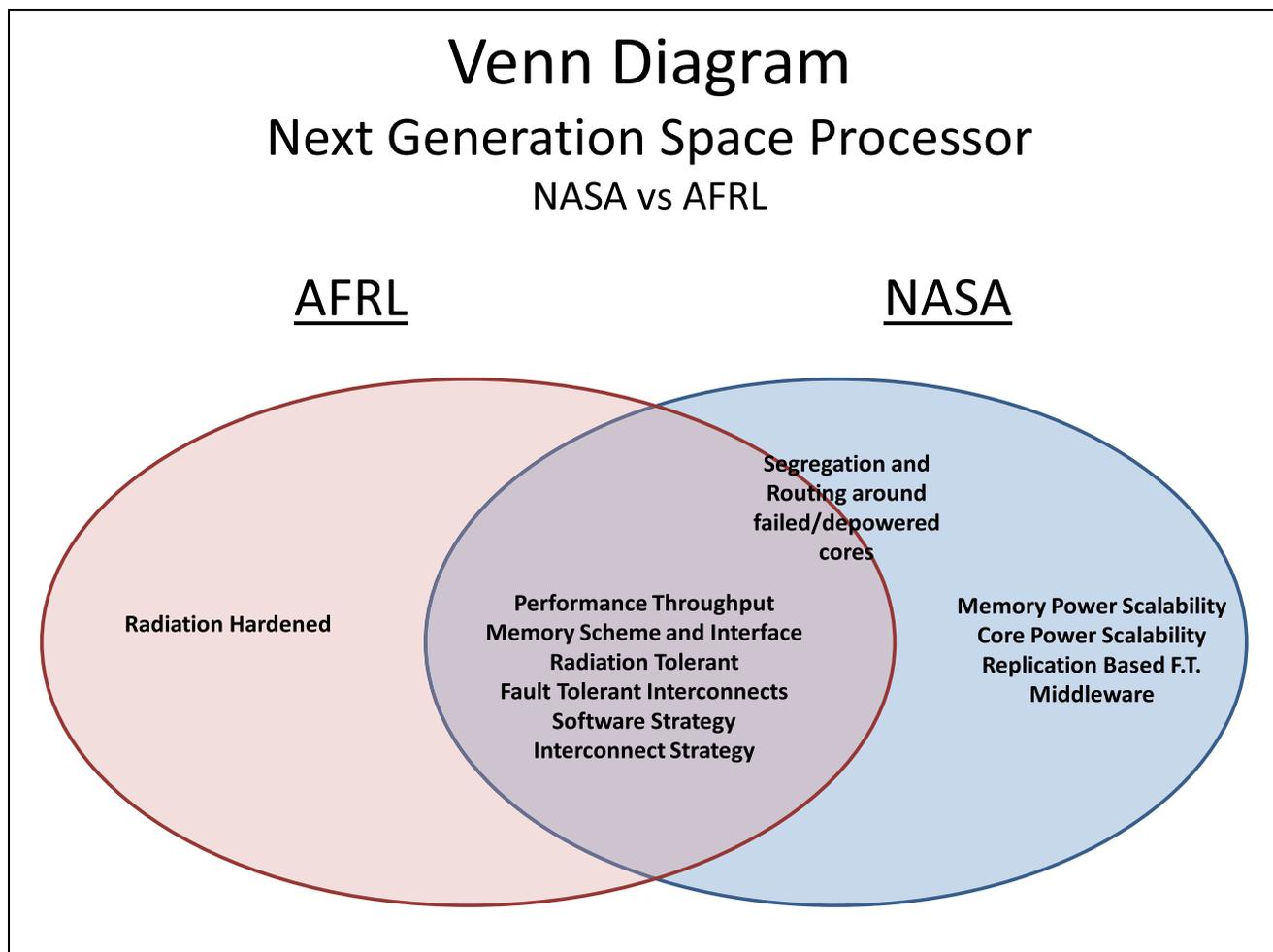
- Focus on **Rad-hard General Purpose Multi-core**
 - Leverage government and industry investments
- Issue a BAA for hardware in FY13
 - Solicit flight computing system concepts in this architecture class
 - Prepare NASA requirements and benchmarks, consistent with KPP and use case analysis
 - Include a competitive **Phase 0**, seeking innovative solutions, focused on risk retirement
- Include a directed software investment
 - Middleware elements for allocating/managing cores for varying operational objectives, working closely with the FSW community, driven by knowledge of the NASA applications
- Product of the investment
 - Multi-core hardware chip with bundled real-time operating system (RTOS), FSW development environment, and middleware elements, integrated on evaluation boards



Recent Development: Enter into a programmatic partnership with AFRL
Similar interests in future flight computing, and similar investment timeframe

Next-Generation Spaceflight Processor (NGSP)

AFRL vs. NASA Requirements



HPSC / NGSP Program Plan

Phase 0 BAA – Requirements

- Based on existing device with mature low-level software and accurate simulator or commercially available development system
- Minimum 20 GOPS/GFLOPS, 5GB/S memory & I/O bandwidth at 6W
- Dynamically power scalable at core level granularity by powering and depowering cores in real time without disrupting system operation, with very low idle power load ($\ll 1W$)
- Provides fault tolerant interconnects between cores and to external I/O and memory devices
- Natively supports multi-level replication based fault tolerance, e.g., N-Modular Redundancy where $N=2, 3$
- Supports segregation and routing around failed and depowered cores
- Radiation tolerant to at least 300kRad TID, Latch-up Immune, with Single Event Upset (SEU) rate of not greater than TBD/day in Adams 90% worst case GEO environment
- Interoperable with other high performance computing architectures, e.g., reconfigurable computing FPGAs

HPSC / NGSP Program Plan

Phase 0 BAA – Success Criteria

- Must identify the process and/or RHBD library to be used for the device, along with test data to substantiate claims of radiation hardness
 - RHBD technology with test data showing TID tolerance to 300kral, SEL immunity to 70MeV
- Must have simulation/emulation results for a set of NASA-defined benchmarks (FFT, search algorithms, etc.), including caveats where the limitations/errors of the simulation/emulation environment are articulated
 - Performance thresholds will be specific to each benchmark
- Must successfully simulate/emulate results for fault tolerance, including caveats where the limitations/errors of the simulation/emulation environment are articulated
 - Demonstrated ability to operate through faults and restore correct operation
- Must provide management plan that makes a credible case that the processor can be developed within BAA cost and schedule constraints
 - List of IP to be used, and agreements in place to acquire if selected
 - Detailed work breakdown structure provided to at least 3 levels
 - Complete device development schedule provided (including at least TBD months of reserves)
 - Detailed development cost estimate provided (including TBD% reserves)

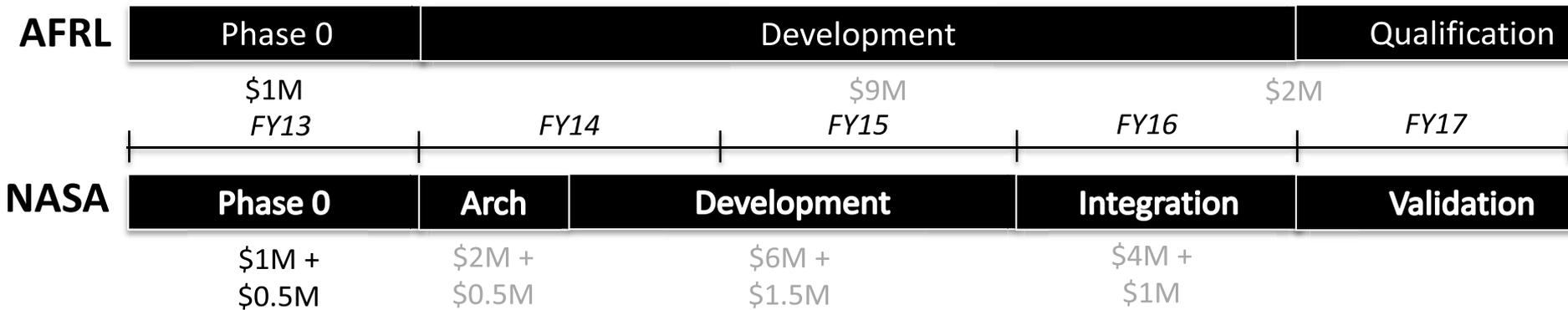
NASA / AFRL Partnership

Joint Phase 0 BAA on Flight Computing

NASA STMD Game Changing – Formulation Phase Complete
Decision per December 13, 2012 NASA Space Technology PMC
 NASA/AFRL to partner to develop a space-qualified high-performance multi-core processor

	FY13	FY14	FY15	FY16	Runout
AFRL - hardware	\$1M	\$3M	\$3M	\$3M	\$10M
NASA - hardware	\$1M	\$4M	\$4M	\$4M	\$13M
NASA - software	\$0.5M	\$1M	\$1M	\$1M	\$3.5M

Phase 0 cost sharing

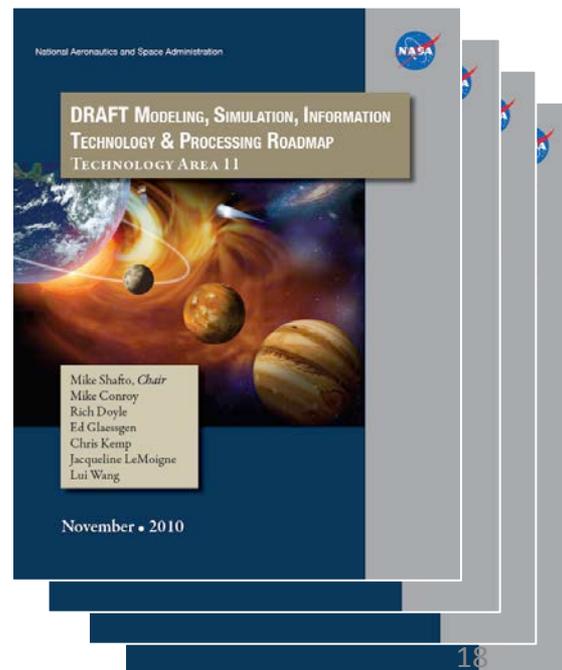


On track to issue joint NASA/AFRL BAA in April 2013
 Joint selection board to make contract awards in August 2013

Why Now?

Multi-core computing should be viewed as a foundation and amplifier for several roadmapped technologies

- (TA04) Advances in high performance low power onboard computers are central to more capable space robotics.
- (TA05) Many of the complex [objectives of] future missions...can be mitigated by making decisions closer to the platform...Clearly this goal is coupled with the need for increased autonomy and flight computing.
- (TA09) Landing challenges include highly capable and low power on-board dedicated compute elements...
- (TA11) Pinpoint landing, hazard avoidance, rendezvous-and-capture, and surface mobility are directly tied to the availability of high-performance space-based computing.



Without investment in a next-generation flight computing solution, the robotics and EDL systems developed in 2020 may be using processors and FPGAs that are a decade old and are ill suited to the task at hand

Transition Plan

Technology Infusion Paths – NASA and AFRL

- Mars Exploration – Future landed Mars missions (2020 and beyond)
 - EDL algorithms for pinpoint landing, for site access and sample return
 - Computation as a limiting factor for surface mobility – drive safely +10X faster
 - Perform science operations during traverse – “Walk and chew gum”
 - *Discussions are underway with the NASA Space Technology Program and the Mars Exploration Program towards a Mars 2020 technology payload concept to demonstrate the above capabilities, supported by HPSC*
- Earth Science – Future Earth-observing missions will carry high data rate instruments (hyper-spectral, radar ...)
- Human Spaceflight – MPCV/Orion: Time/space and memory partitioning will be an important human rating requirement – multi-core can provide natural fault containment structure
- AFRL – Sensor Payload Processing: Future systems will generate large amounts of data (hyper-spectral, hyper-temporal, radar ...)

Summary

Flight Computing for the Future at NASA and AFRL

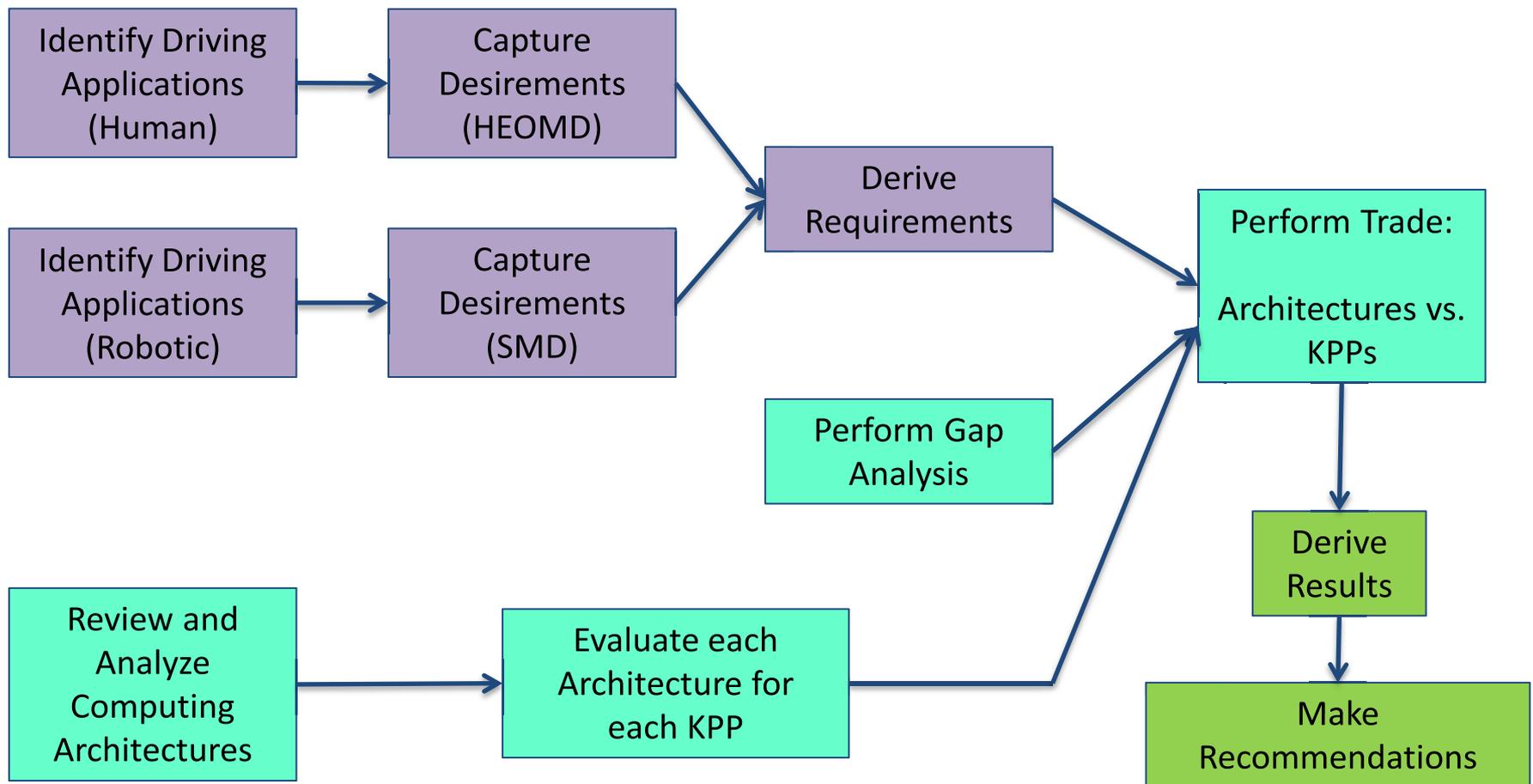
- Future NASA mission scenarios call out for significantly ***improved flight computing*** capability
- Several NASA OCT Roadmaps and the NRC report identify ***improved flight computing*** as a foundational technology
- ***Improved flight computing*** means increased computational performance, energy efficiency, and robust fault tolerance
- Like power and propulsion, flight computing is a ***core flight capability***, and a technology advance in this area is a natural ***multiplier*** which will impact the return from many missions and mission classes

It is time to move beyond the 1990's technology of the RAD750
Redefine the role of computing for space systems

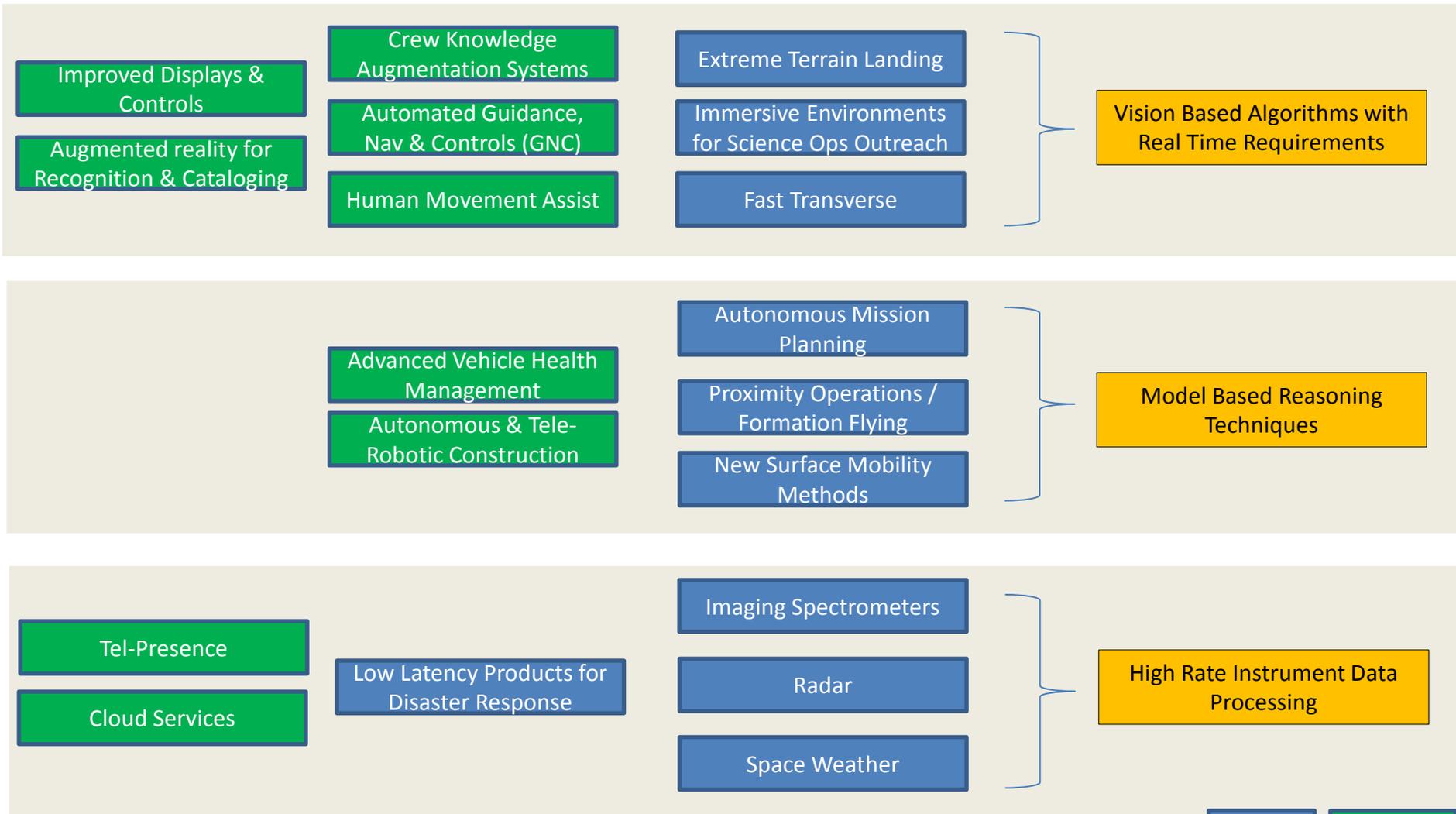


Backup

Formulation Approach *of the HPSC task*



Mapping of Use Cases to the Computation Categories



Eigen-Apps Summary

App to Eigen-App Mapping	DSP	GP	P	Mission Critical	LP
Throughput = 1-10 GOPS					
Autonomous Mission Planning		X	X	X	X
Disaster Response	X	X			X
Hyspiri	X	X	X		
Throughput = 10-50 GOPS					
Fast Traverse	X	X	X	X	X
Extreme Terrain Landing	X	X	X	X	X
Adept		X	X		
Optimum Observation	X	X	X		X
Space Weather	X		X		X
Robotic Servicing	X	X	X	X	
Cloud Service	X	X	X		
Advanced ISHM		X	X		X
Autonomous and Telerobotic Construction		X	X	X	X
Throughput = 50-100s GOPS					
Hyperspectral Imaging	X	X	X		X
RADAR Science	X	X	X		
RADAR EDL	X		X	X	X
Automated GN&C	X	X	X	X	
Human Movement Assist	X	X	X		X
Crew Knowledge Augmentation		X	X		
Improved Displays and Controls		X	X	X	X
Augmented Reality		X	X		X
Telepresence		X	X		X

- Requirements that represent groups of key cross cutting applications
- Derived by selecting low power applications from full applications set and grouping by throughput, processing type, mission criticality

Eigen-App	Throughput	DSP	GP	P	LP	MC
1	1-10 GOPS	X	X	X	X	
2	1-10 GOPS		X	X	X	X
3	10-50 GOPS	X	X	X	X	X
4	10-50 GOPS	X	X	X	X	
5	10-50 GOPS		X	X	X	X
6	10-50 GOPS		X	X	X	
7	50-100 GOPS	X	X	X	X	X
8	50-100 GOPS	X	X	X	X	
9	50-100 GOPS		X	X	X	X
19	50-100 GOPS		X	X	X	

KEY

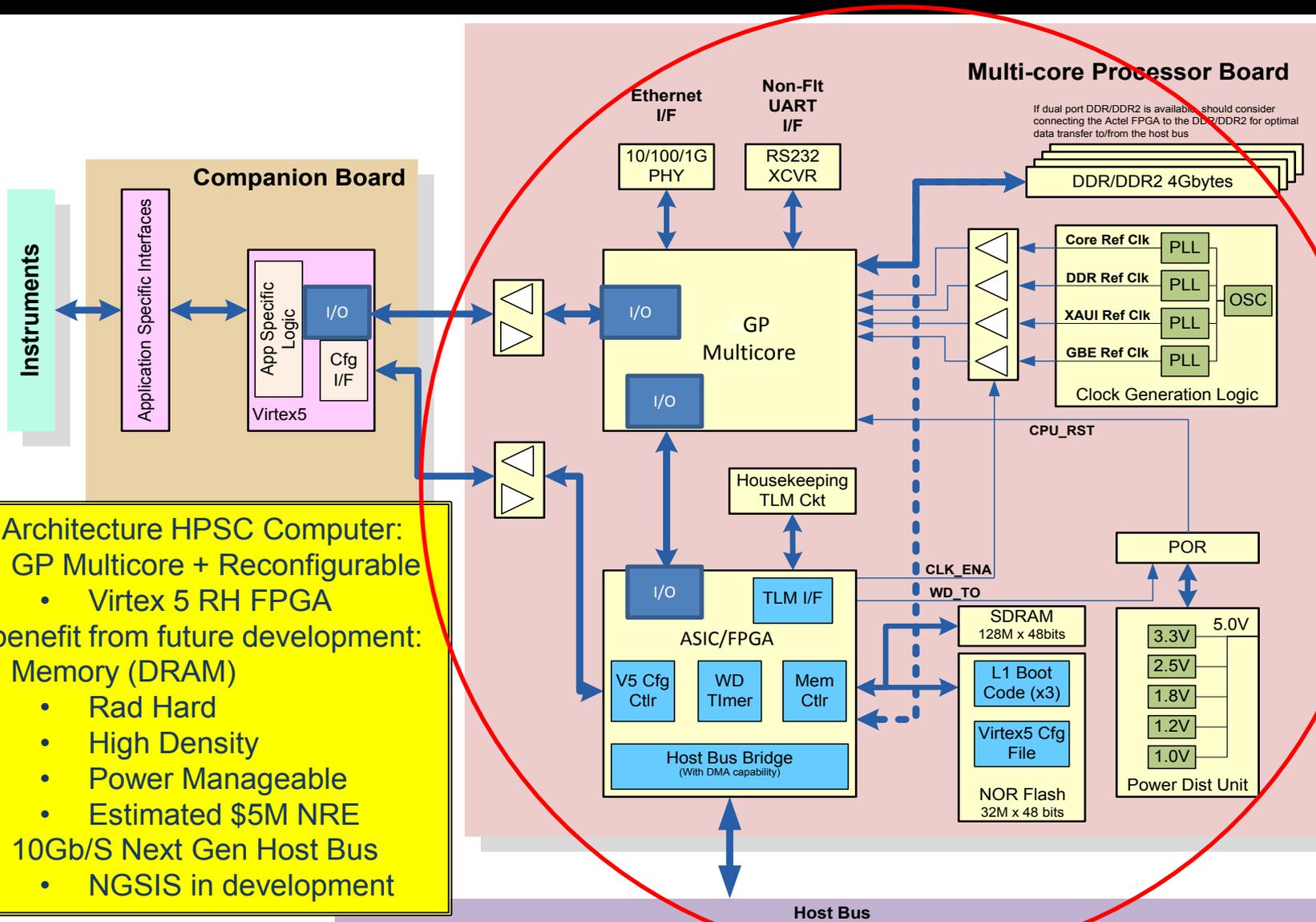
- DSP – Digital Signal Processing
- GP – General Purpose Processing
- P – Parallelizable
- Mission Critical – Requires Additional Fault Tolerance
- LP – Max Power Available for Processor Chip <6W

Gap Analysis

- ***There currently exists no spaceflight processor possessing:***
 - Processing performance and data rate consistent with needs of future HPSC applications (processing at least 30 GOPS, minimum 10Gbps data rate)
 - Ability to accommodate a broad range of processing classes (DSP, matrix/vector math, general purpose control processing)
 - Low power dissipation (less than 7W for the processor)
 - Determinism suitable for use in real-time applications
 - Reliability and fault tolerance suitable for use in human life critical applications
 - Programmability with standard software languages and tools
- ***Some emerging computing architectures are very capable for specific applications, but:***
 - Non-rad-hardened architectures present a basic reliability concern
 - Low power budgets and power scaling ability are absent in most of the architectures

To obtain a flight computing solution with the above set of attributes for future NASA missions, an investment is needed

Flight Computer System Reference Design



- Dual Architecture HPSC Computer:
 - GP Multicore + Reconfigurable
 - Virtex 5 RH FPGA
- Will benefit from future development:
 - Memory (DRAM)
 - Rad Hard
 - High Density
 - Power Manageable
 - Estimated \$5M NRE
 - 10Gb/S Next Gen Host Bus
 - NGSIS in development

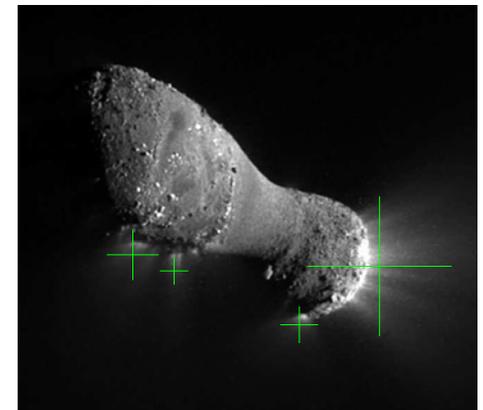
Science Autonomy

Improved performance through HPSC

- Science autonomy enables onboard data processing and response for multiple NASA missions
 - Spacecraft can rapidly identify and respond to key science targets
- Current processor performance limits such capabilities to be run infrequently
 - Currently on MER Mission requires >15 minutes to process one image for science targets
 - Limits capability to be run at most once per day
- Enhanced performance through HPSC could enable several order of magnitude increase in capability use
 - Can run on every image collected
 - Can run continuously during long-range rover traverse
 - Can make key decisions quickly on fast moving platform (e.g., asteroid/comet fly-by)
 - Past studies have shown onboard autonomy could improve number of key science measurements by 5-10X for a Mars Sample Return Mission



Meteorite detection in MER image



Comet plume detection
(Hartley comet captured by EPOXI)

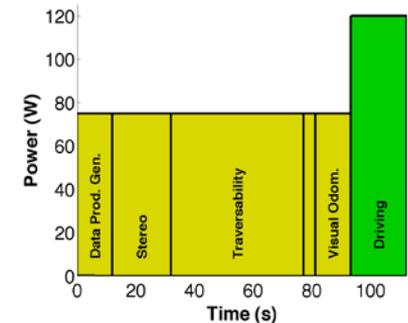
Fast Traverse

Faster, Further and Safer

- Opportunity and Curiosity are slow
 - Vision processing is major bottleneck

Rover	Top Speed	AutoNav Speed	Duty Cycle
MER	133 m/hr	17-25 m/hr	2-3 minutes
MSL	144 m/hr	12-23 m/hr	~2-3 minutes

- Eliminating the computational bottleneck will:
 - **3x Faster:** Allocate more surface time for science
 - **2x Further:** Enable shorter missions, increase sample diversity
 - **Safer:** End the need to ration use of safety features



- Status
 - ~2/3 through porting the portions most amenable for FPGA implementation
 - The vast majority of the code is unsuitable for FPGA but could be accelerated via multi-core.

General-purpose multi-core can host all mobility code

In addition to performance, multi-core offers programmability and V&V advantages