



Next Generation Processing for Space Systems

Raphael R. Some

Chief Technologist: Autonomous Systems Division

Caltech JPL

NASA Technical Authority: High Performance Spaceflight Computing

rsome@jpl.nasa.gov

The work described in this publication was performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract from the National Aeronautics and Space Administration under the STMD GCD Program. Government sponsorship acknowledged © 2018, California Institute of Technology.

Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.



Jet Propulsion Laboratory
California Institute of Technology

The HPSC “Chiplet” Abstract

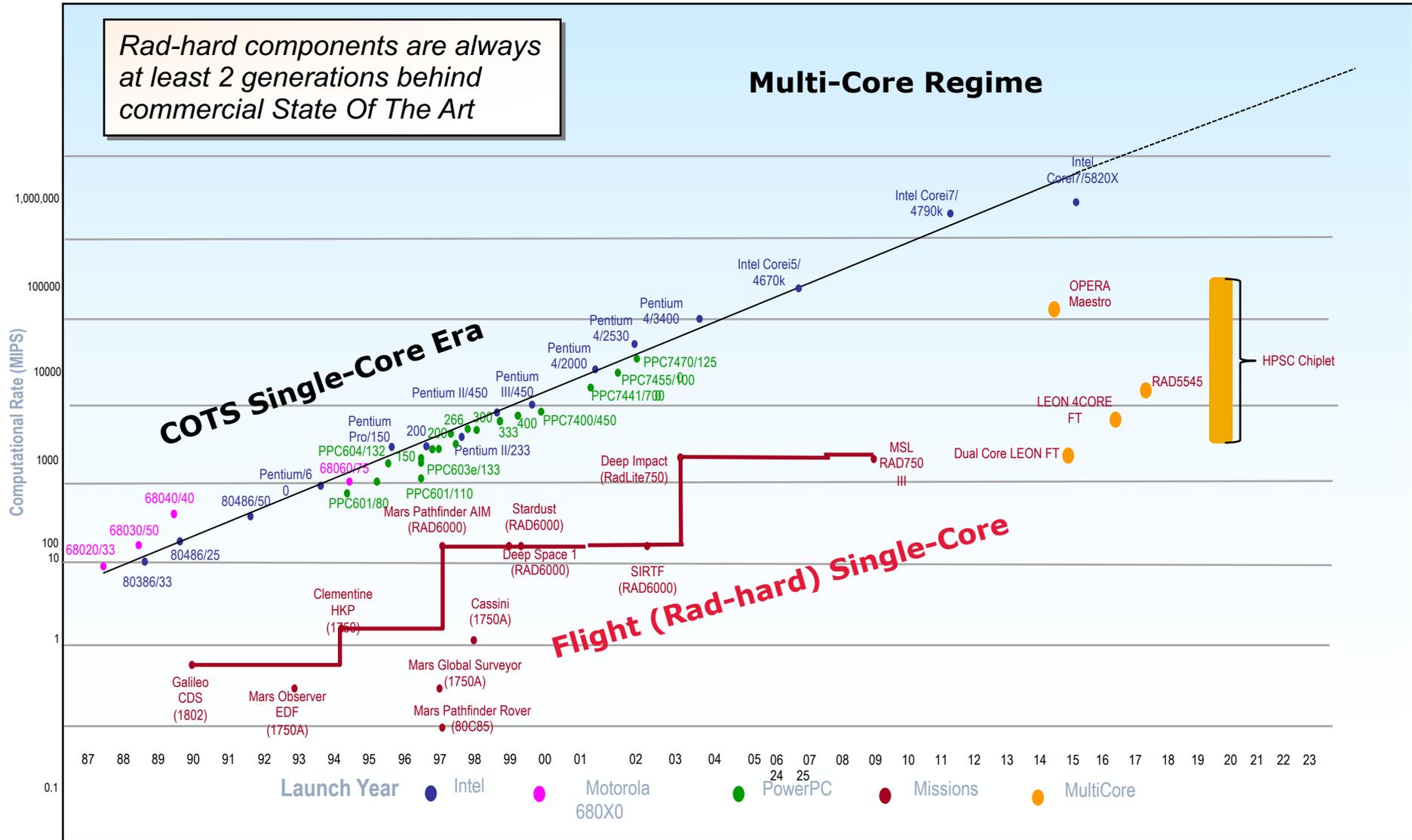
- Future NASA missions will require autonomous capabilities and onboard science data processing in order to overcome communications limitations including link bandwidth and round trip speed of light delays.
- Future United States Air Force (USAF) missions have similar requirements.
- The joint NASA-USAF High Performance Space Computing project (HPSC) is developing a radiation-hardened, fault-tolerant, modular processing element, termed “The Chiplet” to address these future needs.
- The Chiplet concept seeks to develop a family of processors and associated system software that enables “plug and play” (PnP) “system in a package” (SIP) implementations of advanced rad hard computing architectures at an affordable development and deployment cost.
- The HPSC Chiplet project is the first element of this processor family and, if successful, is expected to lead to the development of an “ecosystem” comprising additional heterogeneous processor Chiplets, memories, network elements, operating systems, middleware libraries, software development systems, and the packaging technologies required to achieve 2.5 and 3D SIP space-based computing and avionics systems.
- In this talk we will discuss the concepts behind the HPSC Chiplet, the Chiplet architecture and specifications, the envisioned HPSC ecosystem, current status, and future plans.

Agenda

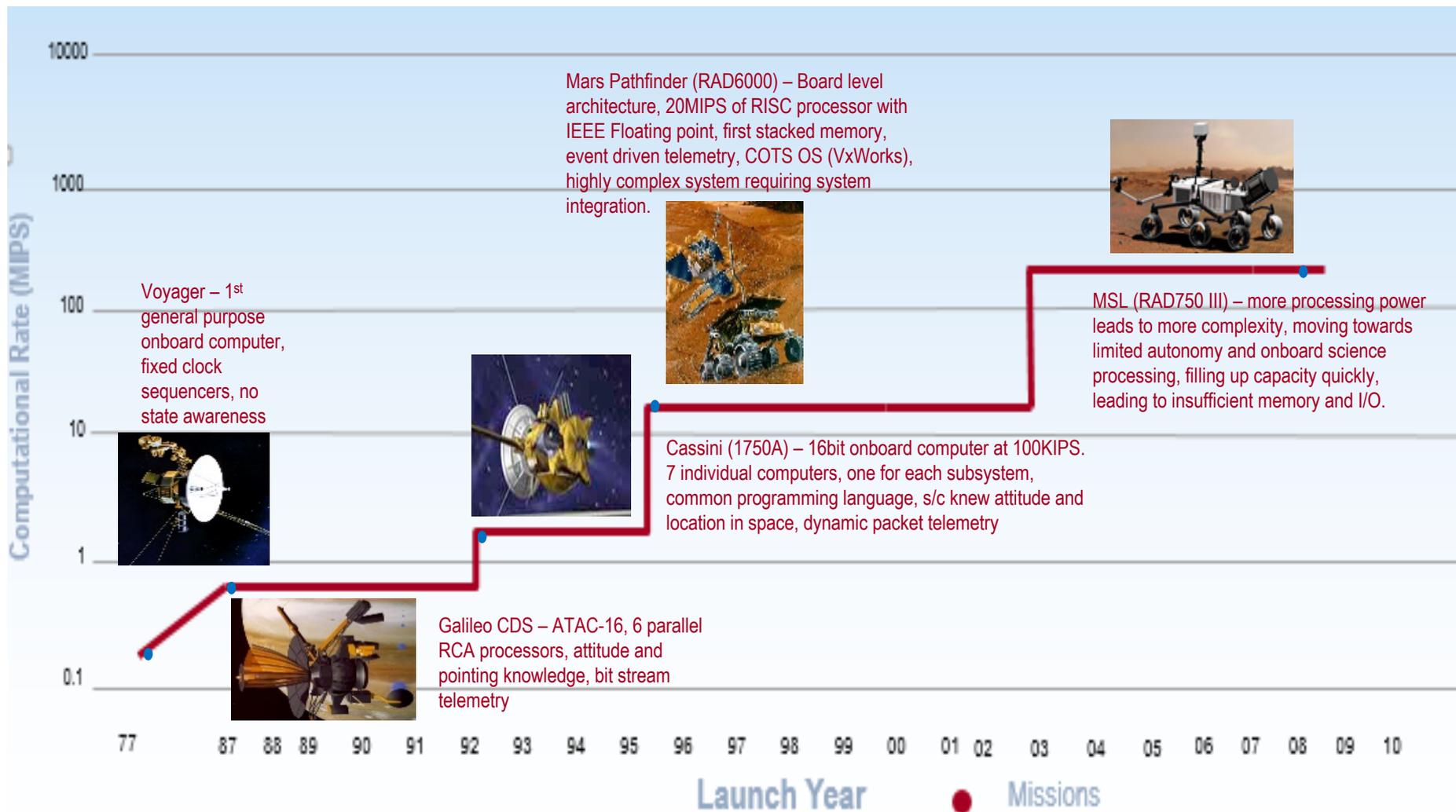
- Brief history of space computing at NASA/JPL
- Need for high performance computing in future NASA space systems
- A perspective on high performance computing and autonomy
- What will be required to achieve the desired capabilities
- Brief review of current space HPC offerings
- Emerging concepts: Chiplet and system in package
- Chiplet overview
- Specifying power, performance, reliability, and radiation tolerance in multicore
- HPSC ecosystem & forward planning
- Q&A

Brief history of space computing at NASA/JPL

JPL Space Flight Avionics and Microcomputer Processor History



Each New Generation Of Processors Enabled New Spacecraft Capabilities (JPL)



Need for high performance computing in future NASA space systems

What new capabilities will we need for future missions

2012 Use Case Study

Human Crewed Missions

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

Robotic Science Missions

1. Extreme Terrain Pinpoint 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

By 2016 We Added New Autonomy Capabilities Requiring Machine Learning, Neural Nets, Model Based AI and Symbolic Reasoning

Visions: Robotic Missions

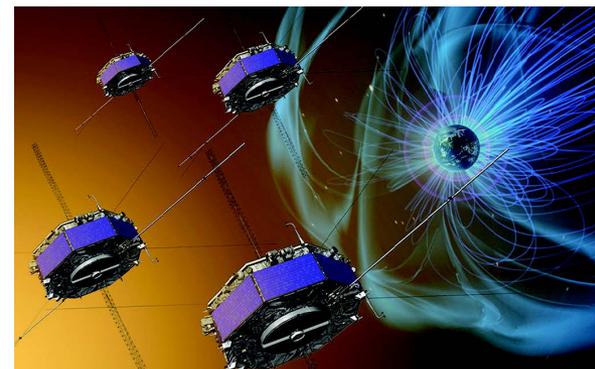
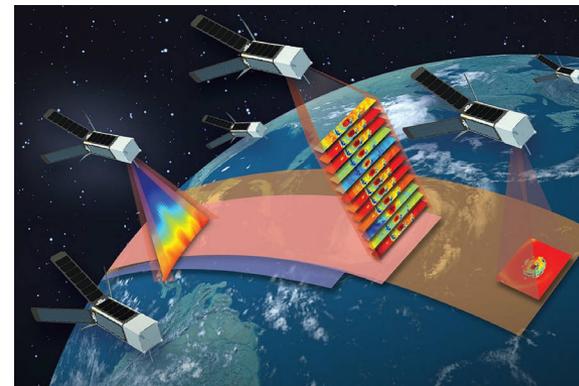
Press this to shut down the instruments, and enter safe mode.



Health management



Goal driven behavior



Constellation & formation flying



Onboard image processing



Mission planning and execution under changing circumstances

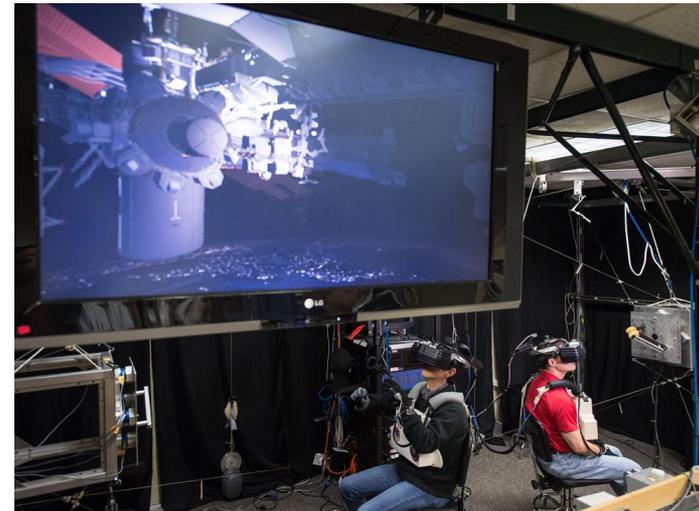
HPC and Autonomy for Robotic Science and Exploration

- Hierarchy of Autonomy
 1. Take care of yourself (*state awareness, fault handling*)
 2. Perform defined mission (*mission planning and execution*)
 3. Determine what to do in order to perform extended exploration and science, i.e.,
 - a. beyond explicitly specified (*goal driven mission planning, opportunistic science*)
 - b. with available resources (*system capability knowledge*)
 - c. in the changing environment (*situational awareness*)
 4. Cooperate with other robots (*constellation, team, swarm operations*)
 5. Science data processing onboard, return *knowledge*

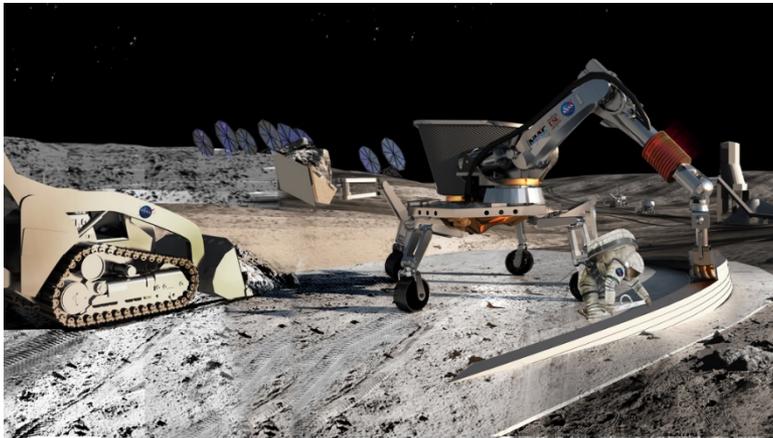
Visions: Crewed Missions



Autonomous repair



Virtual reality, augmented reality



Work independently with minimal direction



Interact with humans on a task

HPC and Autonomy for Crewed Missions

- Hierarchy of Autonomy
 1. Take care of vehicle and crew
 - a. Astronaut assist
 - b. System health monitoring and diagnostics
 - c. Fault handling
 - d. Maintenance and repair
 2. Make it seem like they are at home
 - a. Virtual reality
 - b. Internet in space and delay tolerant communication
 - c. Augmented reality
 - d. Visualizations of missions, remote crew, and robotics operations
 3. Work independently from the human crew with minimal direction
 - a. Repairs, building habitats, or remote stations...
 - i. Teleoperation and virtual presence
 4. Work with the crew in a mixed team, understanding what to do with minimal direction and with maximal safety of humans
 - a. Medical lab, surgeries

**What Will Be Required to
Achieve These Capabilities?**

Application Processing Requirements Robotic Missions

Autonomous
Mission
Planning
1 GOP,
100Mb/S

Extreme Terrain
Landing
25-50GOPS,
30-50Mb/S

Disaster
Response 1-
10GOPS,
200Mb/S-1Gb/S

Radar – Science
10-100sGOPS,
1Gb/S

Hyperspectral
Imaging
10-100GOPS,
Multi Gb/S

Application Processing Requirements Crewed Missions

Advanced Vehicle
Health Management

10-50 GOPS,
10-50Gb/S

Crew Knowledge
Augmentation System

20-50 GOPS,
50-100Gb/S

Augmented Reality

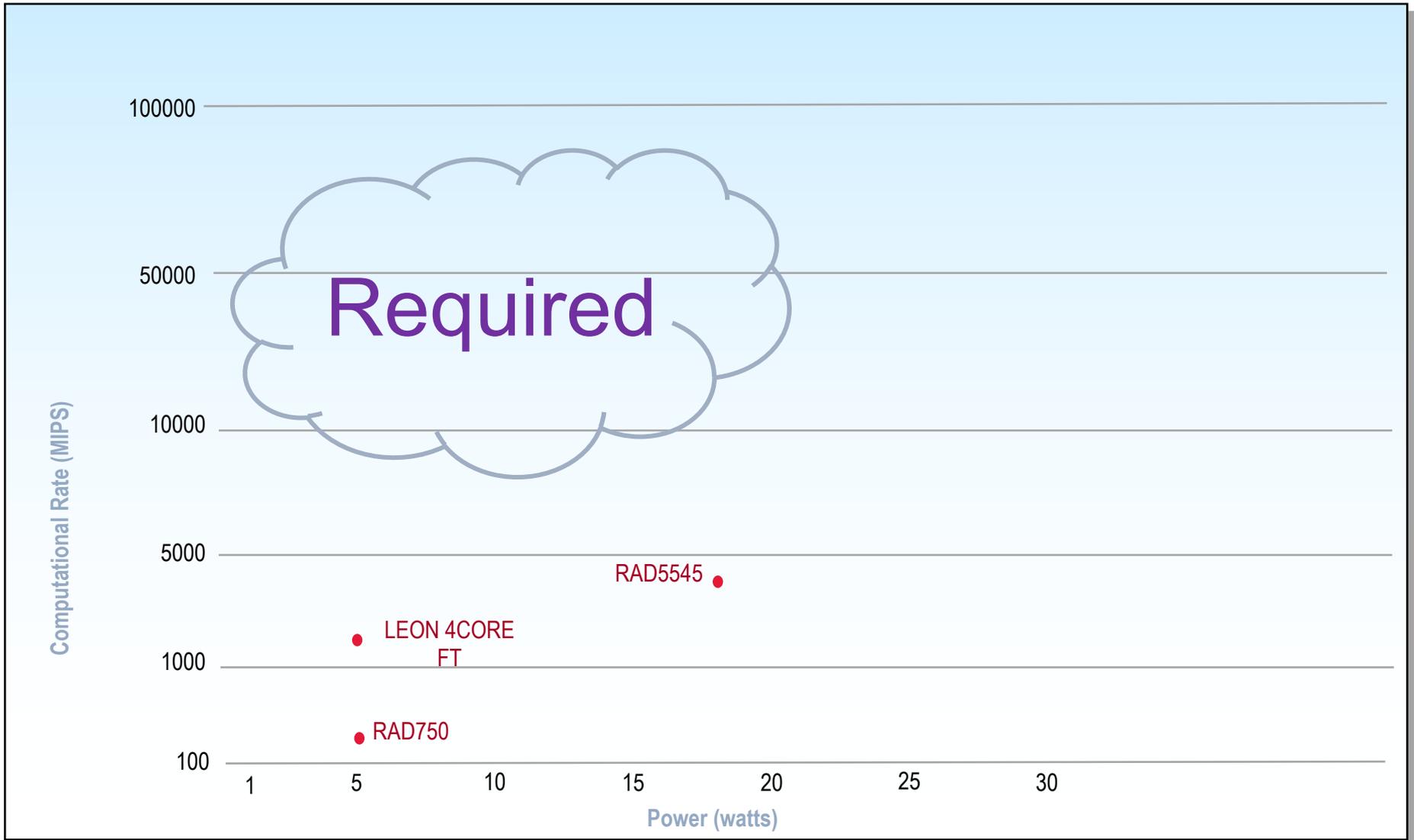
20-60 GOPS,
100+Gb/S

Telepresence

20-60 GOPS,
100+ Gb/S

Brief review of current Space HPC offerings

Multicore Rad Hard Processors Performance: Power



Current Multicore Processors

- Leon 4 & variants, RAD 5545 “System on a Chip” Architectures
 - Self contained
 - Complete system, but limited extensibility
 - 4 processing cores + I/O + memory interface
 - Limited power management
 - Limited fault tolerance strategies
 - Limited resource utilization strategies
 - Binding of “subsystem” to processing core
 - CDH, GnC, Comm, Instrument processing & control
 - Classical SMP
 - Allocation of processor core to next task or thread
 - Other strategies, e.g. AMP, possible, but limited benefit vs complexity
 - Bottlenecks can be a significant issue depending on application
 - Especially memory
 - Straightforward programming with standard OS, compilers, debuggers

What do we want?

What are we looking for in a next gen processing capability? Everything we've discussed thus far.....

- Performance: 10 -100x current capability
- Performance: Power 10-100x current capability
- Dynamic power management capability
- Real time processing capability
- High performance, heterogeneous parallel processing capability
- Low cost in development and in use
- Highly reliable under extreme environmental conditions (Fault Avoidance)
 - Radiation, Temperature, Power fluctuation,.....
- Fault Tolerance and graceful degradation
- Long life – 20+ years of continuous use
- Ease of use
 - Hardware configuration for specific applications
 - Software programming
 - V&V

.... And More

- Future Proofing via modularity and incremental upgradability
 - Evolvability - generational improvement over time
 - Semiconductor process node
 - Next gen IP cores
 - Extensibility - addition of custom coprocessors and accelerators
 - Digital Signal Processors (DSP)
 - Neuromorphic processors
 - Graphic Processing Units (GPU)
 - Processor In Memory (PIM) extensions
 - Software forward compatibility
 - Software compatibility with, and similarity to, terrestrial systems
- Availability
 - Hardware and software elements available from multiple vendors
 - Chips
 - Modules & boards
 - Operating Systems
 - Software development systems
 - Ensured interoperability between vendor products
- Complete, competitive, self sustaining, and self-evolving ecosystem

Emerging Concepts

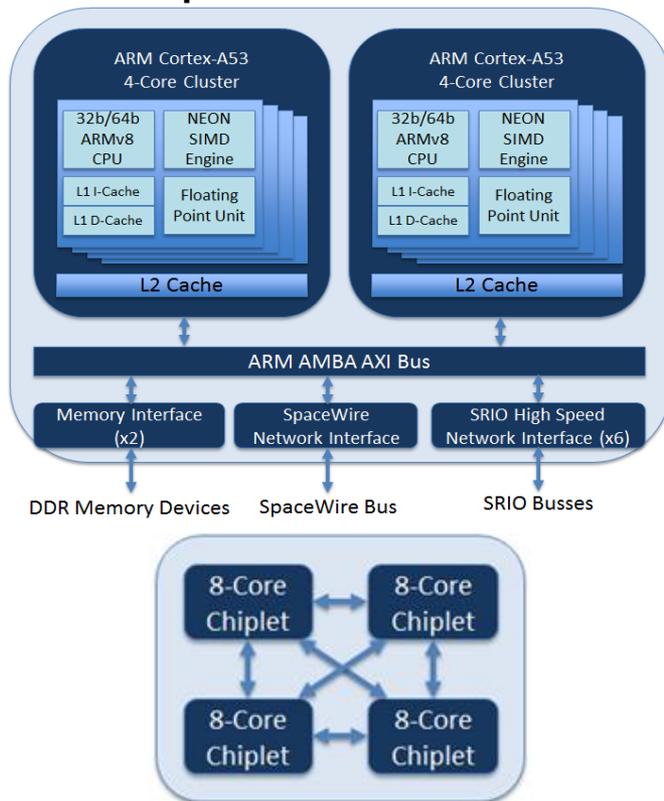
Chiplet

System In Package

The “Chiplet” Concept

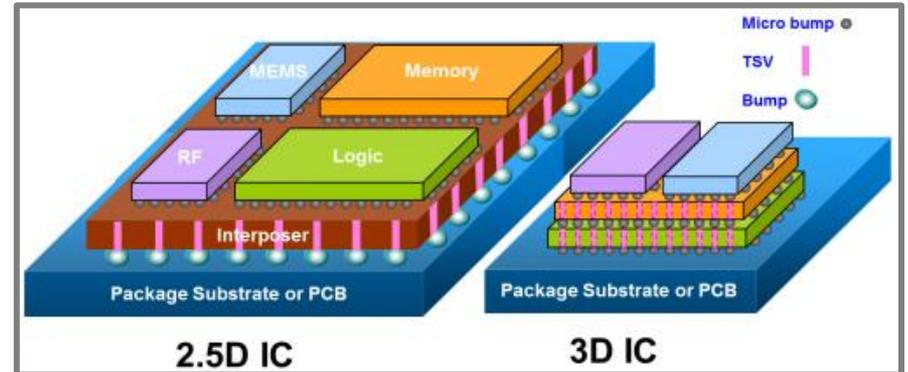
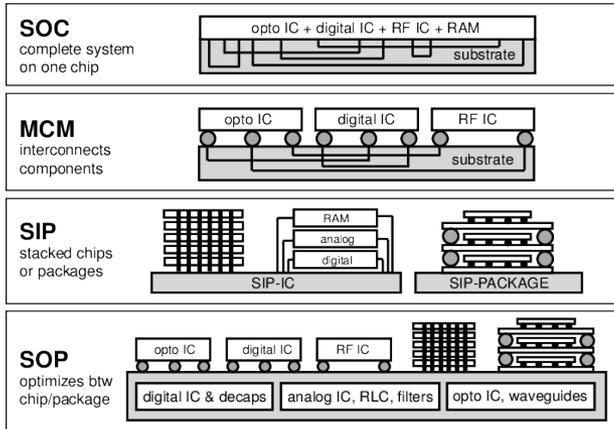
A Chiplet is a small “system on a chip” that provides sufficient capabilities to be used as a stand alone processor OR joined to other Chiplets in order to build a larger, more capable processor.

8 Core Chiplet - Original HPSC Concept



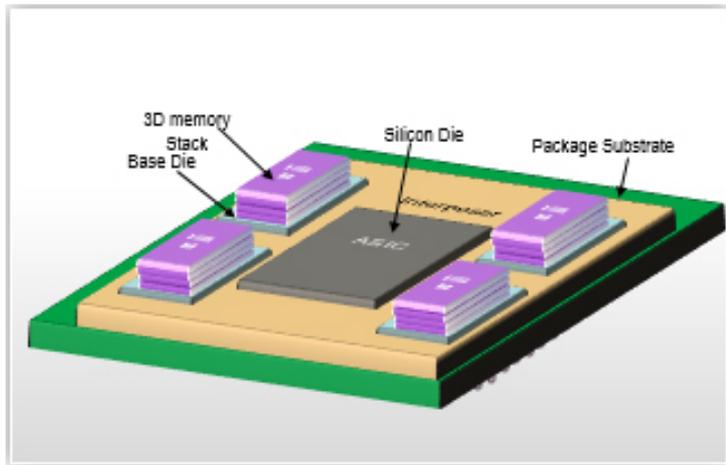
- 2 Arm A53 clusters for high performance processing ~15 GOPS GP + Up to 100 GOPS SIMD Processing
- Typical device power is 7-10 Watts (depending on memory and I/O utilization)
- Power Management – unused cores can be dynamically de-powered or put to sleep
- 2 DDR3/4 memory interfaces with ECC
- 6 Serial RapidIO (SRIO) busses to interconnect other Chiplets, and high bandwidth instruments and subsystems
- 2 SpaceWire Ports
- Misc I/O: NVM, SRAM, GPIO, Boot ROM, SPI, I2C,...
- Multiple levels of fault tolerance – hardware and software implemented – some mandatory, some optional

System on Chip vs System in Package

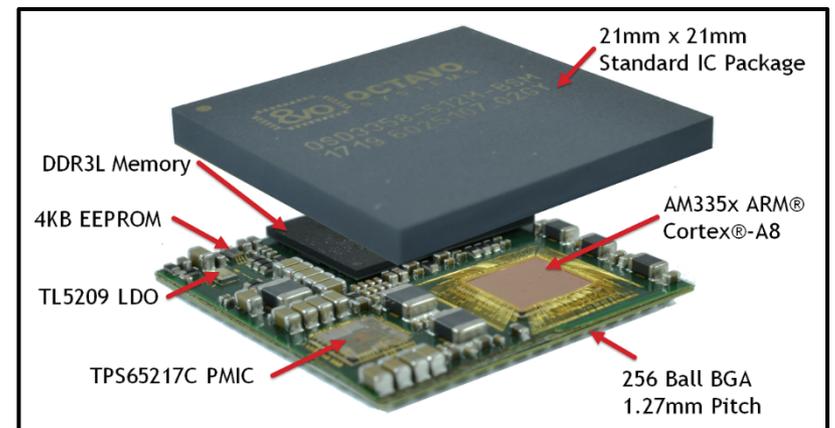


Source: <https://www.skorpiosinc.com/3d-memory-chips-may-beat-3d-hybrid-memory-cube/>

Source: https://www.researchgate.net/figure/Comparison-among-SOC-System-On-Chip-MCM-Multi-Chip-Module-SIP-System-In-Package_fig1_228870734



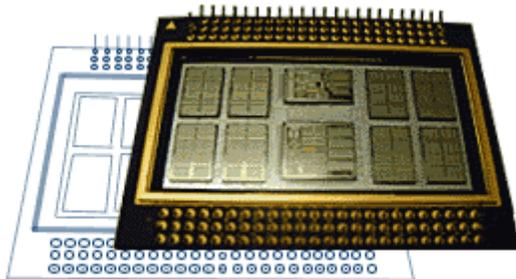
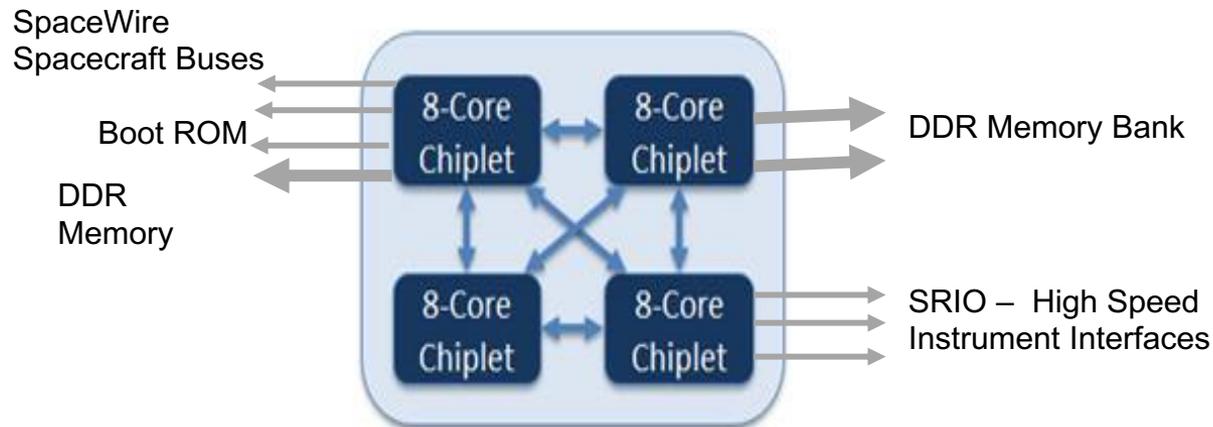
Source: <https://www.chipestimate.com/Is-your-chip-hot-Need-more-bandwidth/eSilicon/Technical-Article/2015/12/22>



Source: https://octavosystems.com/octavo_products/osd335x-sm/

System In Package (SIP)

- A Chiplet based SIP can utilize: 2, 2.5D, or 3D packaging
 - Puts multiple Chiplets, memory chips/stacks, FPGAs, in a single module
 - Provides high density, low impedance “wiring”
 - Allows the use of very low power, high speed drivers between Chiplets
 - Mimics a single large SOC



<https://semiengineering.com/advanced-packaging-picks-steam/>

- Only required interfaces brought out
- Unused interfaces and cores powered off
- Up to 32 processor cores available
- Partitioning and function allocation between Chiplets:
 - Overall control & S/C Real Time
 - Instrument I/O
 - High Performance Parallel Processing

So, Putting it All Together: Chiplet + SIP

- System in Package (SIP) vs System on a Chip
 - Build complex systems from small, reusable modules, aka Chiplets
 - Flexibility/Scalability
 - Multiple Chiplets in arbitrary topology
 - Mix & match Chiplet technologies/generations
 - Multiple modes/levels of fault tolerance with dynamically manageable power
 - Single Chip, 2.5D, 3D packaging (& Chiplet configurations)
 - Extensibility
 - Coprocessor Chiplets: PIM, Neuromorphic, Robotic, DSP
 - FPGAs
 - Evolvability
 - Low cost, rapid evolution of:
 - Chiplets
 - Chiplet-based SIP Computers
 - Affordability
 - Low cost, rapid development
 - Chiplet
 - Processor/computer, module/board

The Chiplet SIP Concept - Software Considerations

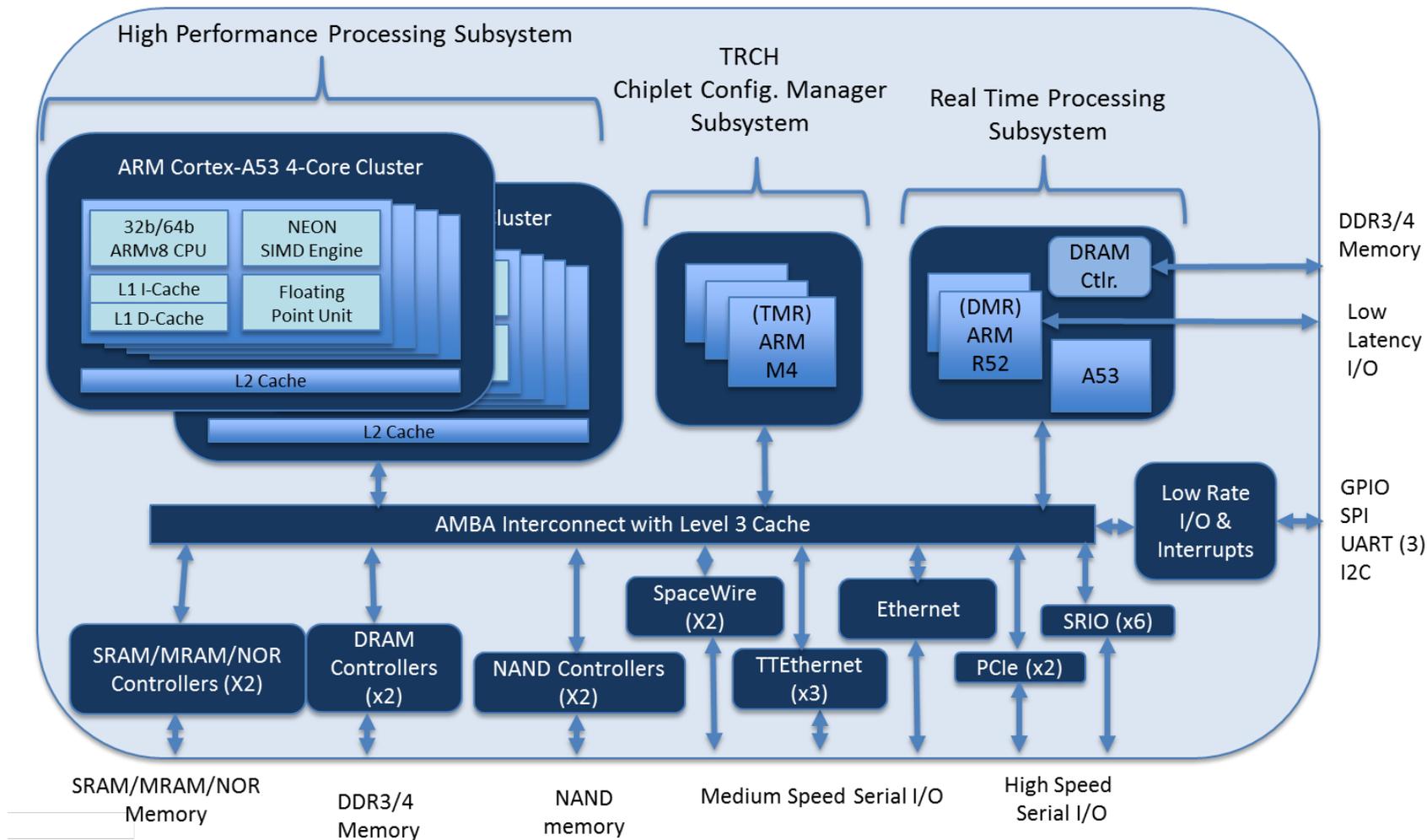
- Software complexity
 - Software needs to handle
 - Multiple widely varying hardware configurations and capabilities
 - Parallel processing (not just multithreading)
 - Dynamically varying hardware resources
 - Dynamically varying software loads with different optimization strategies
 - Software development
 - Rapid (10M+ LOC in 2-3 years)
 - Highly reliable
 - Software modularity, reuse and incremental V&V
 - Near 100% V&V coverage
 - Spacecraft System Level
 - Distributed computing (as well as centralized)
 - Fault tolerance
 - Code migration

HPSC Chiplet Overview

HPSC Program Overview

- Following a competitive procurement, the HPSC cost-plus fixed-fee contract was awarded to Boeing
- Under the base contract, Boeing will provide:
 - Prototype radiation hardened multi-core computing processors (Chiplets), both as bare die and as packaged parts
 - Prototype system software which will operate on the Chiplets
 - Evaluation boards to allow Chiplet test and characterization
 - Chiplet emulators to enable early software development
- Five contract options have been executed to enhance the capability of the Chiplet
 - On-chip Level 3 cache memory
 - Dual real-time processors
 - Dual Time Triggered Ethernet (TTE) interfaces
 - Dual SpaceWire interfaces
 - Package amenable to spaceflight qualification
- Contract deliverables are due April 2021

Current Concept HPSC –8-Core Extensible Chiplet



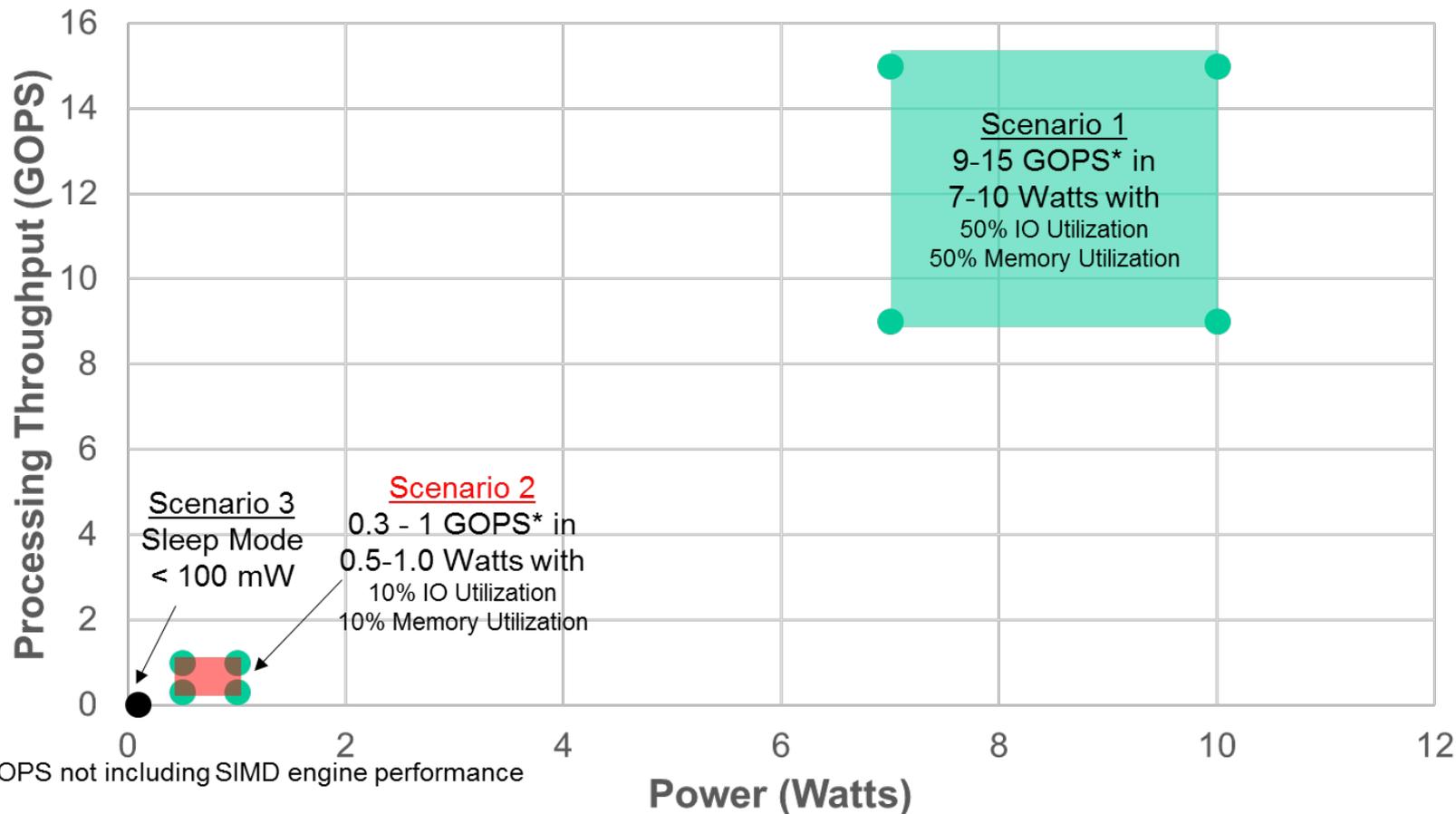
HPSC Chiplet Program Overview

Key Requirements Summary

Processor Cores	<ul style="list-style-type: none"> HPP Subsystem: 8 ARM Cortex-A53 cores with floating point & SIMD engine. Realtime Processing Subsystem (RTPS) with single A53 and dual Cortex-R52 cores
Memory Interfaces	<ul style="list-style-type: none"> 3 DDR3/4: 2 for A53 clusters, 1 for RTPS 4 SRAM/NVRAM Enhanced error correction (ECC) to operate through bit upsets and whole memory device failures
IO Interfaces	<ul style="list-style-type: none"> 6 SRIO 3.1, 2 PCIe Gen2 serial IO Ethernet, SpaceWire, TTE, SPI, UART, I²C, GPIO
Power scaling	<ul style="list-style-type: none"> Able to dynamically power down/up cores, subsystems, & interfaces via software control
Fault tolerance	<ul style="list-style-type: none"> Able to autonomously detect errors & log errors, prevent propagation past established boundaries, and notify software
Trust & Assured Integrity	<ul style="list-style-type: none"> DMEA-accredited Trusted supply chain Free of malicious insertions / alterations
Temperature	<ul style="list-style-type: none"> -55C to 125C

HPSC Performance at Power Requirements

HPSC Chiplet Performance at Power



HPSC Chiplet Program Overview

Key Requirements Summary

Total Ionizing Dose (TID)	Strategic radiation hardness for Air Force applications
Prompt Dose Immunity	
Dose Rate Survivability	
Latchup Immunity	<ul style="list-style-type: none"> • LET \geq 90 MeV-cm²/mg
Single-Event Upset (SEU) (Adams 90% WC GEO)	<ul style="list-style-type: none"> • Discussed In Next Section
Single-Event Upset (SEU) (WC Solar Flare)	<ul style="list-style-type: none"> • Discussed In Next Section
Reliability	<ul style="list-style-type: none"> • \geq 100,000 power-on hours
Software	<ul style="list-style-type: none"> • Multicore operating systems (Linux & RTOS) • Development tools (compilers, debuggers, etc) • Board Support Packages (BSPs) • APIs for fault tolerance, power management
Emulators	<ul style="list-style-type: none"> • Software-based quick emulator • FPGA-based cycle-accurate emulator

HPSC Chiplet Program Overview Approach

- Develop Chiplet using Boeing's Trusted and rad-proven RHBD 32nm SOI design & fabrication flow, which provides:
 - High-performance library and mixed-signal macros
 - Strategic radiation hardness
 - Single-Event-Effects (SEE) mitigations optimized for power efficiency
 - Assured integrity
- Employ core competencies of team comprised of world-class organizations:
 - Boeing Solid-State Electronics Development (SSED)
 - Boeing Secure Computing Solutions (SCS)
 - Boeing Space & Launch
 - USC Information Sciences Institute (ISI)
 - University of Michigan ARM Research Center
- Utilize silicon-proven IP from best-in-class suppliers, including:
 - ARM, Globalfoundries, Synopsys, Praesum, and Uniquify
- Leverage tens of millions of dollars of Government and Boeing investments in related technology areas:
 - DTRA RHBD3, AFRL/NASA NGSP, MAESTRO, DARPA PERFECT, etc..

HPSC Chiplet Program Overview

Program Structure & Schedule

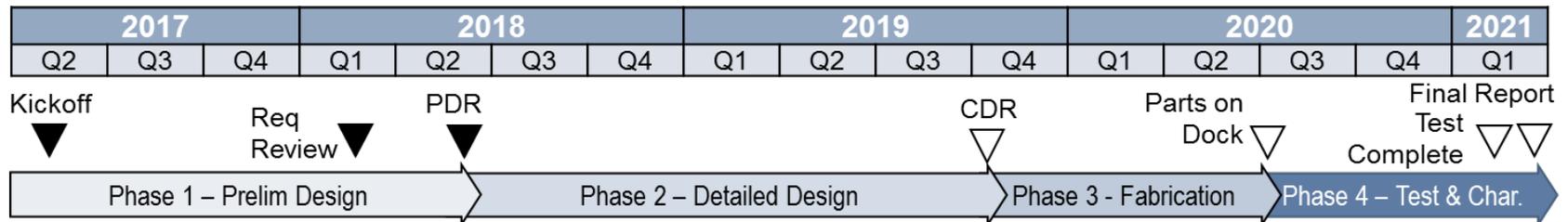
- Tasks (WBS Level 1)

1.0 Management	4.0 System Software Development
2.0 System Engineering	5.0 Evaluation Board Development
3.0 Chiplet Development	6.0 Test and Characterization

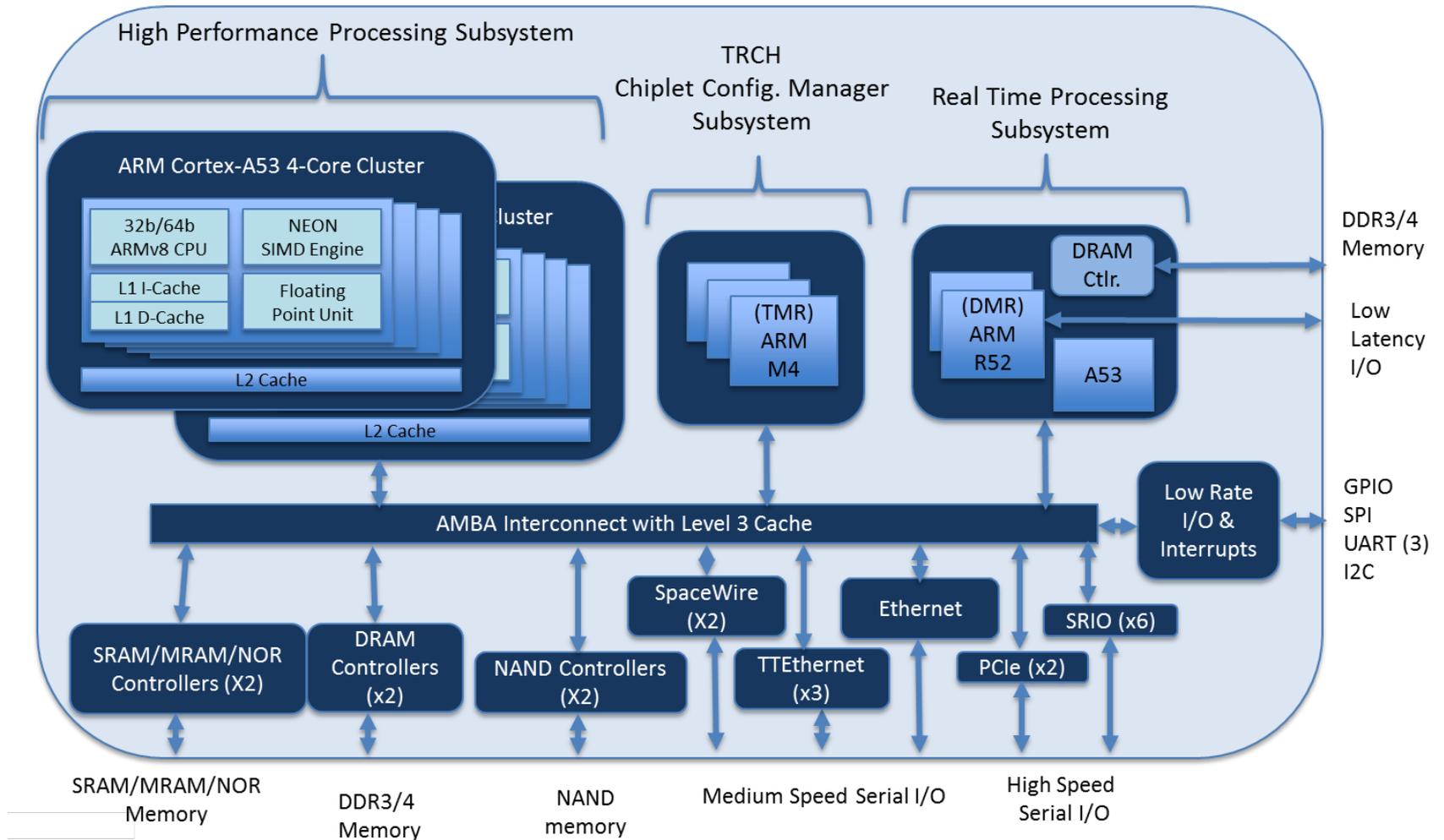
- Phases

Phase	Duration	Period of Performance
1. Preliminary Design	14 months	March 2017 through May 2018
2. Detailed Design	17 months	June 2018 through October 2019
3. Fabrication	9 months	November 2019 through July 2020
4. Test & Characterization	9 months	August 2020 through April 2021

- Schedule



Current Concept HPSC –8-Core Extensible Chiplet



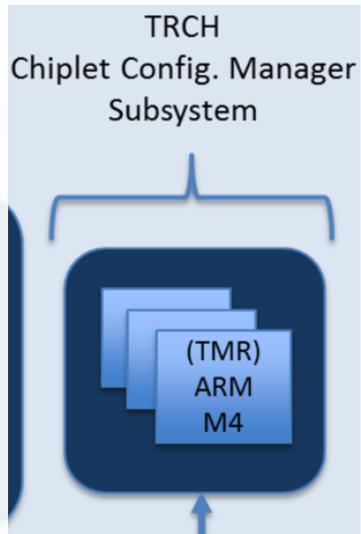
Chiplet Architecture: TRCH Controller

Timing, Reset, Config, & Health (TRCH) Controller

- Built around a small TMR'd Cortex-M4F microcontroller

- Low power and small form factor

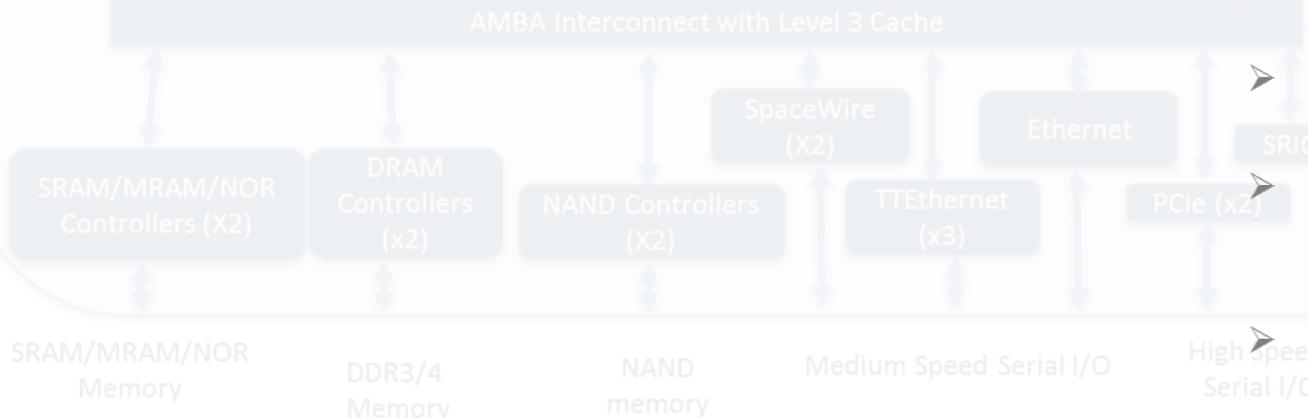
- More flexibility than a finite state machine approach



Real Time Processing

- Provides the following Chiplet functionality:

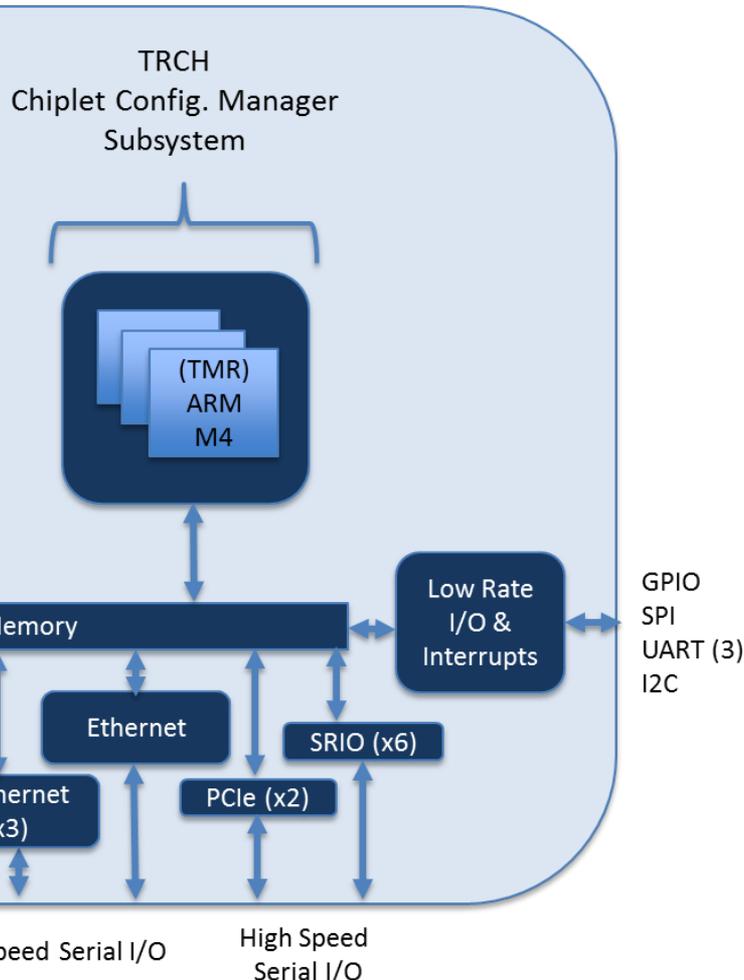
- Power-On-Reset
- Clock Generation
- Power Management
- Configuration Control
- Sleep Mode (< 50mW)
- Health Monitoring / Fault Management
- Built-in-Self-Test, eFuse
- Timers and Synchronization Control
- Low-speed External Interfaces (GPIO, NVRAM, I2C, etc..)



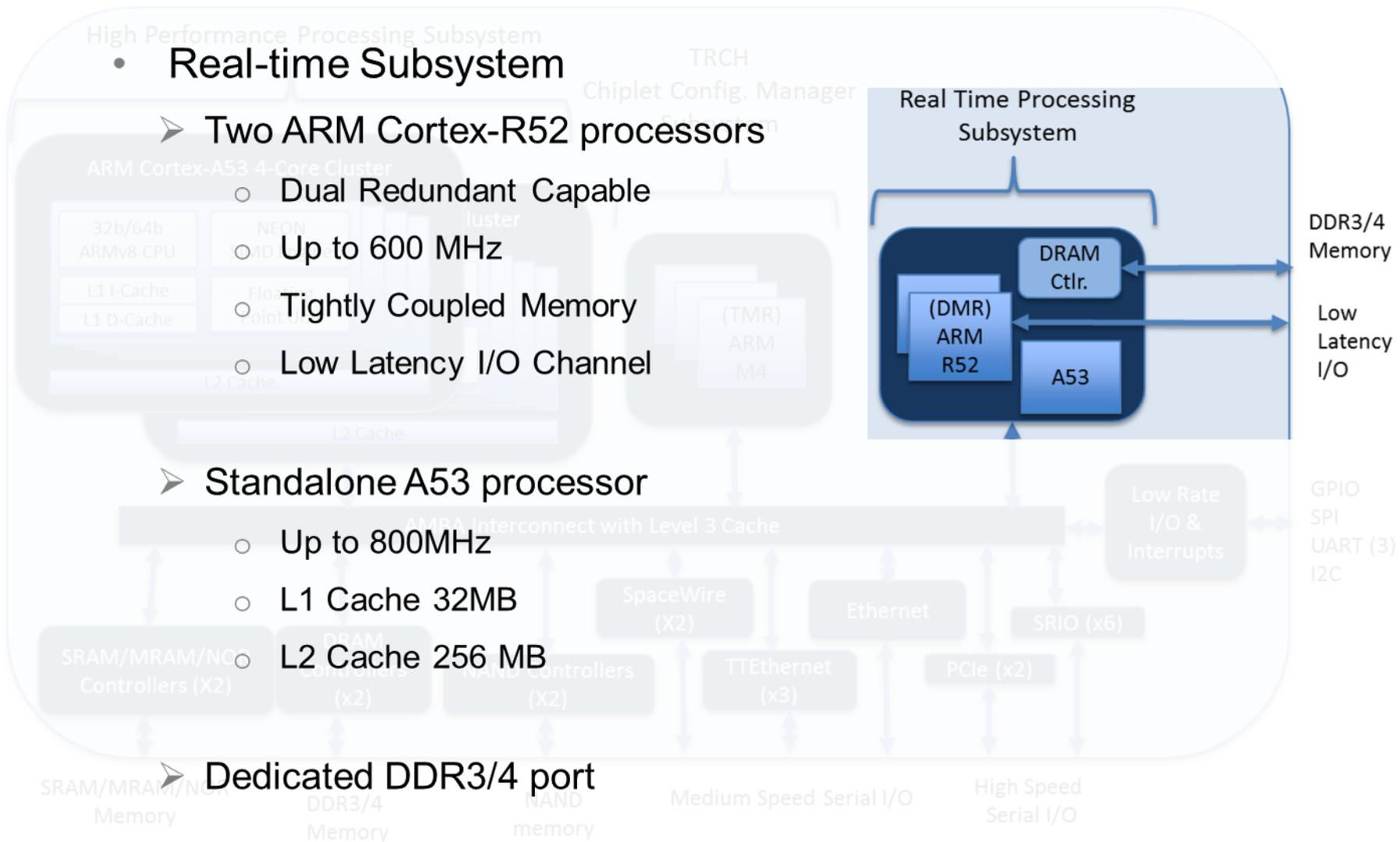
HPSC – TRCH

Timing, Reset, Config, & Health (TRCH) Controller

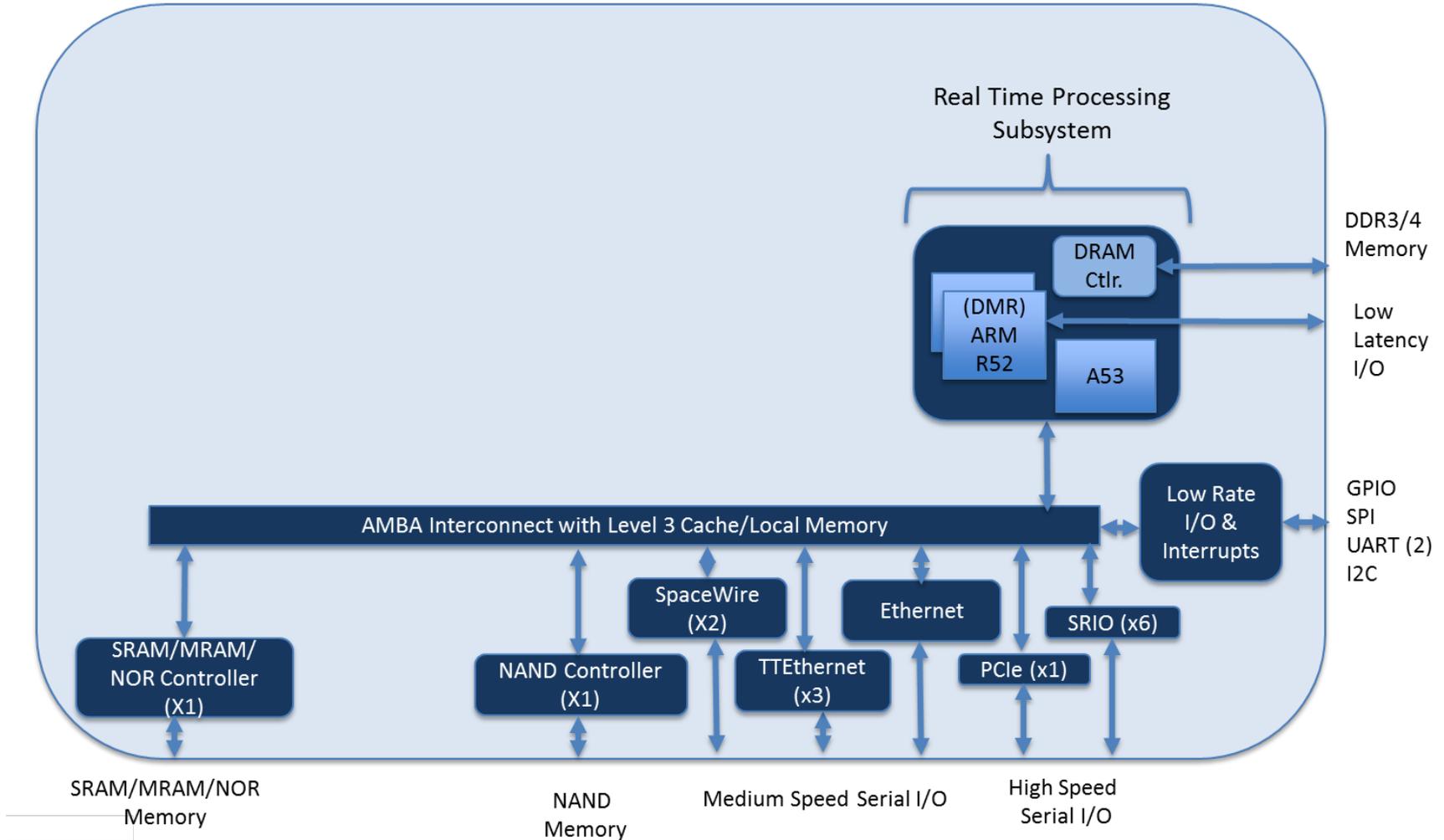
- TMR'd Cortex-M4F microcontroller
 - Low power, small form factor, more flexibility than a finite state machine
- Provides:
 - Power-On-Reset
 - Clock Generation
 - Power Management
 - Configuration Control
 - Sleep Mode (< 50mW)
 - Health Monitoring / Fault Management
 - Built-in-Self-Test
 - Timers and Synchronization Control
 - Low-speed External Interfaces (GPIO, NVRAM, I2C, etc..)



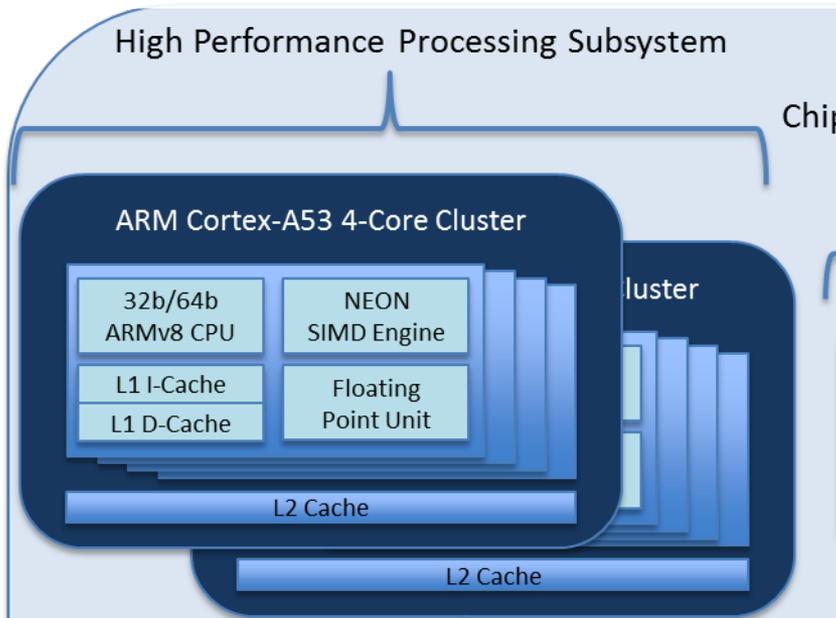
Chiplet Architecture: Real Time Processor (RTPS)



HPSC –Real Time Processor Subsystem



Chiplet Architecture: High-performance Cores



Two (2) ARM Cortex-A53 MPCore Clusters

- Four (4) cores per cluster
- ARMv8 64-bit ISA
- Up to 800 MHz frequency
 - 1840 Dhrystone MIPS (DMIPS) per core @ 800MHZ
 - 7360 DMIPS/Quad Cluster, Total of 14,720 DMIPS
- 256 KByte L2 Cache per quad cluster
- 32 KByte L1 instruction and data cache per core

- Floating Point Unit (FPU) and NEON SIMD engine per core

- NEON Engine – 100GOPS @ 8 bit lanes, 50 GOPS @ 16 bit lanes, 25GOPS @ 32bit lanes, 12.5GOPS @ 64bit lanes

SRAM/MRAM/NOR Memory

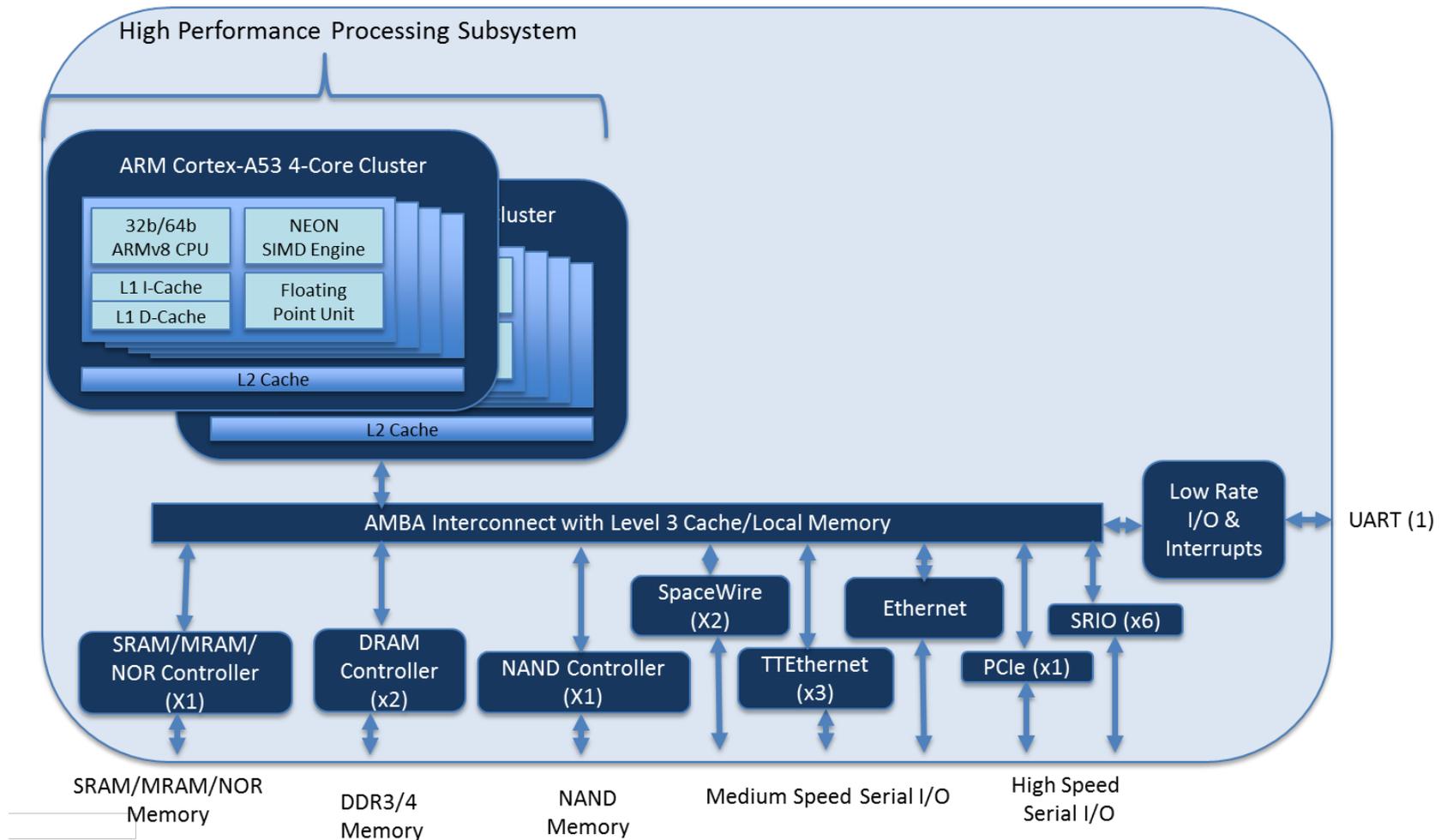
DDR3/4 Memory

NAND memory

Medium Speed Serial I/O

High Speed Serial I/O

HPSC –High Performance Processing Subsystem



Chiplet Architecture: Other Features & Options

- 4MB L3 Cache / Local Memory
- Core Sight & JTAG Interfaces
- 6 Port SRIO Interface (40Gb/s)
- 3 Port Time-Triggered Ethernet (1Gb/s)
- 2 Port SpaceWire Interface (400Mb/s)
- 2 PCIe Interfaces (10Gb/s)
- 10/100 Ethernet
- GPIO
 - Single Ended
 - Differential
- SPI
- 12C
- 3 UART Interfaces
- 2 SRAM, MRAM, NOR Interfaces
- 2 NAND interfaces
- External Interrupts

Software and Ancillary Products

- HPSC System Software
 - HPPS – Linux OS, C/C++, Parallel Trace & Debug
 - RTPS – RTEMS OS, C/C++, Trace & Debug
 - TRCH – RTEMS OS, C/C++, Trace & Debug
- HPSC Chiplet Emulator
 - QEMU (software based emulator)
 - HAPS (FPGA Simulator)
- Evaluation Boards
 - 1 Chiplet per board
 - Full complement of memory and I/O
- NASA Middleware
 - Configuration, power, fault tolerance management
 - Resource allocation
- Additional software tools and products TBD

Specifying Power, Performance, Reliability, and Radiation Tolerance in Multicore Processors

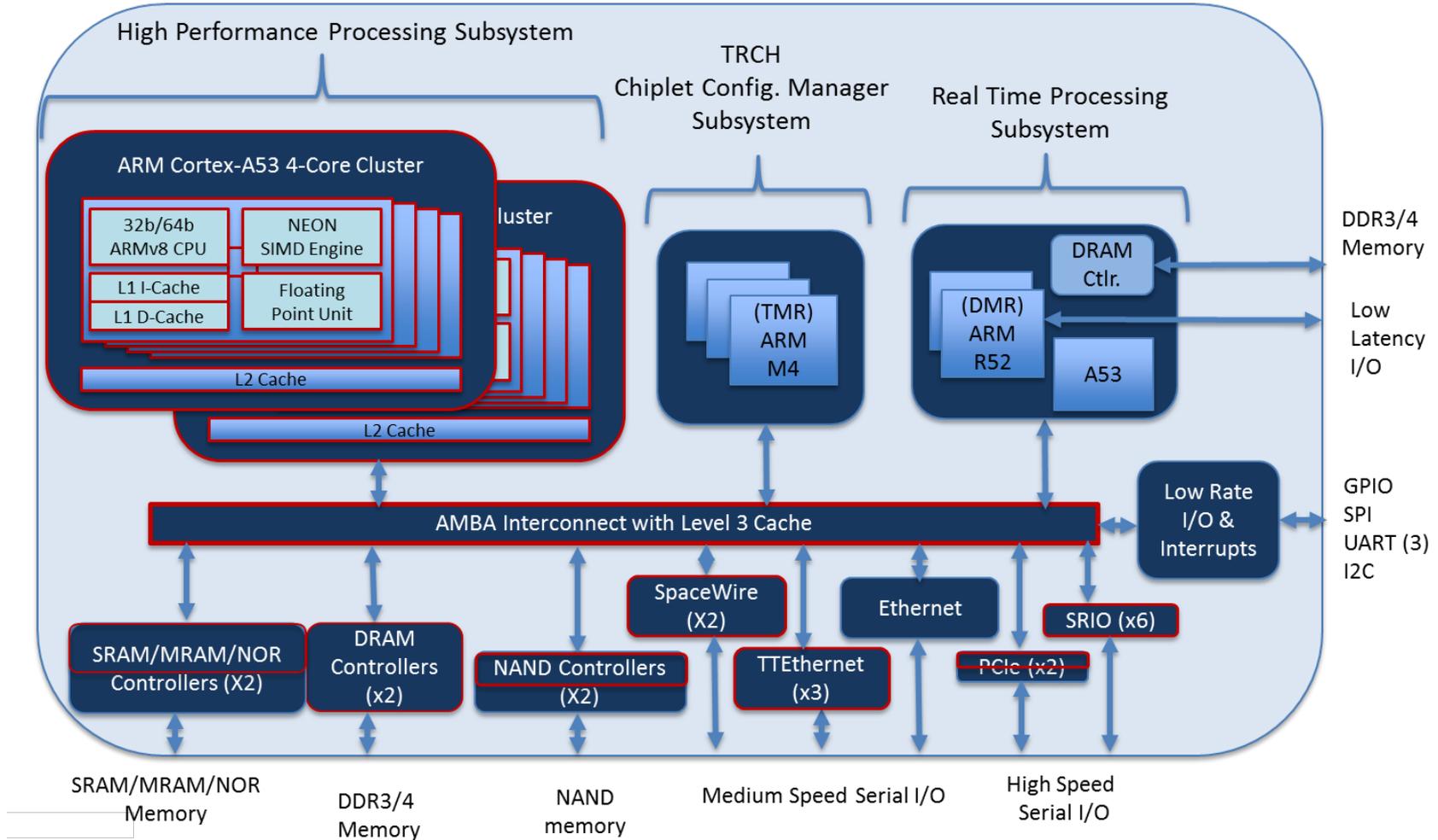
The Problem and Proposed Approach

- Modern many-core SOC's can be configured and used in many ways
- Traditional approach to power, performance, reliability, radiation susceptibility, and fault tolerance, i.e., “worst case analysis”, is not *useful*.
- Proposed approach:
 - Estimate worst case per IP Core (and any associated circuitry)
 - Provide user with models that can be used to estimate power, rad tolerance susceptibility, reliability and performance for a given configuration and usage condition.
 - Tools for IP core estimations already exist
 - Models can be fairly simple and straightforward to build and to use
 - Must account for power off, sleep, disable modes
 - Must account for latent faults during powered but unused (sleep, disable) periods
 - Must account for associations, e.g., cores that must be used together
 - Should account for usage/loading conditions (industry defaults can be utilized)
 - Specify key configurations and usage conditions

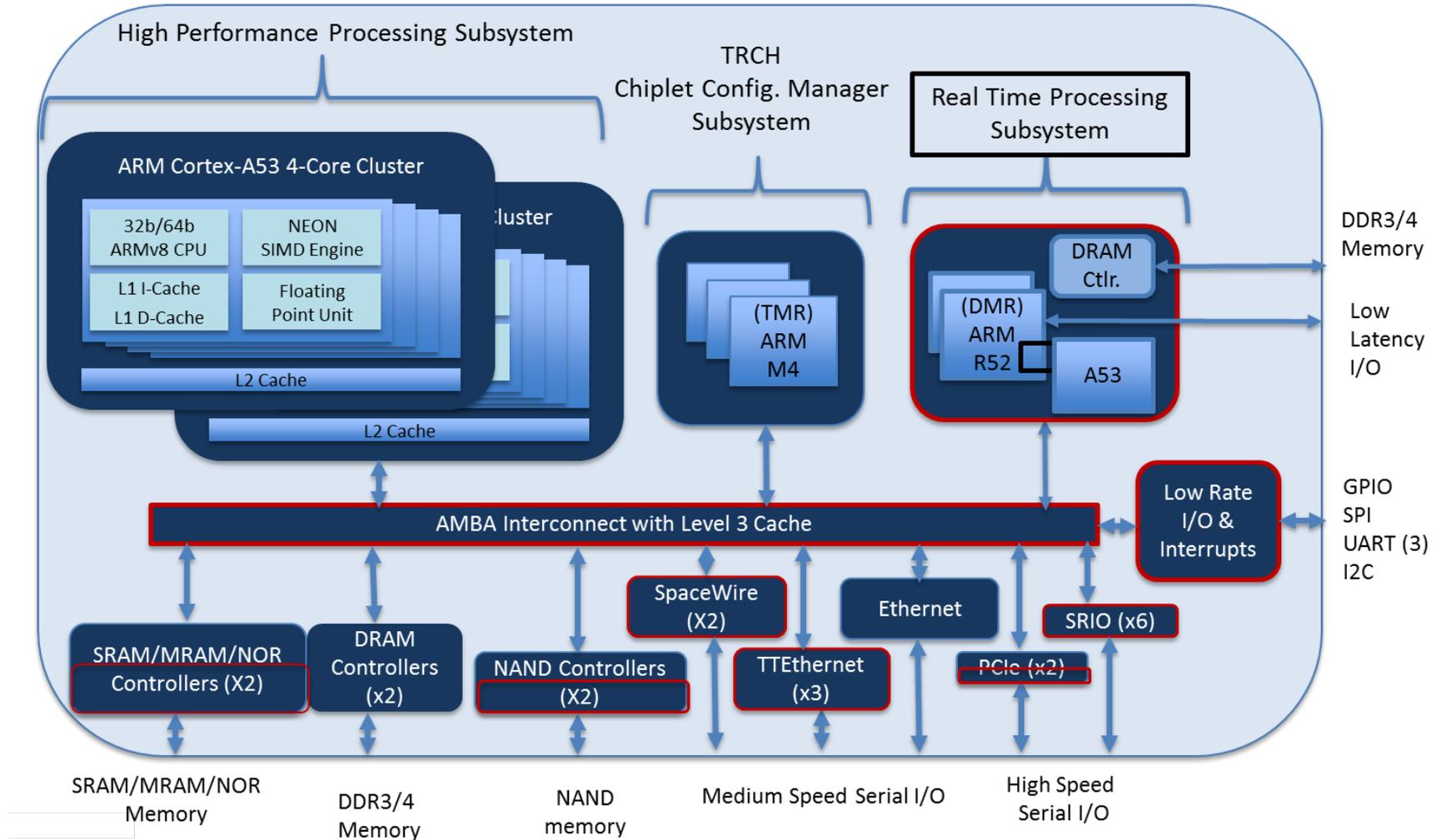
SEE Tolerance Example

- **Traditional Approach: specify the worst case**
 - Everything turned on
 - Errors/bit-day x total bits = errors/device day
 - All processors and interfaces working at max rates
 - Gives unrealistic and overly pessimistic result
- **Proposed Approach: specify at subsystem level, based on usage**
 - Everything in subsystem turned on
 - Allow overlapping elements, i.e., shared resources, are double counted
 - Memory
 - I/O
 - Interrupt Controllers
 - DMA Engines
 - Internal Interconnects
 - Still pessimistic, possibly overly so, but a more realistic system view

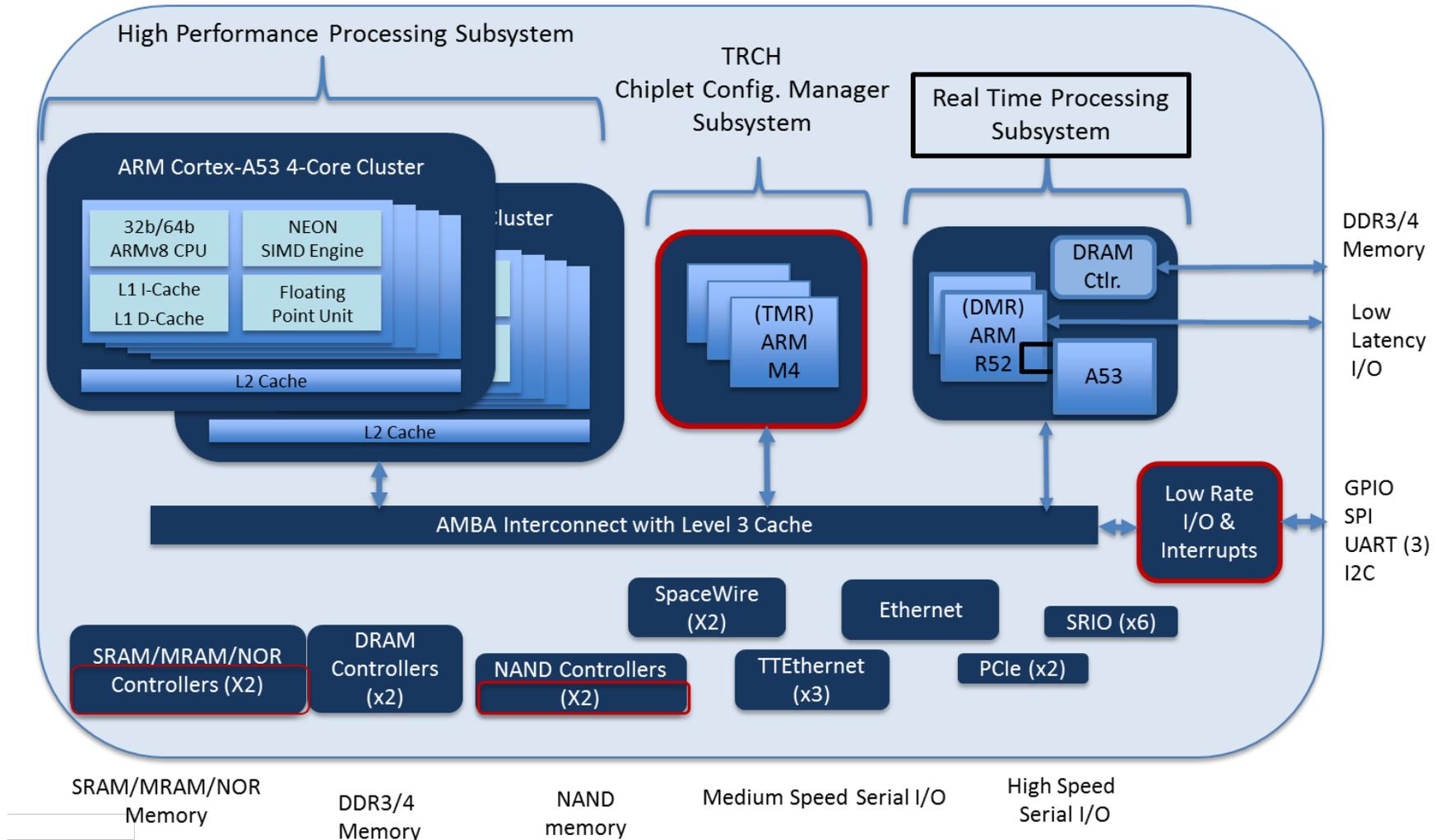
SEE Tolerance Example HPPS (1X)



SEE Tolerance Example RTPS (10X)



SEE Tolerance Example TRCH (100X)



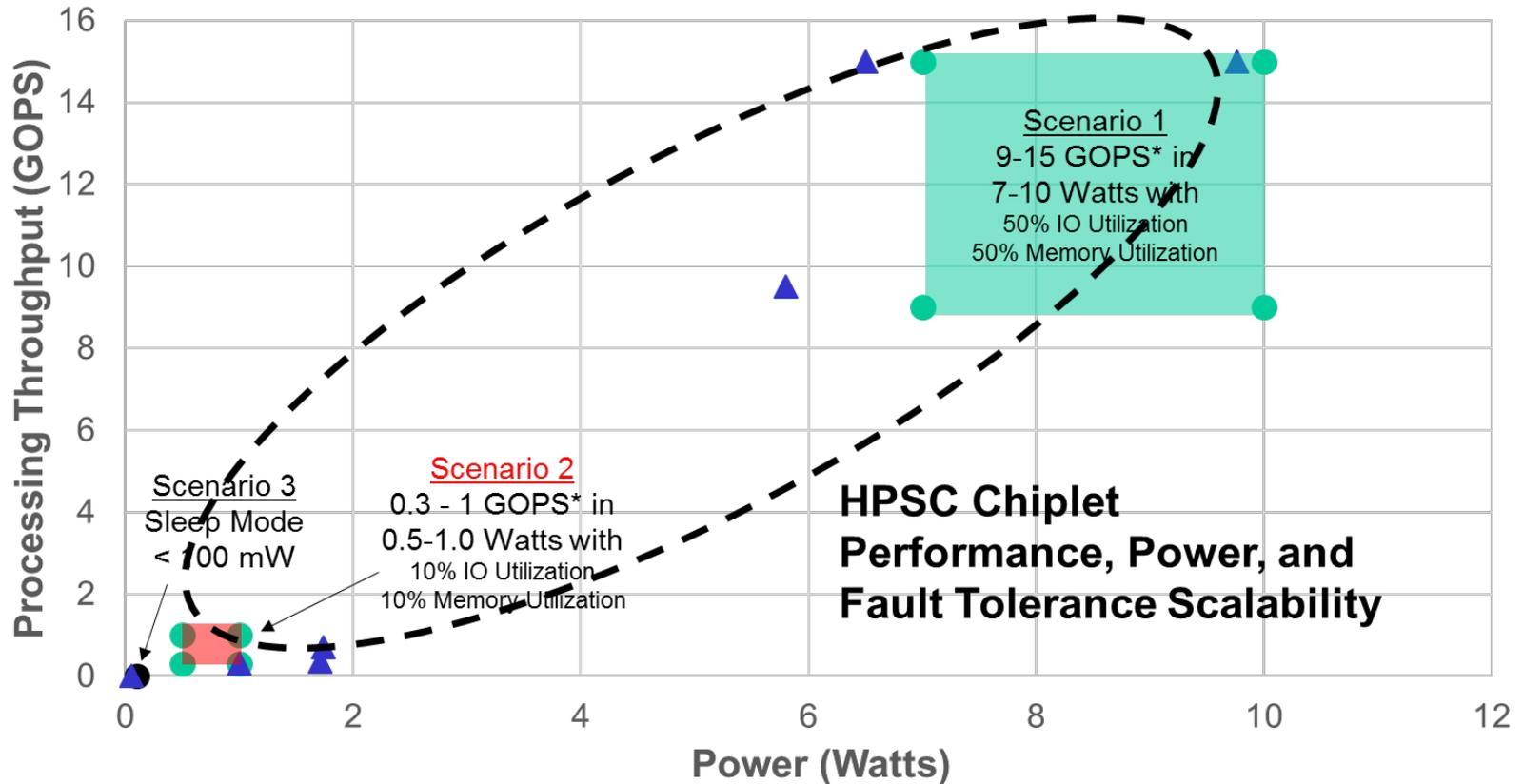
HPSC Chiplet Program Overview

Key Requirements Summary

Total Ionizing Dose (TID)	Strategic radiation hardness for Air Force applications
Prompt Dose Immunity	
Dose Rate Survivability	
Latchup Immunity	<ul style="list-style-type: none"> • LET ≥ 90 MeV-cm²/mg
Single-Event Upset (SEU) (Adams 90% WC GEO)	<ul style="list-style-type: none"> • HP Subsystem (A53 Array): $\leq 1E-3$ errors/device-day • Realtime Subsystem: $\leq 1E-4$ errors/device-day • Timing/Reset/Clock/Health Ctrlr: $\leq 1E-5$ errors/device-day
Single-Event Upset (SEU) (WC Solar Flare)	<ul style="list-style-type: none"> • HP Subsystem (A53 Array): $\leq 1E-1$ errors/device-min • Realtime Subsystem: $\leq 1E-2$ errors/device-min • Timing/Reset/Clock/Health Ctrlr: $\leq 1E-3$ errors/device-min
Reliability	<ul style="list-style-type: none"> • $\geq 100,000$ power-on hours
Software	<ul style="list-style-type: none"> • Multicore operating systems (Linux & RTOS) • Development tools (compilers, debuggers, etc) • Board Support Packages (BSPs) • APIs for fault tolerance, power management
Emulators	<ul style="list-style-type: none"> • Software-based quick emulator • FPGA-based cycle-accurate emulator

HPSC Performance @ Power Requirements and Analyses Example

HPSC Chiplet Performance at Power

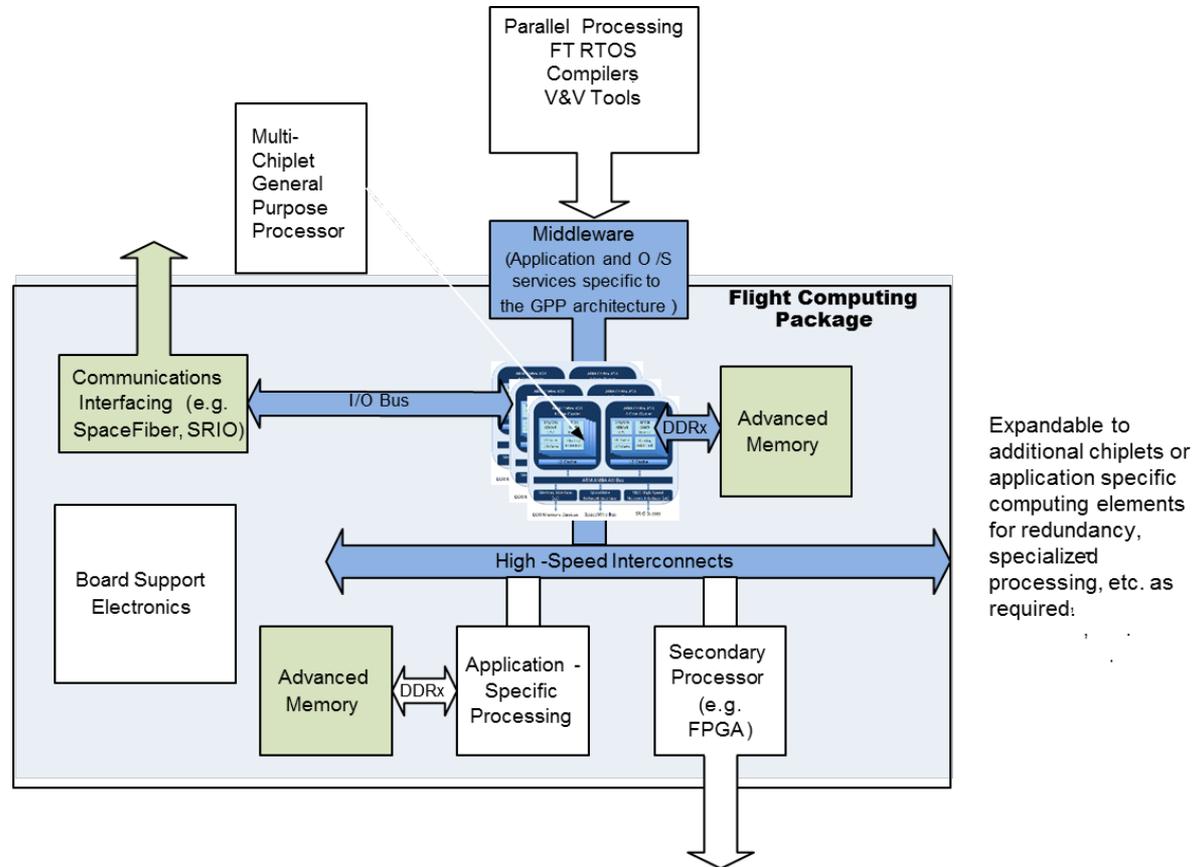


* GOPS not including SIMD engine performance

● Required ▲ Predicted

HPSC Ecosystem & Forward Planning

HPSC Ecological Elements



The overall High Performance Spaceflight Computing (HPSC) architecture is an “ecology” formed by the processor and supporting hardware and software elements to make a modern, scalable, rad hard computing environment.

HPSC

Ecosystem Roadmap

- The HPSC Roadmap partitions the HPSC “ecology” into the following categories:

Advanced Rad Hard Space Memory	High capacity/speed/reliability space qualified, volatile and non-volatile memory components for processor, instrument and mass storage devices
Co-Processors/Accelerators and “Micro-Chiplet”	Custom processors that provide extremely high performance at low power for specialized types of computation including: digital signal processing (DSP), graphics (GPU) and machine vision (MV) processing, deep learning/neuromorphic processing, as well as a microprocessor version of the chiplet for extremely low cost, instrument/subsystem-embedded applications
System Software and Execution Environment	Next gen, parallel, secure operating systems, hypervisors and middleware
Software Development Environment	Compilers, debuggers and V&V tools for high reliability, real time, and parallel codes
Power Supply and Smart Power Bus	Efficiently provides requisite voltages, and intelligent power distribution system required for complex digital systems
Board Level Products	Prototype single board computers, and non volatile memory modules

Summary

- The joint NASA-USAF High Performance Space Computing project (HPSC) is developing a radiation-hardened, fault-tolerant, modular processing element, termed “The Chiplet” to address future space computing needs.
- The Chiplet concept seeks to develop a family of processors and associated system software that enables “plug and play” (PnP) “system in a package” (SIP) implementations of advanced rad hard computing architectures at an affordable development and deployment cost.
- The HPSC Chiplet project is the first element of this processor family and is expected to lead to the development of an “ecosystem” comprising additional heterogeneous processor Chiplets, memories, network elements, operating systems, middleware libraries, software development systems, and the packaging technologies required to achieve 2.5 and 3D SIP space-based computing and avionics systems.
- A new approach to radiation tolerance, reliability, power utilization and performance, specification (and analysis) of complex electronic components divides the component into overlapping “worst case” subsystems and operational configurations.
- For additional information or to discuss potential collaboration,
 - Contact Raphael R. Some rsome@jpl.nasa.gov



Jet Propulsion Laboratory
California Institute of Technology

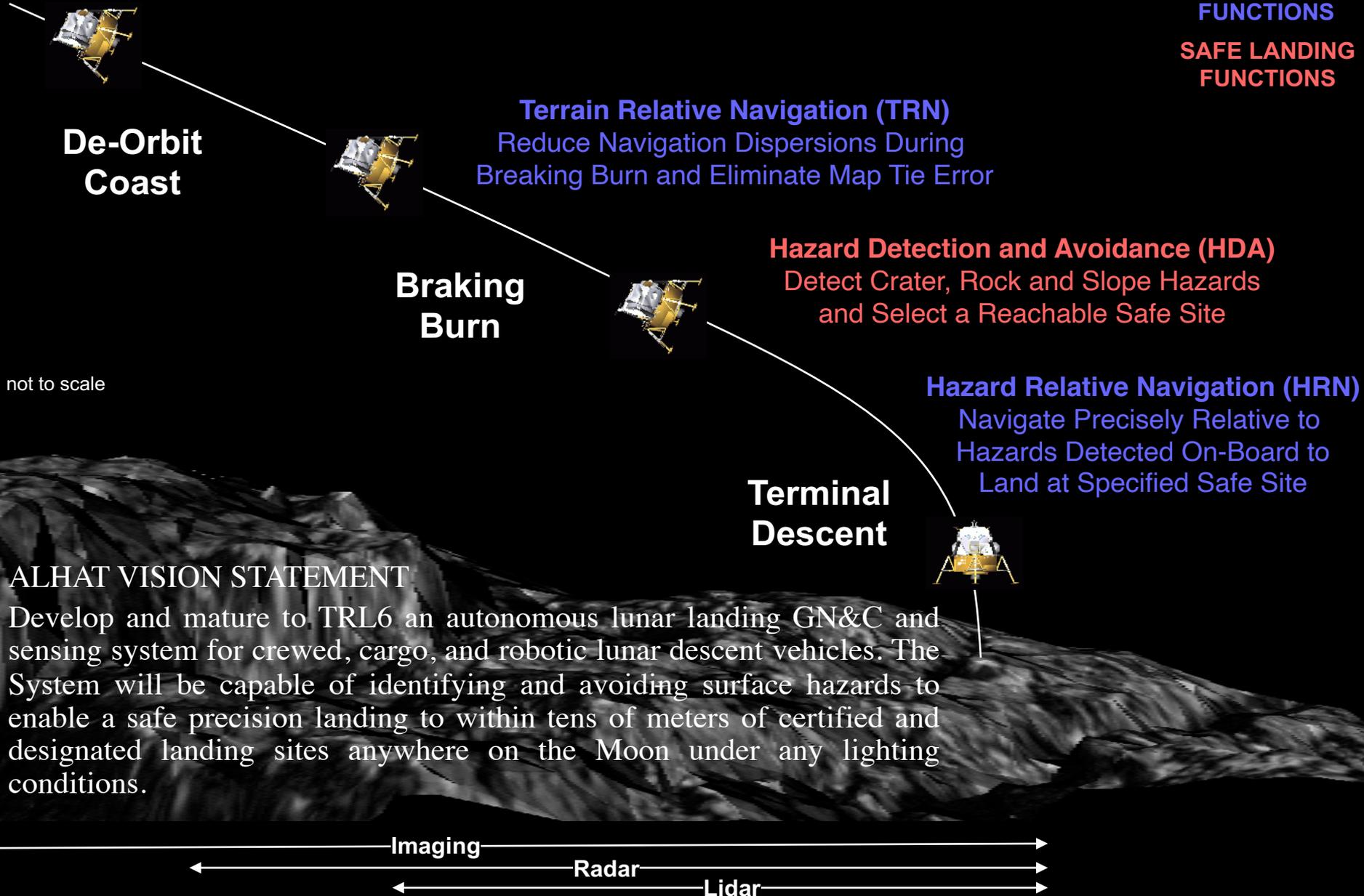
Backup Charts

Brief History of Space Computing at NASA/JPL

Need for high performance
computing in future NASA
space systems

ALHAT: Extreme Terrain Pinpoint Landing

PRECISION
LANDING
FUNCTIONS
SAFE LANDING
FUNCTIONS



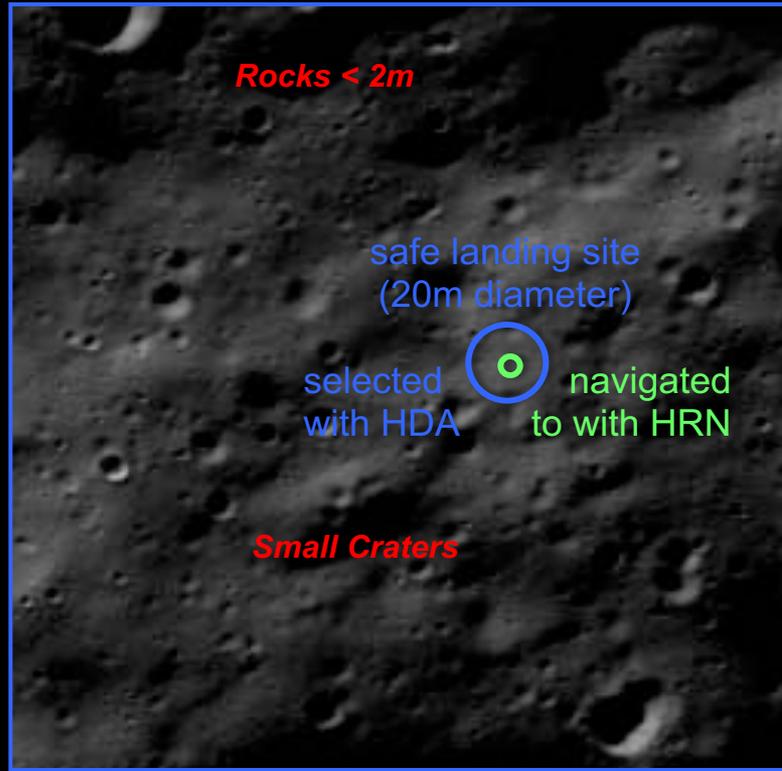
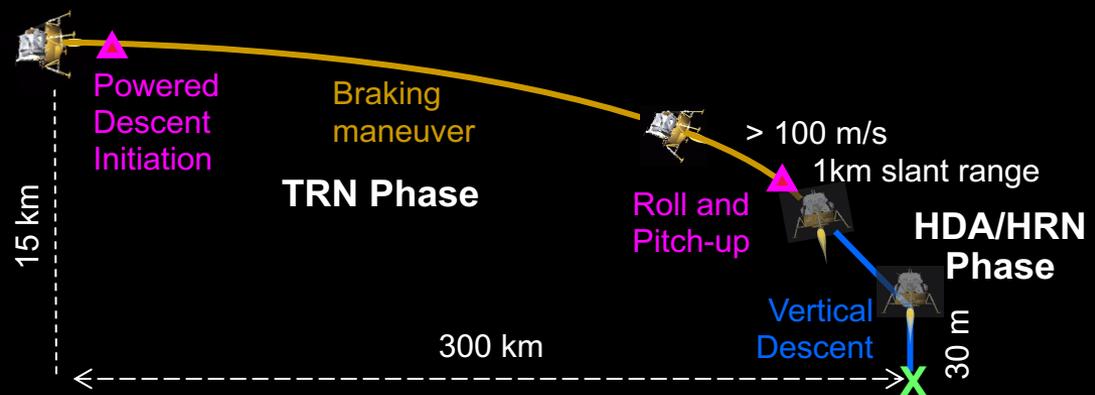
ALHAT

Extreme Terrain Pinpoint Landing

TRN = Terrain Relative Navigation

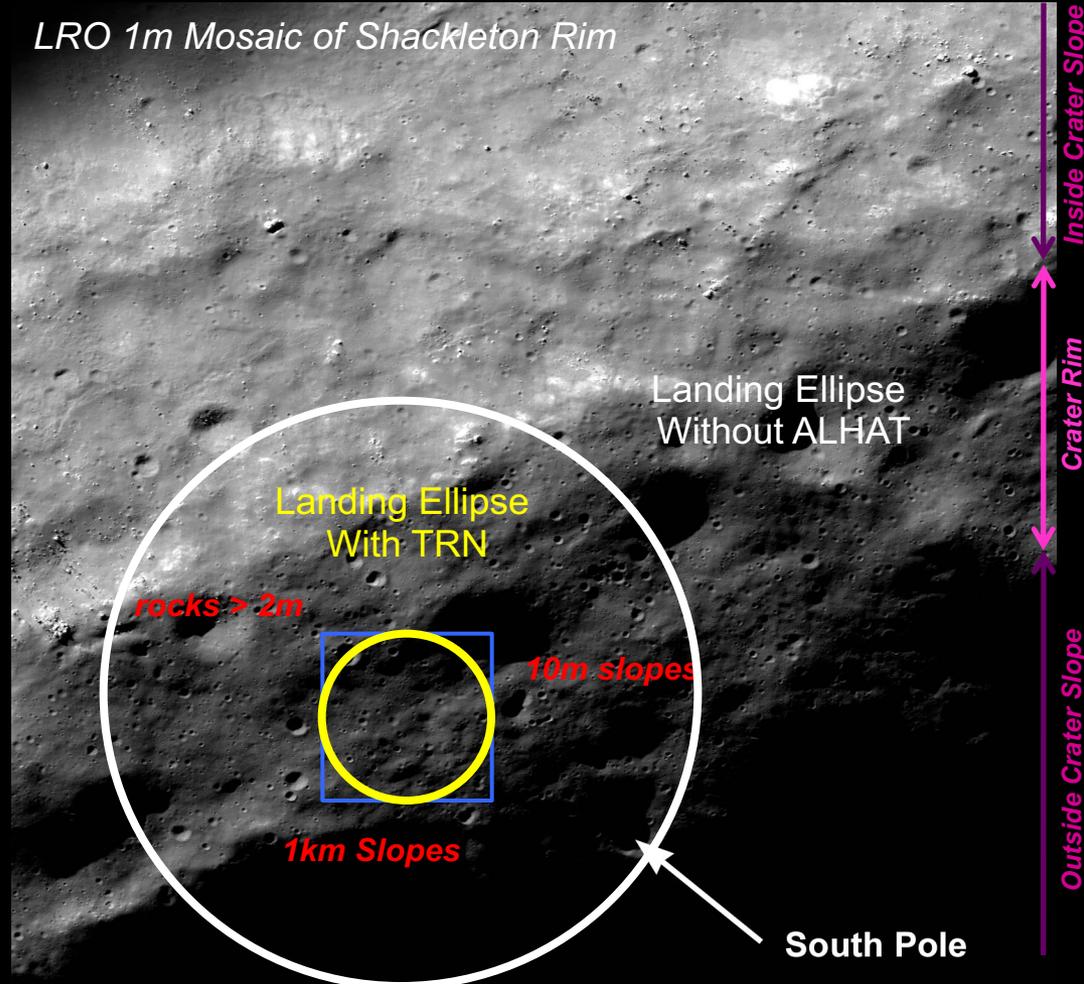
HDA = Hazard Detection and Avoidance

HRN = Hazard Relative Navigation



Hazard Map Area

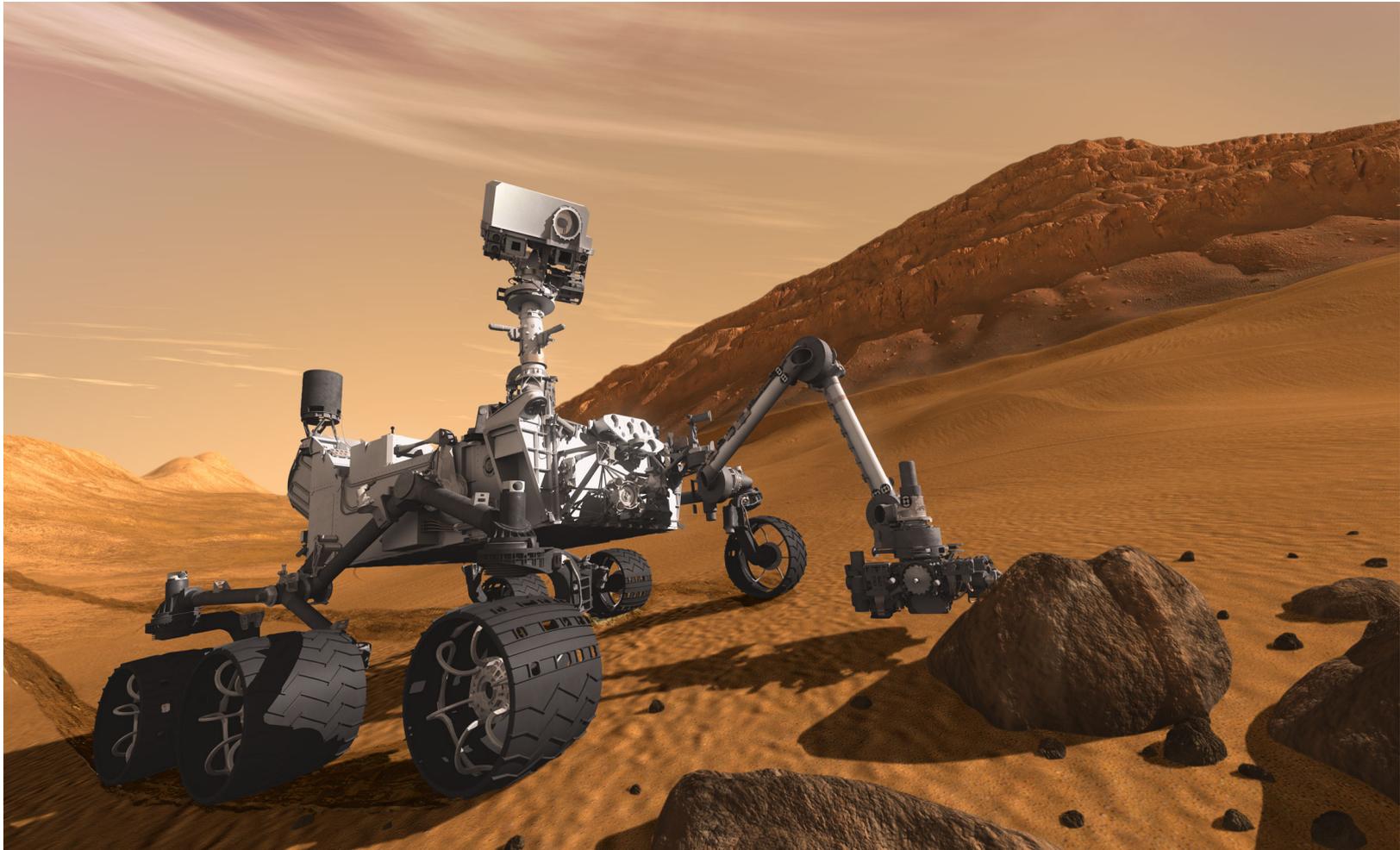
LRO 1m Mosaic of Shackleton Rim



Rover Fast Traverse and Autonomous Operations

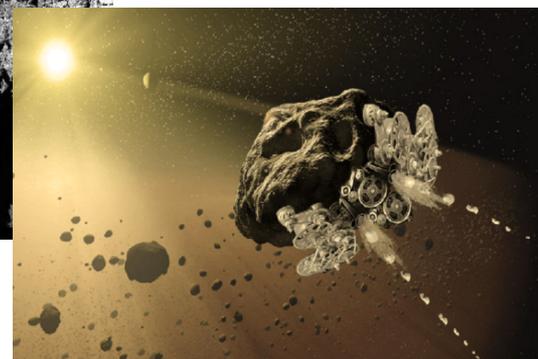
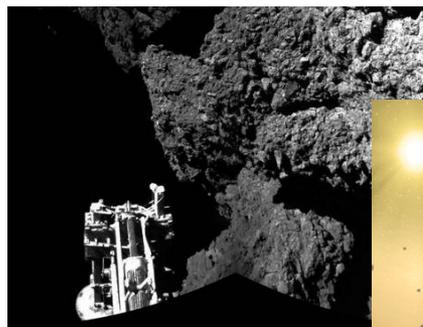
- Currently we can't walk and think at the same time
 - Look -> Think -> Walk -> Repeat
 - Stereoscopic photos, analyze terrain, plan path, turn wheels,
 - Extremely slow traverse
 - No significant science while driving (opportunistic science)
- What we'd like to do with high performance processing onboard
 - Real time terrain processing including soil/sand rock analysis
 - Real time situational awareness (internal and external)
 - Highly enervated (high density tactile, chemical sensors) "robotic skin" for engineering and science
 - High speed traverse over unknown terrain utilizing proprioception, wheel/chassis dynamics sensing, environmental sensing
 - Opportunistic science and autonomous science driven (re)planning
 - Coordination with orbiter and high resolution remote sense platform via overhead cloud computing resources
 - Coordination of multi-rover teams, swarms, helicopters

Mars 2020: Pin Point Landing, Fast Traverse, Autonomous Mission Planning and Execution.... Sort of

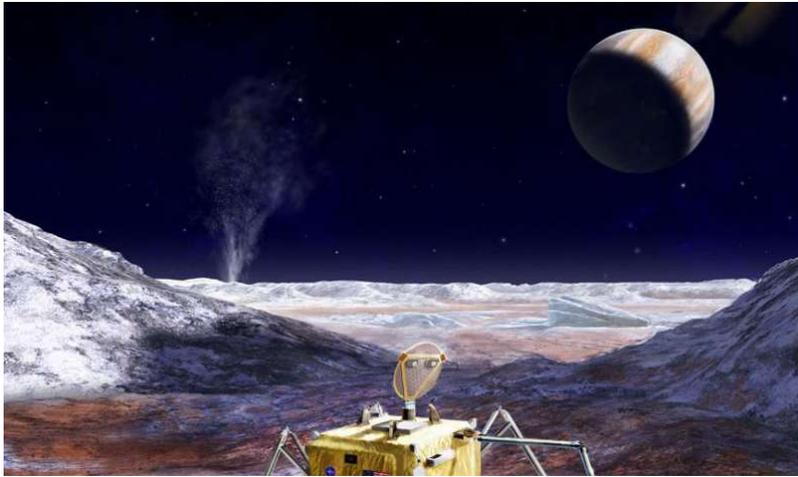


Close Proximity Operations

- Unknown and dynamically changing environment
- Science requires both remote sense and in situ/sampling operations
 - Gravity mapping
 - Terrain/topology mapping
 - Imaging and spectroscopy
 - Sample acquisition and analysis
- Multiple visits
 - Different sites
 - Follow up
- Multi-platform teams and swarms
 - Specialized platforms for different science and follow up visits
 - Coordination and collaboration
- Multiple-body extended missions
- Virtual (reality) visitation
 - Delay Tolerant Network Com
 - Downlink bandwidth minimization
 - Knowledge transmission (vs data)

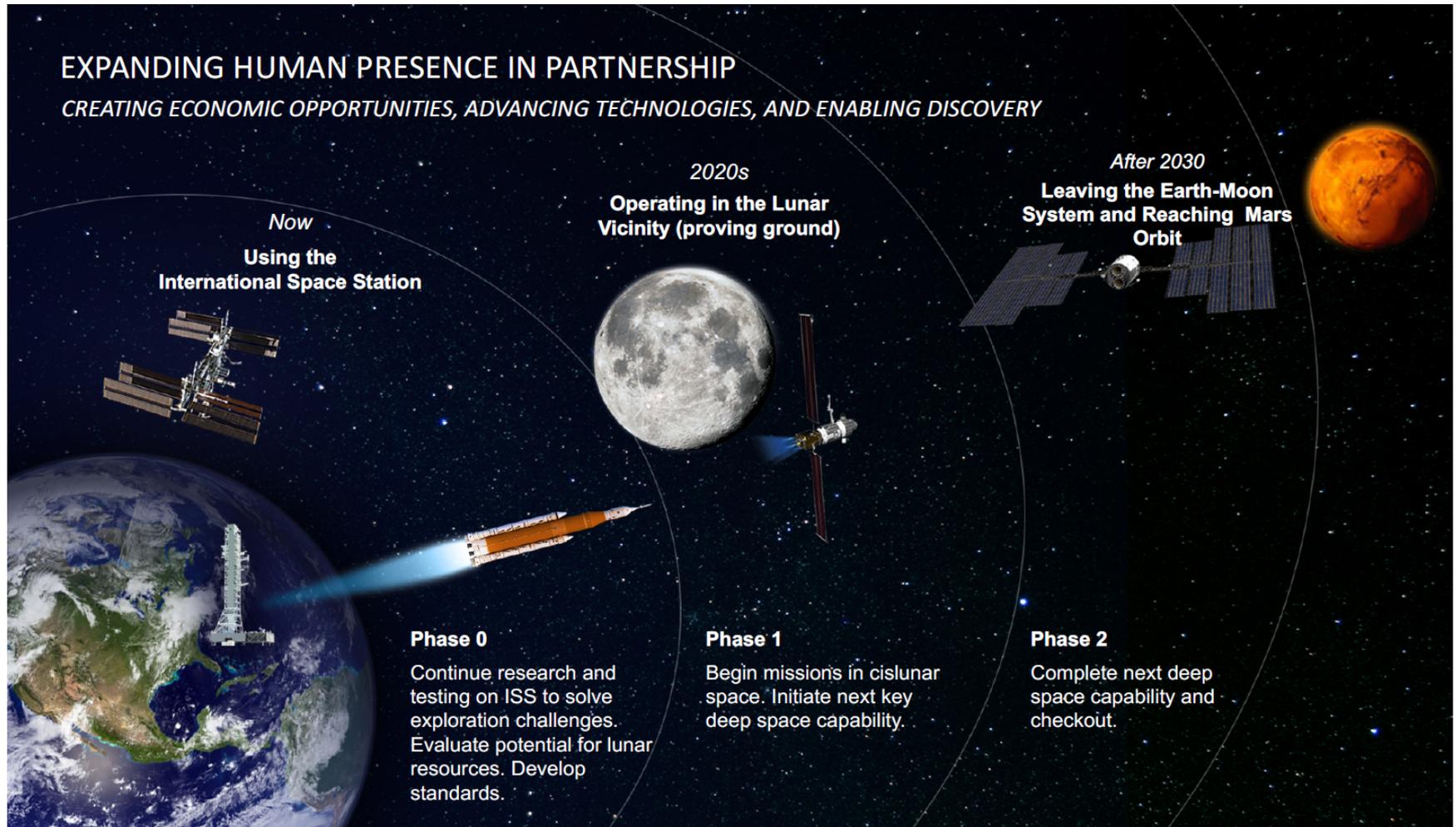


Ocean Worlds (e.g. Europa) Ocean Exploration

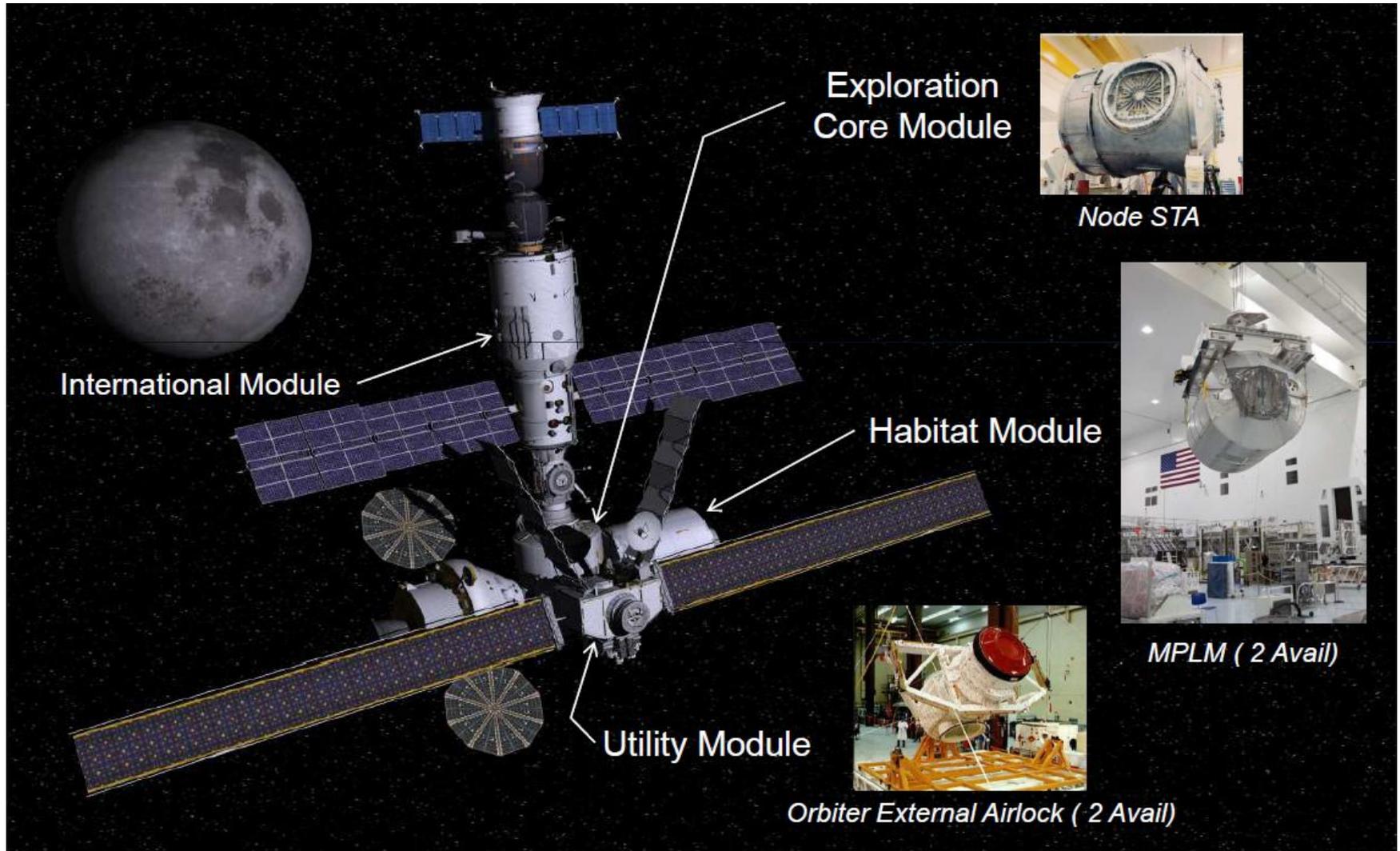


- Autonomous exploration and science
 - Minimal communication under Ice and Ocean
 - Occasional low data rate comm to surface and lander/orbiter via small, low power relays
- Autonomous under water navigation and mapping
 - SLAM (Simultaneous Location and Mapping)
 - Dead reckoning and relay assisted navigation
 - Chemical and thermal mapping and gradient/boundary detection
- Aqueous science
- Life detection
- Ice-bottom and ocean bottom science
- Multi-platform collaboration and coordination

A Long Term Vision for Crewed Missions

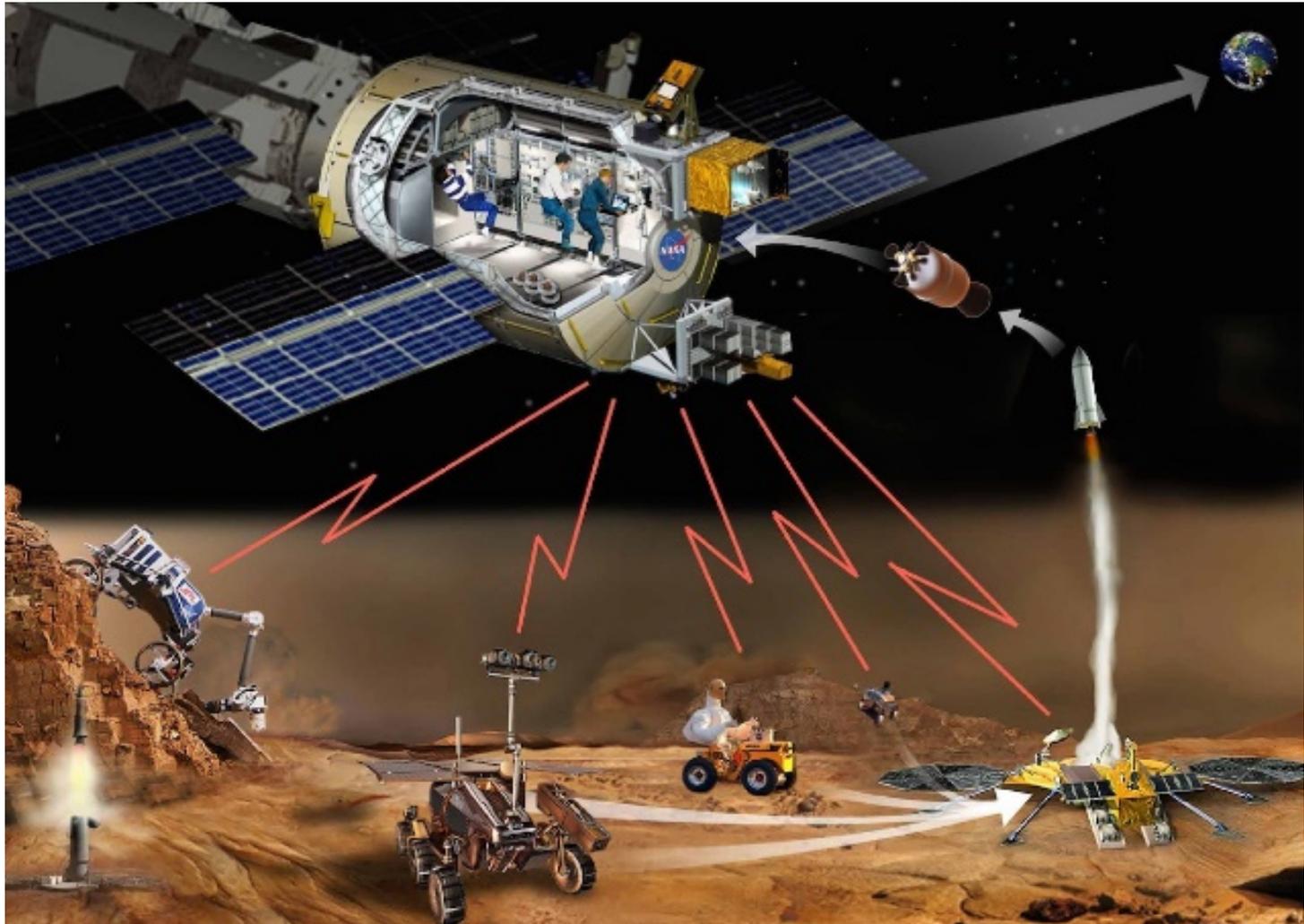


NASA Gateway: A Cis-Lunar Platform



A Vision for The Future of Exploration

Teleoperation with Virtual Reality Immersion Goal Level Direction of Robotic Teams
Autonomous Robotic Operations



Application Processing Requirements

Robotic Missions

Application	Criticality/ Fault Tolerance	Throughput (GOPS)	Real Time	Type of Processing	Memory Access	Data Rate
Autonomous Mission Planning	Mission/ Life critical	1 GOP+	Soft Real Time – seconds	<ul style="list-style-type: none"> • Database ops • Heuristic search • Parallelizable • Standard Math, Non-DSP, Floating Point preferred 	<ul style="list-style-type: none"> • Random • Memory Intensive 	Memory: 1Gb/S+ I/O: 100Mb/S+
Hyperspectral Imaging	Error detection	10s-100s GOPS+	Soft Real Time	<ul style="list-style-type: none"> • DSP • Parallelizable • FP Desirable • Matrix/ vector math 	<ul style="list-style-type: none"> • Random access • Repeated passes through a data set. 	Memory: Multi Gb/S
Radar – science	Error detection	10s-100s GOPS	Soft Real Time	<ul style="list-style-type: none"> • DSP • Data Base Ops • Parallelizable • Data Flow Amenable 	<ul style="list-style-type: none"> • Continuous access • Sequential access • Random access 	Memory: 1Gb/S+
Extreme Terrain Landing	Mission critical	25-50 GOPS	Hard Real Time - 1 second	<ul style="list-style-type: none"> • DSP • Control Code • Parallelizable • FP Desirable 	<ul style="list-style-type: none"> • Continuous access • Sequential access • Random access 	Memory: 30- 50MB/S
Disaster Response Constellation	Not safety critical, and some data loss is permissible	1-10 GOPS	Soft Real Time	<ul style="list-style-type: none"> • DSP • Onboard spectral signature matching • Real-time High-Def video compression and data handling • Standard Earth land/ocean data product generation pipeline, migrated to space platforms • Change detection (various strategies) 	<ul style="list-style-type: none"> • Regular Sequential access • Possible multi-instrument data fusion. • Random access 	Memory: 200MB/S- 1Gb/S

Application Processing Requirements

Crewed Missions

Application	Criticality/ Fault Tolerance	Throughput (GOPS)	Real Time	Type of Processing	Memory Access	Data Rate
Advanced Vehicle Health Management	Failure could lead to Mission degradation Shutdown due to fault should recover automatically	18 GIPS/27 GOPS and 18 GFLOPS currently	Multiple second response times are acceptable	<ul style="list-style-type: none"> Model-Based Reasoning Techniques High Rate Instrument Data Processing Knowledge Retrieval / Synthesis 	<ul style="list-style-type: none"> Continuous interleaved and irregular or random reads and writes 	Memory: 10 – 50 Gb/S I/O: 1 – 10 Gb/S
Crew Knowledge Augmentation System (Watson in Space)	Failure could lead to Mission degradation Shutdown due to fault should recover automatically Live lock-ups and manual restarts acceptable	18 GIPS/27 GOPS and 18 GFLOPS currently	Soft Real Time	<ul style="list-style-type: none"> Model-Based Reasoning Techniques High Rate Instrument Data Processing Knowledge Retrieval / Synthesis 	<ul style="list-style-type: none"> Continuous interleaved and irregular or random reads and writes 	Memory: 10 – 50 Gb/S I/O: 1 – 10 Gb/S
Augmented Reality	Failure could lead to Mission degradation Shutdown due to fault should recover automatically	24 GIPS/36 GOPS and 24 GFLOPS	Timing requirement is < 1 sec	<ul style="list-style-type: none"> Vision-Based Algorithms. High Rate Instrument Data Processing. Knowledge Retrieval / Synthesis. 	<ul style="list-style-type: none"> Continuous interleaved and irregular or random reads and writes 	Memory: 100+ Gb/S I/O: 50+ Gb/S
Telepresence	Failure could lead to Mission degradation Shutdown due to fault should recover automatically Live lock-ups and manual restarts acceptable	24 GIPS/36 GOPS and 24 GFLOPS currently	Timing requirement is < 1 sec	<ul style="list-style-type: none"> Vision-Based Algorithms. High Rate Instrument Data Processing. 	<ul style="list-style-type: none"> Continuous interleaved and irregular or random reads and writes 	Memory: 100+ Gb/S I/O: 50+ Gb/S

Application Processing Requirements Summary

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

A Perspective on High Performance Computing and Autonomy

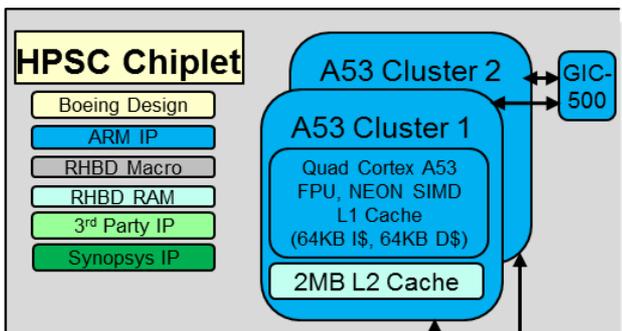
What will be required to
achieve the desired
capabilities

Brief Review of Current Space HPC offerings

Emerging Concepts: Chiplet and System in Package

Chiplet Overview

Chiplet Architecture: High-performance Cores



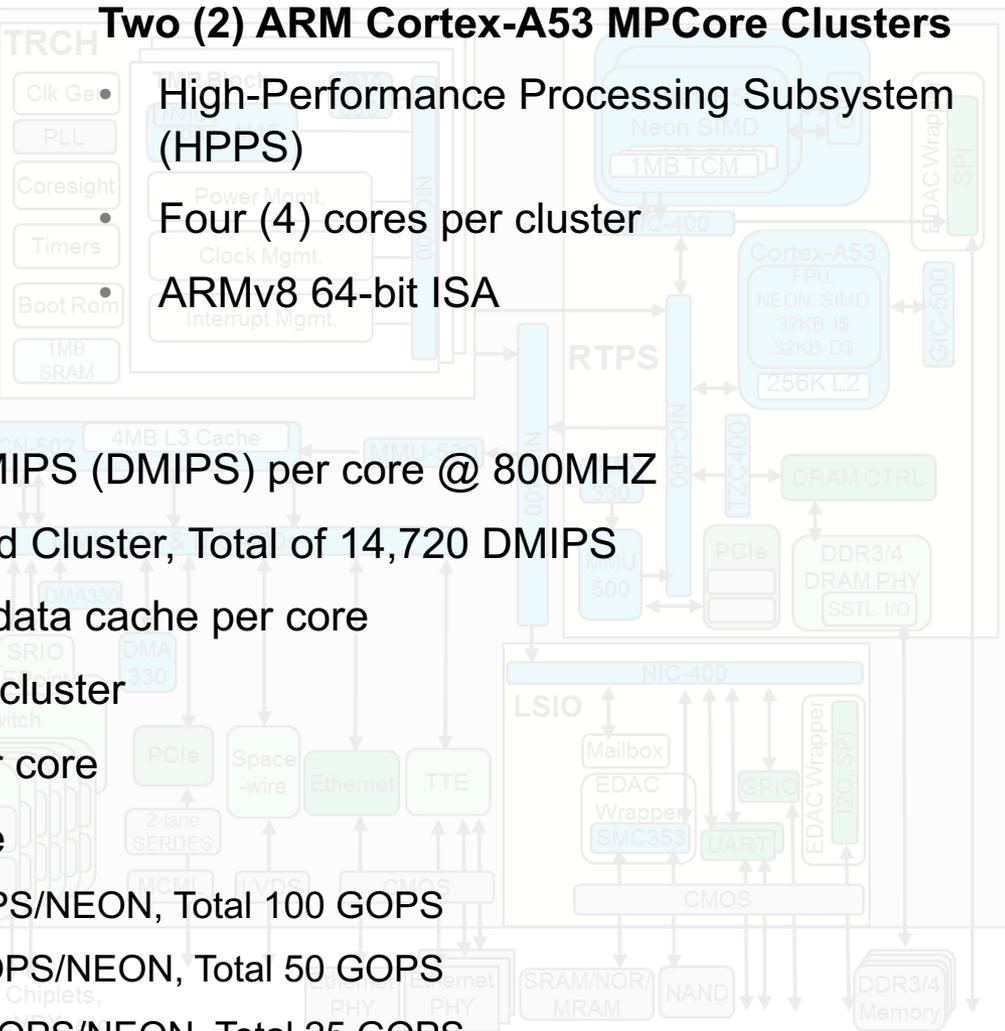
Two (2) ARM Cortex-A53 MPCore Clusters

High-Performance Processing Subsystem (HPPS)

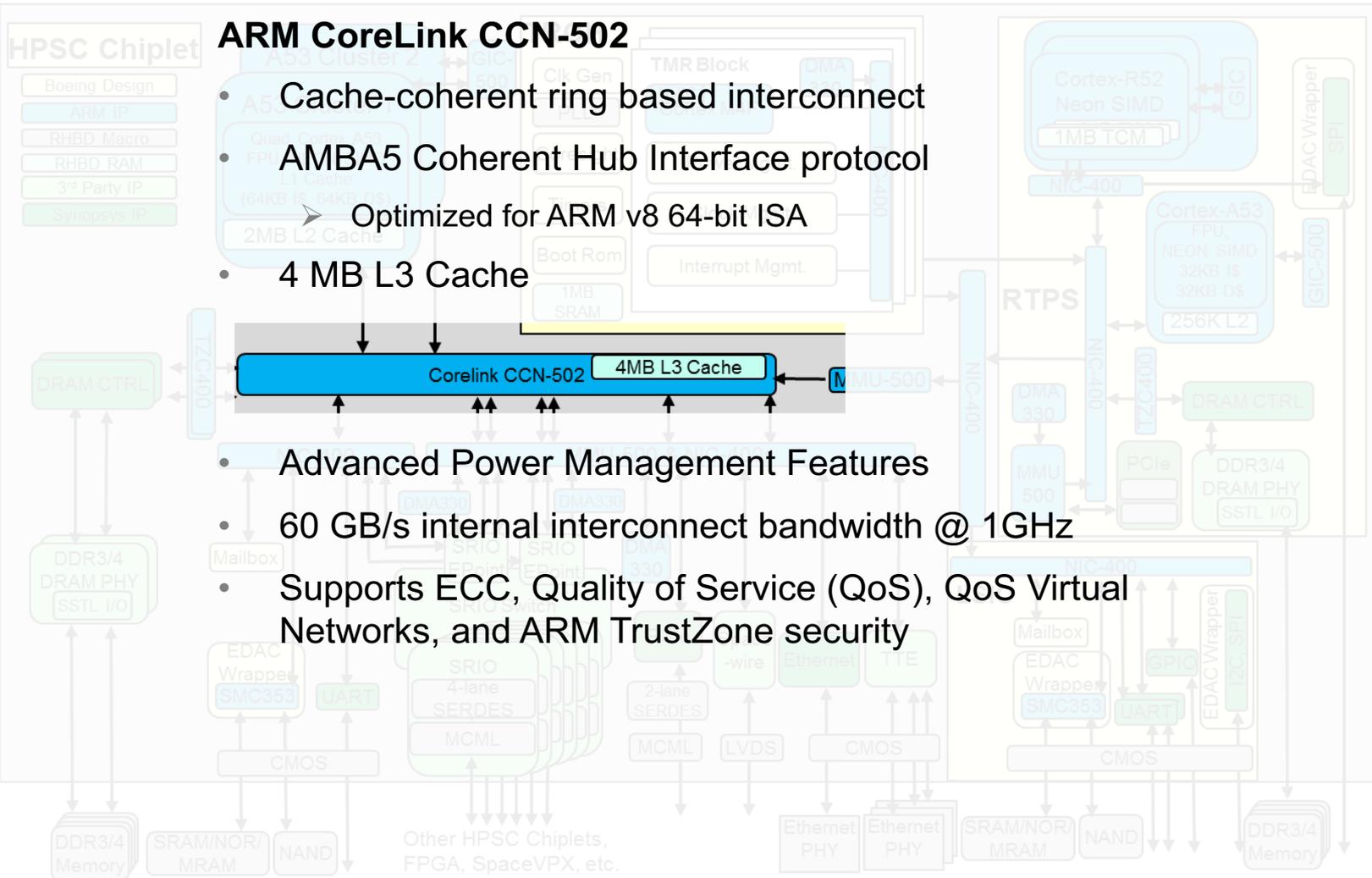
Four (4) cores per cluster

ARMv8 64-bit ISA

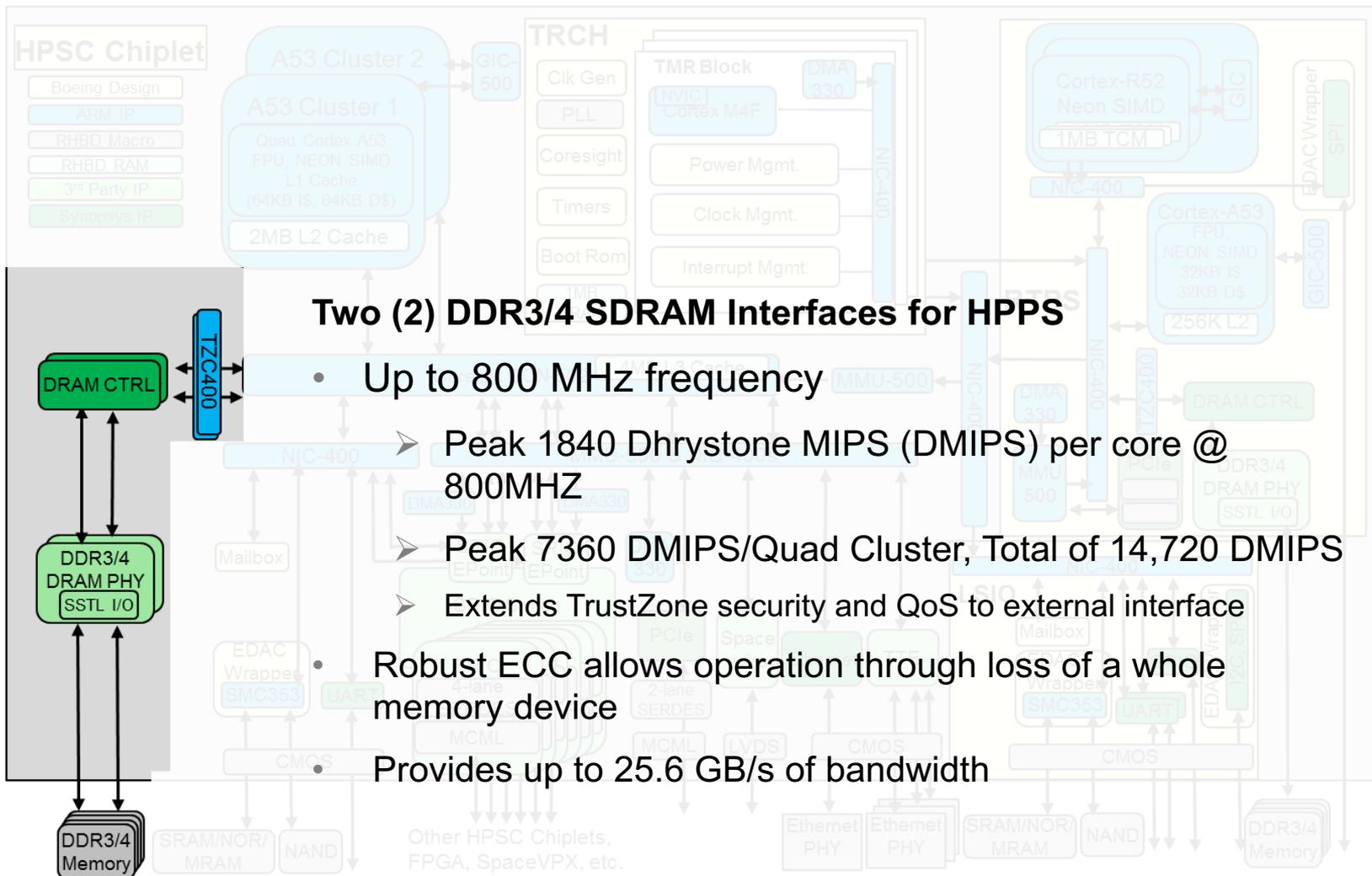
- Up to 800 MHz frequency
 - Peak 1840 Dhrystone MIPS (DMIPS) per core @ 800MHZ
 - Peak 7360 DMIPS/Quad Cluster, Total of 14,720 DMIPS
- 64 KByte L1 instruction and data cache per core
- 2 MByte L2 Cache per quad cluster
- Floating Point Unit (FPU) per core
- NEON SIMD engine per core
 - 8-bit mode: Peak 12.5 GOPS/NEON, Total 100 GOPS
 - 16-bit mode: Peak 6.25 GOPS/NEON, Total 50 GOPS
 - 32-bit mode: Peak 3.125 GOPS/NEON, Total 25 GOPS



Chiplet Architecture: Interconnect



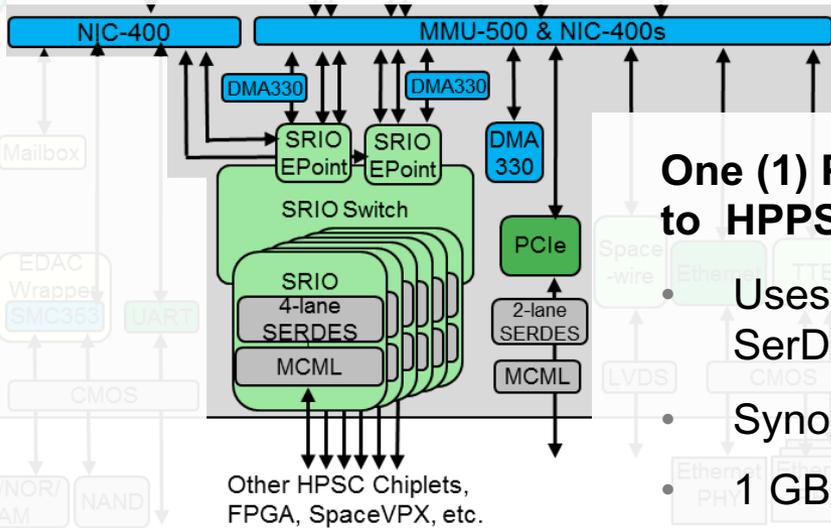
Chiplet Architecture: DRAM Interfaces



Chiplet Architecture: Serial I/O Interfaces

Six (6) Serial RapidIO® (SRIO) 3.1 ports

- Four (4) lanes per port, each lane up to 10.3125 GBd full-duplex
- Uses Boeing's 32nm RHBD SerDes macro
- Supports high-speed serial connections to:
 - Other Chiplet devices
 - Expansion FPGA
 - Protocol conversion
- Provides up to 59.1 GB/s of Serial I/O bandwidth



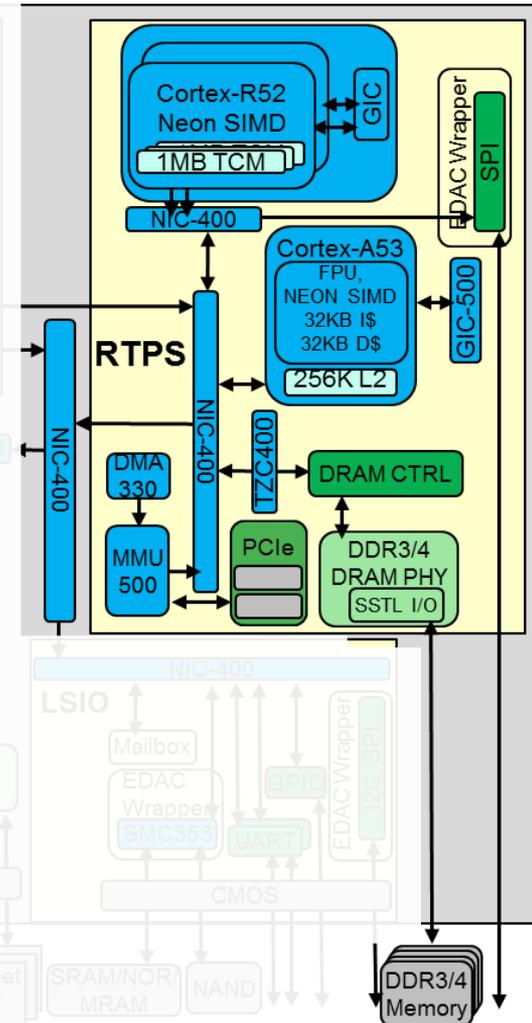
One (1) PCIe Gen2 port dedicated to HPPS

- Uses Boeing's 32nm RHBD SerDes macro (2 lanes)
- Synopsys PCIe Controller
- 1 GB/s Bandwidth

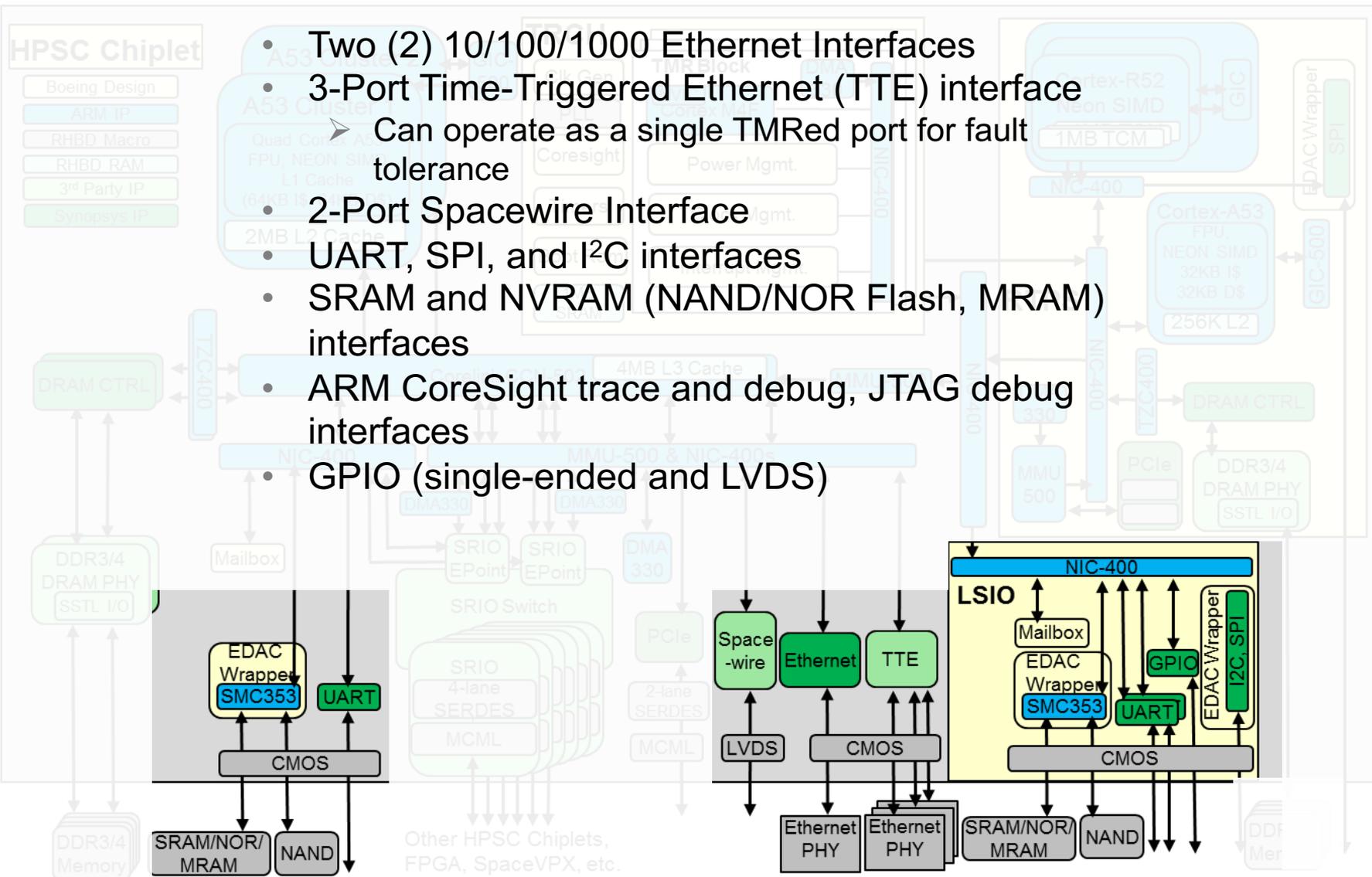
Chiplet Architecture: Realtime Processing Subsystem

Realtime Processing Subsystem (RTPS)

- Single Cortex-A53 core managing two (2) Cortex-R52 Realtime cores (ARM v8 64b)
 - Supports virtualization and time & space partitioning / ARINC 653, as well as realtime performance needs
- RTPS Dedicated Memory & IO interfaces:
 - One (1) DDR3/4 interface
 - One (1) PCIe Gen2 interface
 - One (1) SPI interface
- R52 cores provide:
 - ARM's highest level of safety features, including Dual-Core Lock Step (DCLS) operation
 - Up to 600 MHz frequency
 - Peak 1296 Dhrystone MIPS (DMIPS) per core @ 600MHZ
 - Floating Point Unit (FPU), NEON SIMD engine, and 1 MB Tightly Coupled Memory per core
- A53 core provides:
 - Peak 1380 Dhrystone MIPS (DMIPS) @ 800MHZ
 - 32 KB L1, 256 KB L2 Caches

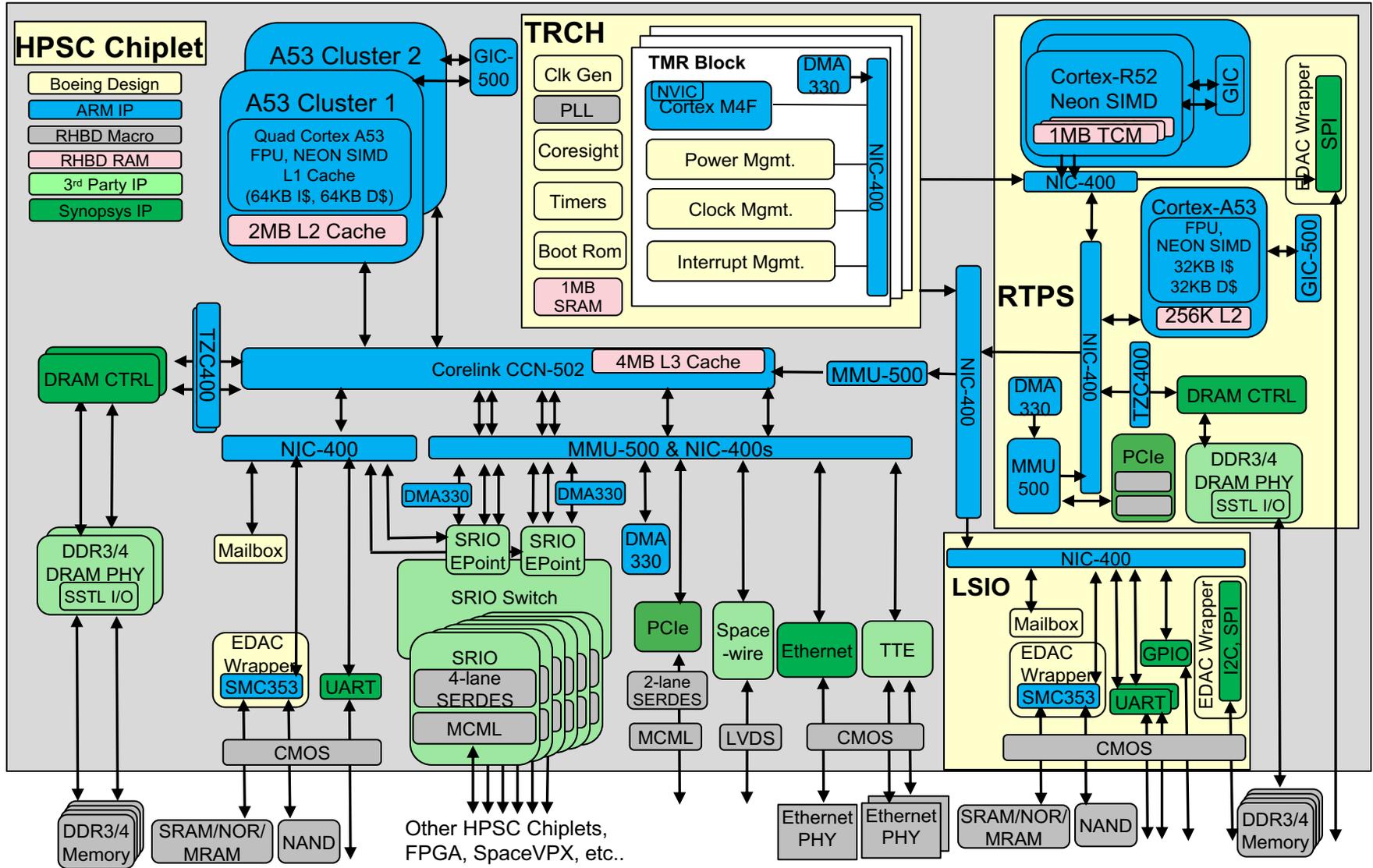


Chiplet Architecture: Other IO Interfaces

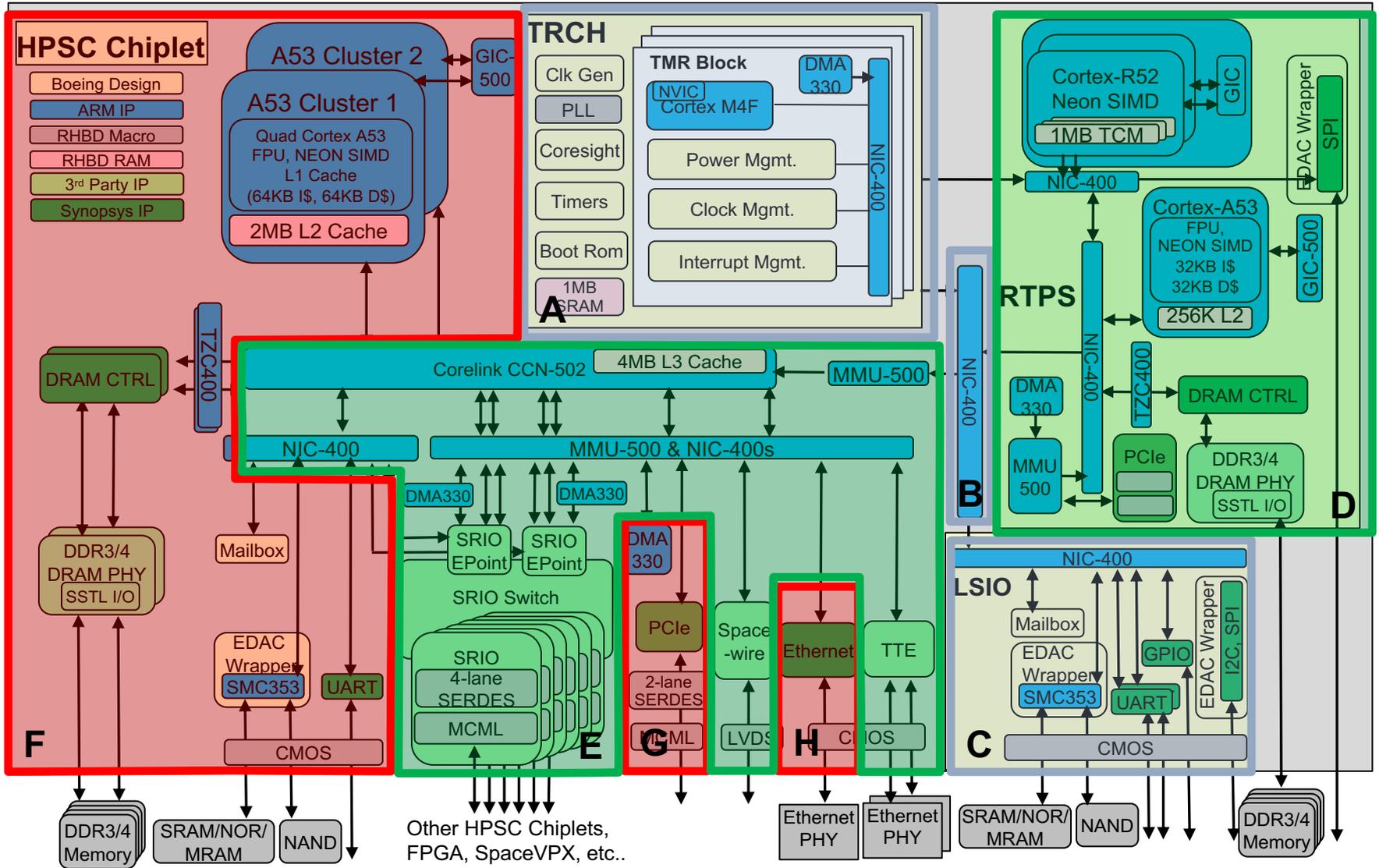


Specifying Power, Performance, Reliability & Radiation Tolerance in Multicore

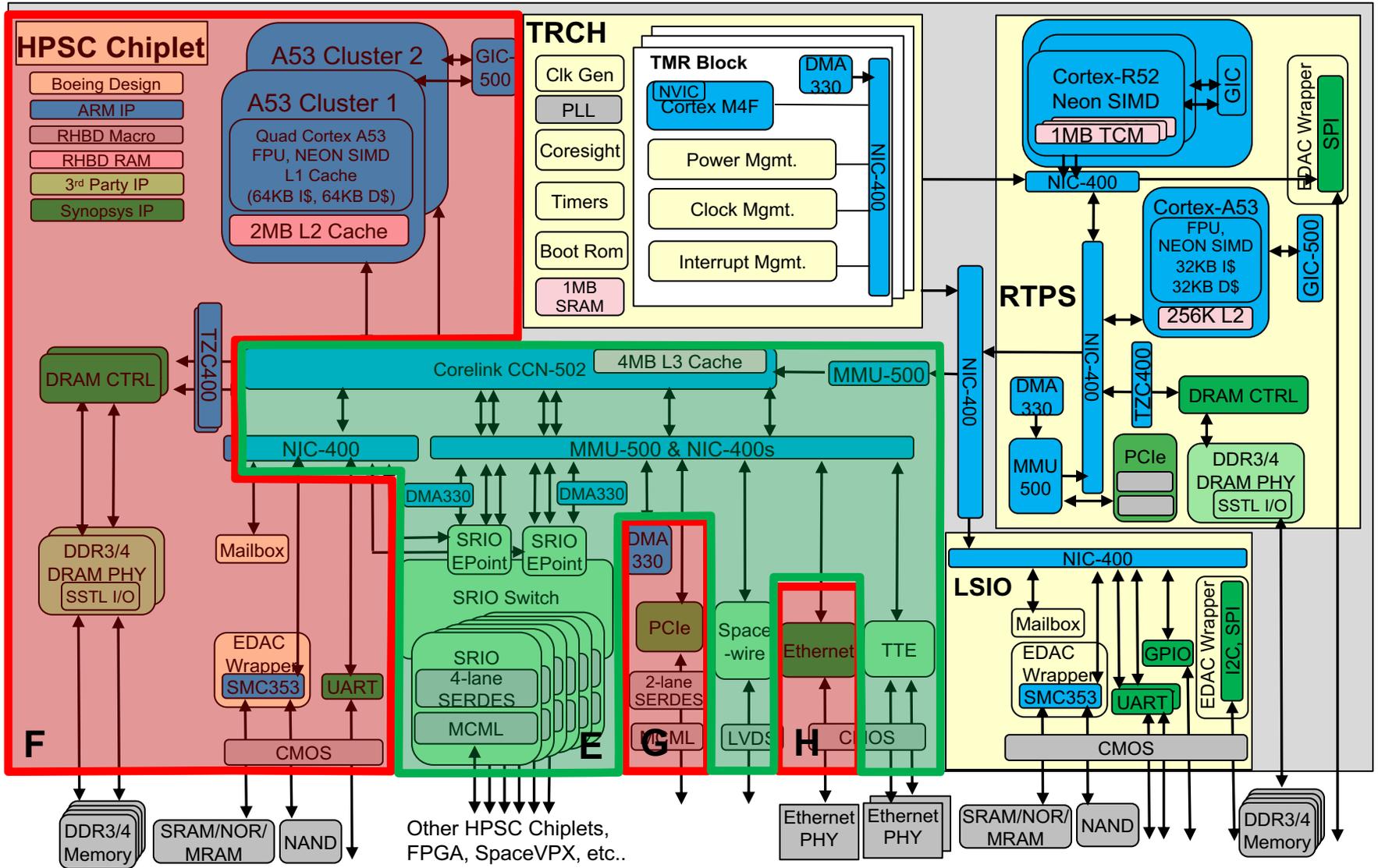
HPSC Chiptlet: Top Level



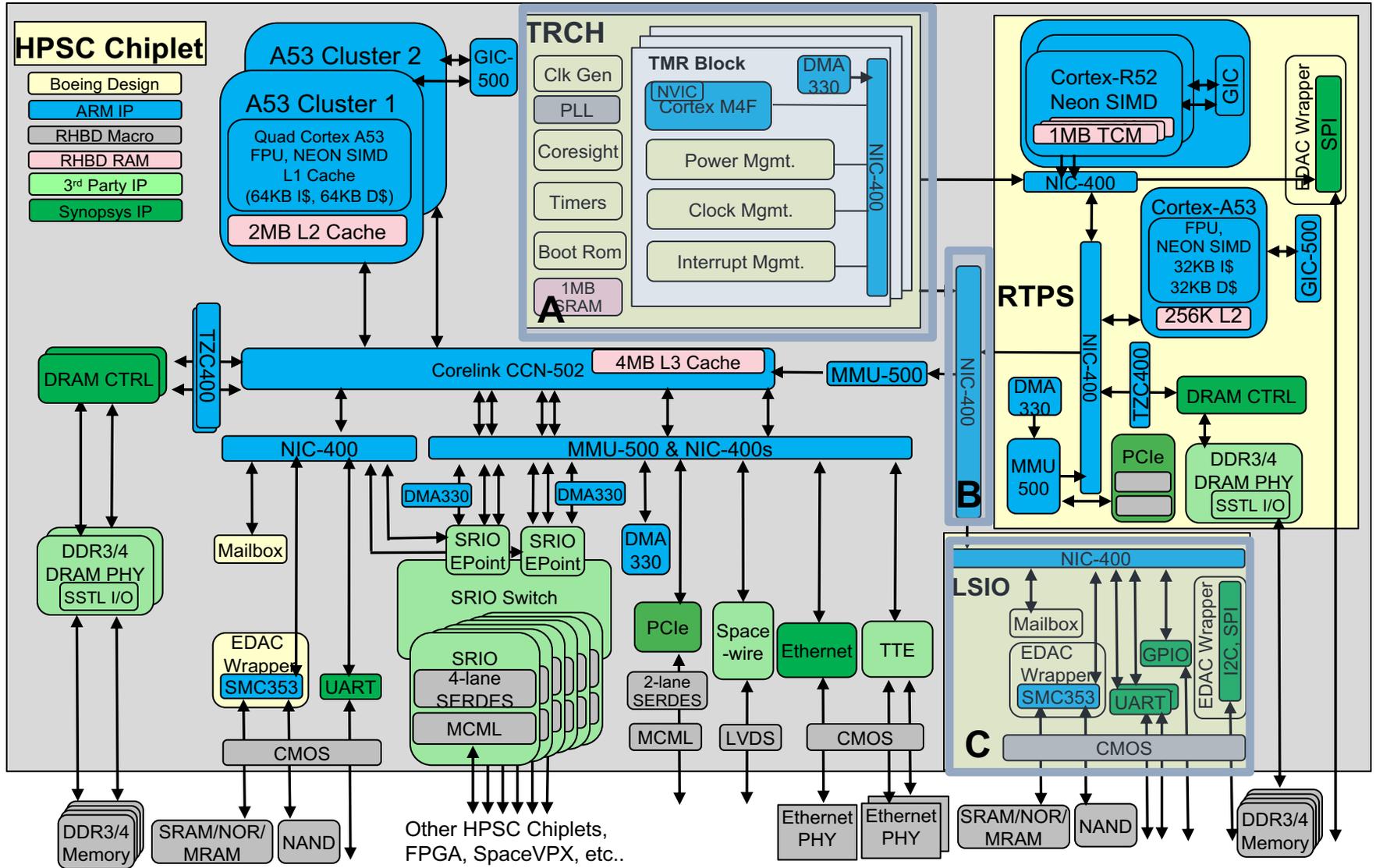
SEE Regions



1x SEE Analysis Scenario



100x SEE Analysis Scenario



HPSC Ecosystem & Forward Planning

Use Cases

Chiplet System Software

The HPSC System Software

- The HPSC Chiplet inherits a large complement of existing open source software including:
 - Libraries, operating systems, compilers, and debuggers.
- We're able to leverage much of this software unmodified.
- The HPSC System Software effort largely encompasses 4 thrusts:
 1. Board support packages for Linux and RTOS;
 2. Development tools (e.g., compilers, debuggers, IDEs);
 3. Software-based fault tolerance; and
 4. Chiplet emulators.
- Our goal is to build a sustainable software ecosystem to enable full lifecycle software development.

The HPSC System Software

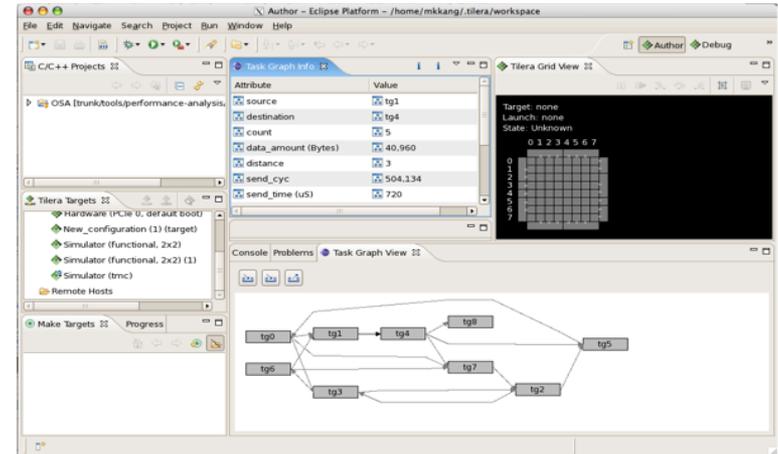
- The HPSC Chiptlet inherits a large complement of existing open source software including:
 - Libraries, operating systems, compilers, and debuggers.
 - Much of this software can be leveraged unmodified.
- The HPSC System Software effort largely encompasses 4 thrusts:
 1. Board support packages for Linux and RTOS;
 2. Development tools (e.g., compilers, debuggers, IDEs);
 3. Software-based fault tolerance; and
 4. Chiptlet emulators.
- The goal is to build a sustainable software ecosystem to enable full lifecycle software development.

Linux and RTOS BSPs

- SMP Linux:
 - A Yocto/OpenEmbedded-based solution is anticipated
 - The Yocto project is a Linux Foundation project targeting the embedded Linux community.
 - It's a software distribution plus BSP specification for embedded architectures.
 - Includes Intel, Xilinx, Wind River, among contributing members
 - Linux already supports Cortex-A53 with limited support for MMU-less architectures (e.g., Cortex-R).
- RTEMS RTOS:
 - Port to Cortex-A53.
- Also looking at other solutions, e.g. VxWorks, Integrity.

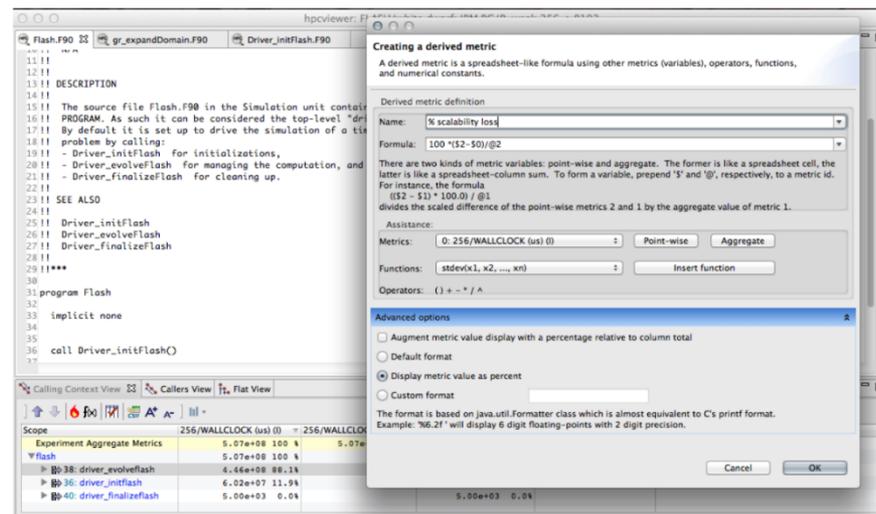
Development Tools

- IDE-based on Eclipse with standard developer tools including:
 - GCC and LLVM compilers
 - GDB debugger
 - Open source performance analysis tools
- Will also support ARM DS-5 IDE
 - Allows use of ARM Compiler 6
 - Helpful for board bring-up
 - Includes ARM performance analysis tools



Parallel Programming and Performance Analysis

- The HPSC Chiptlet natively supports a full complement of programming languages and parallel programming models:
 - Including C, C++, Fortran, Java, etc..
 - MPI, OpenMP, Pthreads
 - We inherit these languages and parallelism models for free.
- The considerable open source HPC performance analysis ecosystem will be leveraged:
 - HPCToolkit, Tau, gprof, PAPI
- The ARM Streamline Performance analyzer will be supported.



Example HPCToolkit derived metric dialog.

HPSC Middleware Concept

- HPSC Middleware is a software layer that provides services to the higher-level application software to achieve:
 - Configuration management
 - Resource allocation
 - Power/performance management
 - Fault tolerance management
- Serving as a bridge between the upper application layer and lower operating system or hypervisor, the middleware will significantly reduce the complexity of developing applications for the HPSC Chiplet

INTEGRATED STACK CONCEPT

Mission Applications

Core Flight Software /Spacecraft Control

**HPSC Middleware – Resource Management
Mission-Friendly Interface for
Managing/Allocating Cores for
Performance vs. Power vs. Fault Tolerance**

Traditional System Software – RTOS or
Hypervisor

Hardware – Multi-core Processor Chips,
Evaluation Boards

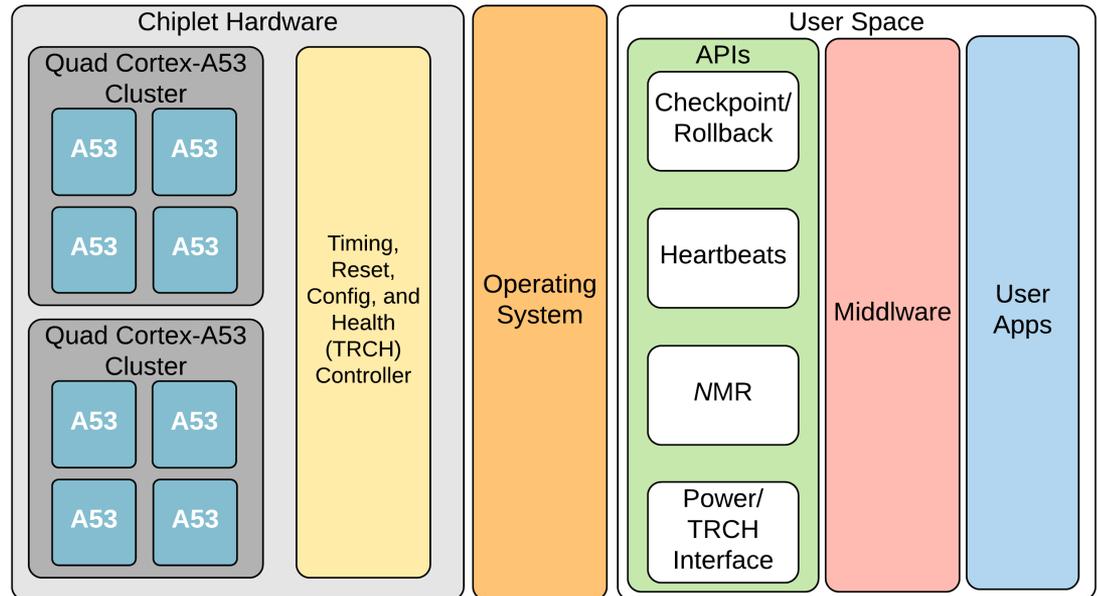
Middleware Access to Hardware Configuration and Fault Tolerance

- The fault tolerance infrastructure is tunable to anticipated fault conditions:

- Redundancy level
- Checkpoint interval
- Heartbeat interval and failure response

- Chiplet hardware:

- Clock frequency
- Clock gating and sleep modes
- Number of active cores
- Interrupts, Memory Map
- Protocol Checkers
- Memory ECC and scrubbing
- Low level fault detection/mitigation
- Processor access to hardware resources
- BIST
- Multi-Chiplet Operations (boot/initialization, resource allocation, fault tolerance)



A well-defined API will enable application and middleware control of the Chiplet's fault tolerance and power features.

The HPSC Emulator

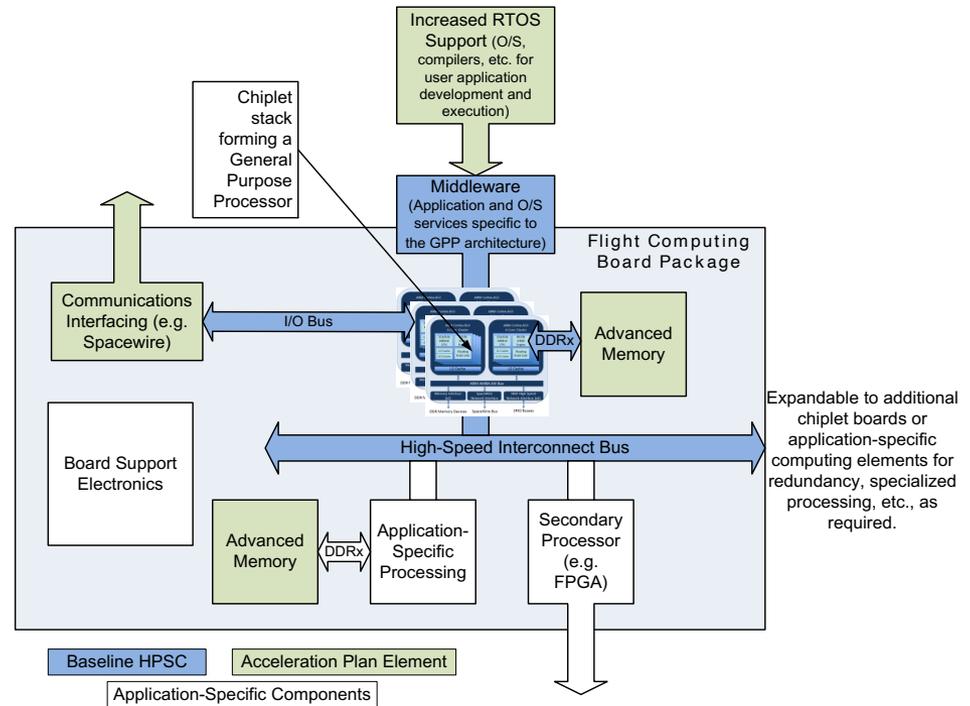
- Two emulation approaches are being developed for the HPSC Chiplet to address the multiple community needs:
 - QEMU-based emulator for application, operating system, and middleware development
 - FPGA-based emulator for timing-critical code and middleware, fault analysis, and system software validation
- The QEMU emulator is open source and will be available to developers
 - Initial version expected June 2018
 - Final version expected January 2019

HPSC Ecosystem and Roadmap

HPSC Avionics Ecosystem

Avionics Ecosystem

The overall High Performance Spaceflight Computing (HPSC) architecture is an “ecosystem” formed by the processor and supporting hardware and software elements to make a modern, scalable, rad-hard flight computing environment for sensors/instruments and mission functions.



Example of *HPSC* in a “full up” computing configuration where the *HPSC* “Chiplet” is the General Purpose Processor (GPP) that utilizes memory, specialized co-processing, and data communications interfaces.

HPSC

Ecosystem Roadmap

- The HPSC Roadmap partitions the HPSC “ecology” into the following categories:
 - Advanced Rad Hard Space Memory – high capacity/speed/reliability space qualified, volatile and non-volatile memory components for processor, instrument and mass storage devices
 - Co-processors/accelerators and “micro-Chiplet” – custom processors that provide extremely high performance at low power for specialized types of computation including: digital signal processing (DSP), graphics (GPU) and machine vision (MV) processing, deep learning/neuromorphic processing, as well as a microprocessor version of the Chiplet for extremely low cost, instrument/subsystem-embedded applications
 - System Software and Execution Environment –next gen, parallel, secure operating systems, hypervisors and middleware
 - Software Development Environment – compilers, debuggers and V&V tools for high reliability, real time, parallel codes
 - Power Supply and Smart Power Bus – efficiently provides requisite voltages, and intelligent power distribution system required for complex digital systems
 - Initial prototype single board computer and non volatile memory modules

HPSC Memory

Capability Description*

Product	Capability Description
Volatile Chip (VC)	300krad, DDR3/4, 1Gb
Non-Volatile Chip (NVC)	300krad, DDR3/4, 4Gb
Volatile Stack (VS)	300krad, Serial I/O, 8+Gb
Non-Volatile Stack (NVS)	300krad, Serial I/O, 32+Gb

High Density Memory Package



Mission Applications

Most NASA space mission would benefit from application of these memories:

- HPSC-based s/c - improved radiation hardness at COTS-like performance for working memory
- Non-HPSC-based s/c - improved performance and radiation hardness for generic processors and FPGA-based applications
- Instrument processors – improved performance at significantly reduced SWAP-C
- Missions requiring large mass store capabilities – high capacity, high speed, non-volatile memory

Capability Status*

Product	Developer	TRL Level	Current Status
VC	Boeing/Tezzeron	3	IRAD-TBD
NVC	Cobham-Aeroflex	3	IRAD-TBD
VS	Micron	3	COTS non-rad hard
NVS	Honeywell-Tezzeron	3	SBIR - TBD

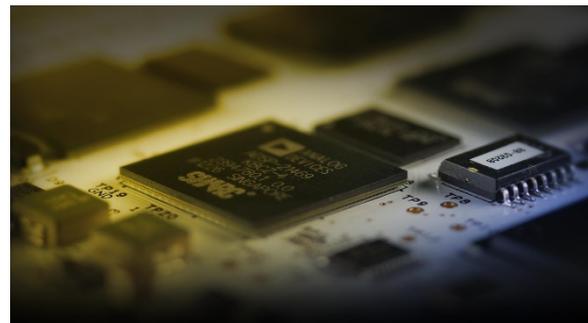
*Ongoing Advanced Memory Study will refine estimates and recommend implementation plan

HPSC

Co-Processors/Accelerators

Capability Description*

Product	Capability Description
DSP	Digital Signal Processor extension to HPSC
GPU	Graphics Processor extension to HPSC
Vision**	Machine Vision Processor for HPSC
Neuromorphic	Deep learning/recognition engine for HPSC
U-Chiplet	Single A53 Core embedded processor
FPGA*	Next Gen High Performance Rad Hard



GPU

Mission Applications

- DSP- RADAR, HSI, LIDAR other digital signal processing intensive sensor systems
- GPU – HEOMD crewed missions
- Vision – robotic autonomous exploration and science missions
- Neuromorphic – robotic autonomous exploration and science missions
- U-Chiplet (micro-Chiplet) – virtually all missions either as main processor for SmallSats or as subsystem/instrument-embedded processor
- FPGA – virtually all missions – multiple uses

Capability Status*

Product	Developer	TRL Level	Current Status
DSP	TI	COTS	Non-Rad Hard
GPU	ARM	COTS	Non Rad Hard
Vision**	ARM	3	Non-Rad Hard
Neuro	TBD	3	TBD
U-Chiplet	ARM	COTS	Non-Rad Hard
FPGA*	TBD	COTS	Non-Rad Hard

Approach, in all cases, is to leverage existing IP, rad harden, and provide AMBA bridge to Chiplet

*Being developed under Title III Program, AFRL lead, no cost to NASA

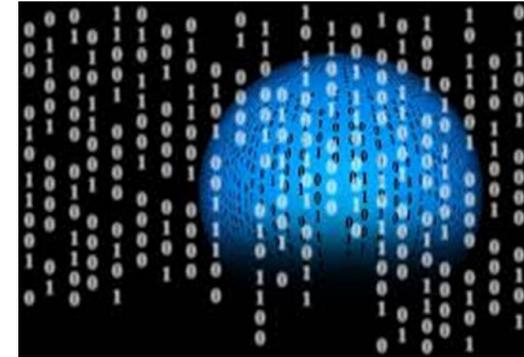
**May be combined with GPU for overall cost reduction

HPSC

System Software

Capability Description*

Product	Capability Description
PRTOS	Qualified space RTOS suitable for NASA missions
CFE/CFS	Common flight software package for standard s/c avionics
Configuration Middleware (CM)	Provides management of HPSC resources, fault tolerance, and power
Parallel Proc'g Middleware (PM)	MPI, and other parallel processing packages
Math Libs (ML)	Parallel as well serial math packages



Advanced Parallel Processing Software

Mission Applications

Virtually all s/c will require an RTOS and CFE/CFS. Most HPSC based missions will also require configuration middleware to simplify HPSC management. Many missions will also need significant parallel processing for onboard science data processing, autonomy processing and robotics processing – these will require parallel processing middleware and math libraries.

Note: The two primary flight software packages are CFE and CFS from GSFC and JPL respectively – it is assumed that each center will port an initial version of its package to HPSC

Capability Status*

Product	Developer	TRL Level	Current Status
PRTOS	Wind River/ Green Hills	3	Needs to be ported to HPSC
CFE/CFS	GSFC/JPL	3	Needs to be ported
CM	GSFC/JPL	2	In development
PM	Several	3	Needs to be ported
ML	Several	3	Needs to be ported

HPSC

Development Environment

Capability Description*

Product	Capability Description
State-based Compiler(s) (SBC)	Discretely handle state as a variable – for high rel software
Static Analysis V&V (SA)	Statically analyze HPSC software
Dynamic Analysis V&V (DA)	Dynamically analyze HPSC software – for State-based Compilers



Advanced V&V Tools

Mission Applications

All missions will benefit from static analysis of software –most are currently using some form of this today. State based compilers and dynamic analysis will benefit all missions due to it's improved reliability, fault tolerance and correct-code generation capabilities.

Capability Status*

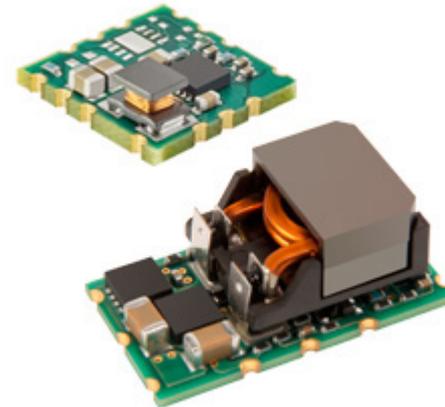
Product	Developer	TRL Level	Current Status
SBC	JPL/TBD	2	Concept work and early technology demo completed via (cancelled) STMD Robotics Software Validation project
SA	Various	6	Leverage existing COTS
DA	JPL/TBD	2	Concept work and early technology demo completed via (cancelled) STMD Robotics Software Validation project

HPSC

Power

Capability Description*

Product	Capability Description
Multi-Output Point of Load Converter	Discretely handle state as a variable – for high rel software
Smart Power Bus	Statically analyze HPSC software



Mission Applications

Modern computer systems of the HPSC ilk, are plagued with a large number of required voltages (but not at high current levels). In order to efficiently provide the requisite voltages, a single Point of Load (POL) converter with multiple outputs is required. Similarly, a smart power bus will provide the power system monitoring, command and status functions, built in test, and fault tolerance required for high reliability complex digital systems. All missions will utilize these elements, once developed and qualified.

Capability Status*

Product	Developer	TRL Level	Current Status
MOPOL	JPL/TBD	2	Fundamental elements exist but new packaging and POL design required
SPB	JPL/TBD	3	Leverage existing COTS design

HPSC Computer

Capability Description*

Product	Capability Description
Flight Computer Prototype Gen 1	Chiplet based prototype flight board with core flight software, RTOS and middleware
Flight Computer Prototype Gen 2	Update design with advanced space memories
NVM Board Prototype	Dynamically analyze HPSC software – for State-based Compilers



3U Space VPX HPSC SBC

Mission Applications

This set of tasks develops prototypes of the single board computers and ancillary modules that will be required for mission infusion. Once the prototypes are qualified and infused into a first mission or a tech demo, adoption by most, if not all, future mission is expected.

Capability Status*

Product	Developer	TRL Level	Current Status
FC1	TBD	1	
FC2	TBD	1	
NVM	TBD	1	

HPSC Roadmap

Instrumentation Technologies	Near Term Requirements			Mid Term Requirements			Far Term Requirements							
	FY18	FY19	FY20	FY21	FY22	FY23	FY24	FY25	FY26	FY27	FY28	FY29	FY30	FY31
<u>Advanced Space Memory</u>														
Volatile Chip (VC)	▲													
Non-Volatile Chip (NVC)			▲											
Volatile Stack (VS)				▲										
Non-Volatile Stack (NVS)						▲								
<u>Co-Processors/Accelerators</u>														
DSP				▲										
GPU				▲										
Vision				▲										
Neuromorphic						▲								
Micro-Chiplet				▲										
FPGA	▲													
<u>System Software</u>														
RTOS		▲	▲											
CFE/FCFS		▲	▲											
Configuration Middleware (CM)	▲		▲											
Parallel Proc'g Middleware (PM)		▲												
Math Libs (ML)		▲												

HPSC Roadmap

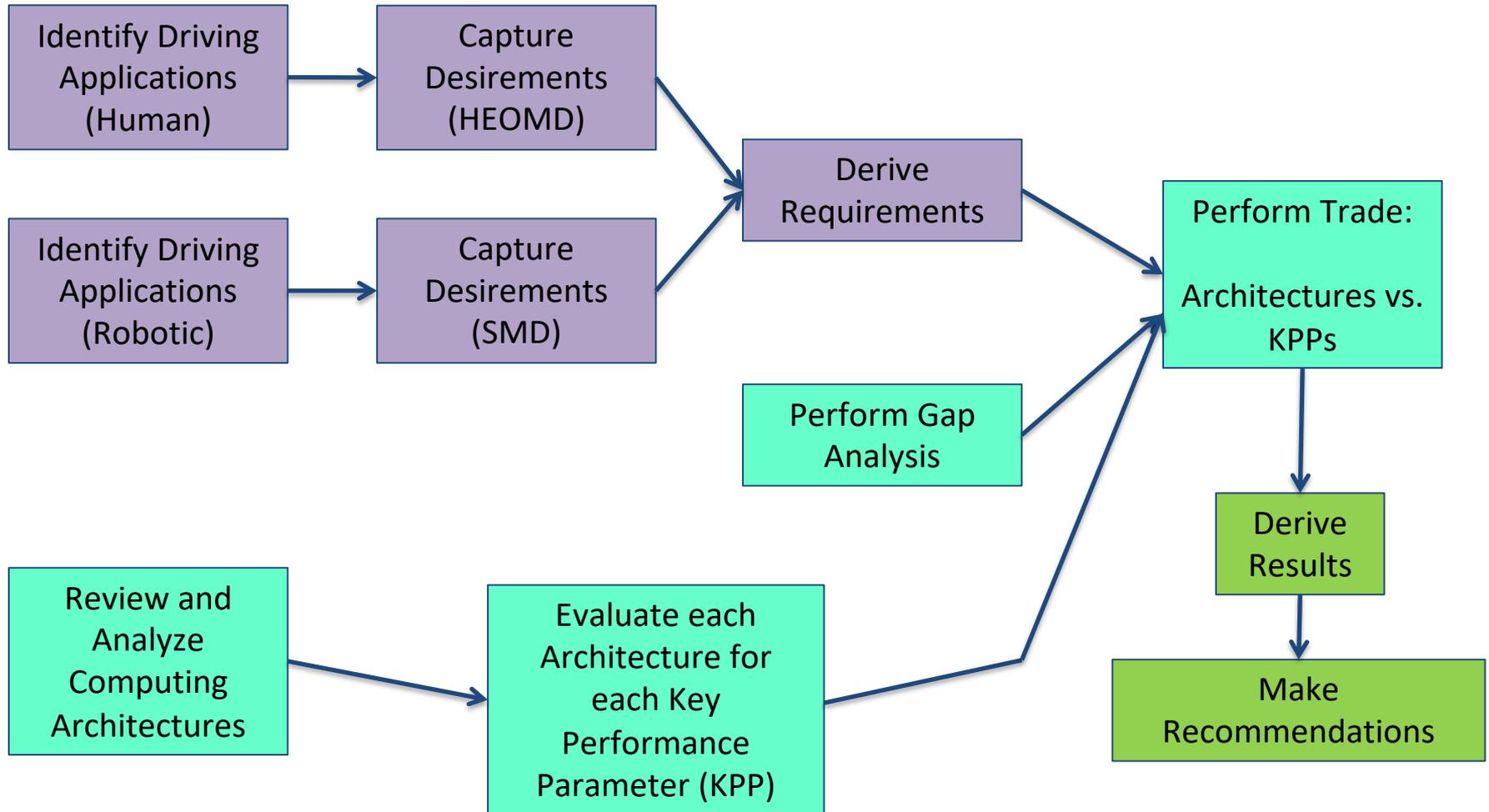
Instrumentation Technologies	Near Term Requirements	Mid Term Requirements	Far Term Requirements											
	FY18	FY19	FY20	FY21	FY22	FY23	FY24	FY25	FY26	FY27	FY28	FY29	FY30	FY31
Technology Gap Areas														
<u>Development Environment</u>														
State-based Compiler(s) (SBC)		▲	—————											▲
Static Analysis V&V (SA)		▲	—————	▲										
Dynamic Analysis V&V (DA)		▲	—————											▲
<u>Power</u>														
Multi-Output Point of Load Converter		▲	—————	▲										
Smart Power Bus		▲	—————	▲										
<u>Computer</u>														
Flight Computer Prototype Gen 1			▲	—————	▲									
Flight Computer Prototype Gen 2				▲	—————	▲								
Flight Computer Prototype Gen 2						▲	—————	▲						

Chiplet Project History

HPSC Formulation Study

NASA Space Technology Mission Directorate (STMD)

Game Changing Development Program (GCDP) commissioned
High Performance Spaceflight Computing (HPSC) Study



NASA Use Cases

HEOMD 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

SMD 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

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

Benefits to NASA Missions

Enabling New Mission Capabilities and Increasing Science Return Enhancing Crew Capabilities, Effectiveness, and Safety

- **Entry, Descent & Landing**
 - Enables reliable and safe landing in hazardous terrain by accommodating algorithms for Terrain Relative Navigation and Hazard Detection and Avoidance within one flight computer (benchmarked by Mars Program as requiring six (6) dedicated RAD750s)
- **Fast Surface Mobility**
 - 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 (e.g., plumes, dust devils) from ~10% to ~75%, with <5% false identifications, for increased and more timely science return
- **Vehicle Health Management**
 - Continuous monitoring and analysis of large amounts of vehicle data with responses to problems, reducing crew workload and improving vehicle maintenance during untended operations
- **Crew / Robot Interaction**
 - Robots can follow high-level instructions from crew or ground personnel while maintaining safe operations and interactions with the crew
- **Automated GNC**
 - Move intensive GNC-related computations onboard for faster docking, reduced collision avoidance planning; enable automated precision landing within an affordable power budget

**No longer need to size science / mission scope to flight computing capability.
Brings in-space computing closer to state-of-the-art terrestrial processing capabilities.**

NASA Flight Computing Requirements

As derived from the HEOMD and SMD 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

Future NASA use cases require dramatic improvement over RAD750 (~200 MOPS)

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 architectures:
 - 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

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 Evolvability	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

Note: KPP scores above mean are shaded green

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.

Ranking of the Architectures

The Runners-Up

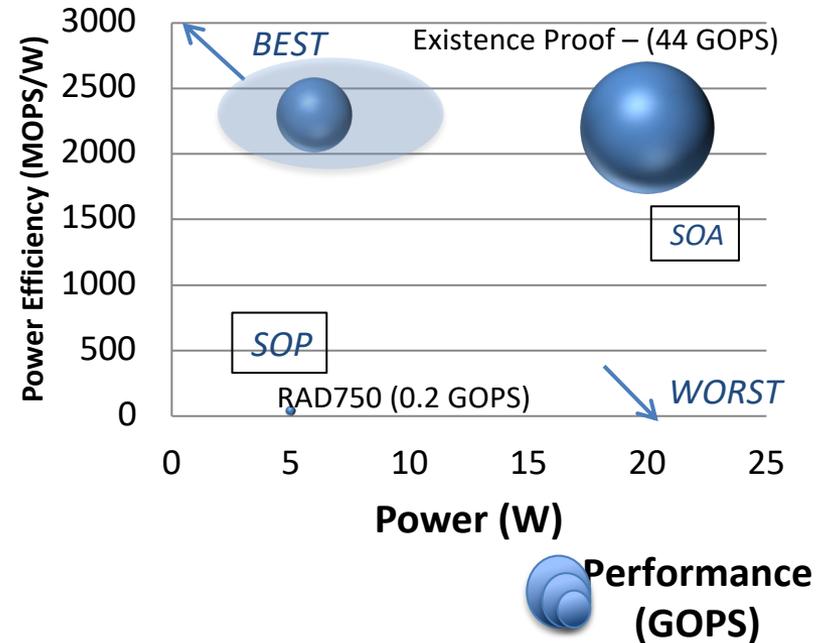
- COTS-based single-core – Honeywell
 - Implements Orion PPC Processor + TTGbE Bus Interface Unit on a single rad hard chip
 - Essentially one half of an Orion processing node
 - Designed to work in a self checking pair mode
 - Dramatically reduces size, mass, power of Orion processing node
 - May be suitable for other HEOMD platforms utilizing the Orion processing architecture
 - 225 MIPS theoretical peak at 7 Watts assuming use of external DRAM
 - 446 MIPS theoretical peak at 4.5 Watts assuming internal DRAM only
 - Supports ARINC 653 and Greenhills Integrity operating system
 - Provides high levels of fault tolerance and continuous real time self checking
 - Not suitable for high performance processing
 - 0.25GOPS vs 10-100GOPS
 - 1Gb/S I/O vs 40+Gb/S I/O
 - Does not support parallel processing
 - Not power scalable
 - Not extensible or evolvable
 - No provision for utilization in hybrid mode, i.e. with co-processor

Study Recommendation Multi-core

Rad-hard General Purpose

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)
- **Conducive to Power Scaling at core-level**
 - **Power Dissipation** issues to address fit within available investment resource envelope
- **Conducive to thread-based Fault Tolerance**
 - Fault detection/correction/isolation
 - Ability to segregate failed cores from the pool of available cores in support of graceful degradation
- **Scalable to 10x 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

Study Recommendations

Investment Focus and Approach

- Focus on **Rad-hard General-purpose Multi-core**
 - Leverage government and industry investments
- Issue a BAA for hardware architecture designs in FY13
 - Solicit flight computing system concepts
 - Prepare NASA requirements and benchmarks for early evaluation of architectures
 - A competitive initial phase, seeking innovative solutions and early risk retirement
- Product of the investment
 - Multi-core hardware chip with bundled real-time operating system (RTOS) and FSW development environment, integrated on an evaluation board
- 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

Challenge the community to develop an innovative, extremely high performance, low power, flexible, rad-hard GP multi-core processor within available budget and schedule

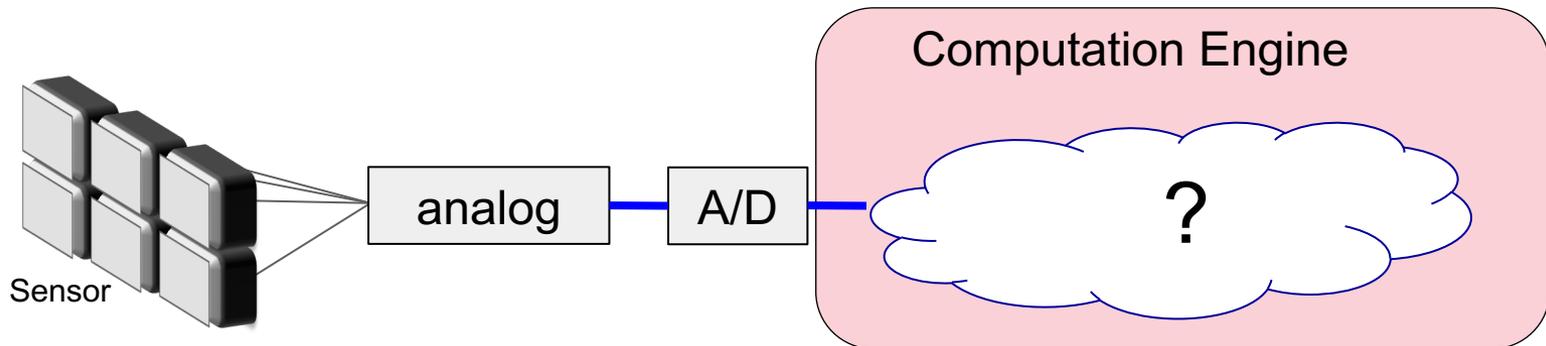
Enter AFRL!

AFRL NGSP BAA

AFRL Perspective

NGSP Study is ...

- Our quest to responsibly understand the future of computing for warfighting space platforms
 - What are the real requirements?
- Our strategy to best leverage and focus limited resources in the most impactful way
 - Not just about building a chip



NGSP : What is the future of computation in space?

AFRL Perspective

Why do a Trade Study?

- Starting point: Core-Functional Support Plan (Space Superiority, 2013*)

Impact Rating	Tech Need Date	AFSPC Technology Need
High	2016	Processor – 500 MIPS ¹ : > 500 MIPS - single core
Med	2018	Processor – 1200 MIPS (1.2 GIPS²) → 10 - 40 GIPS by 2022-2025*
Enh	2018	Next Generation Reprogrammable FPGA ³ : 18M-Gates
Med	2018	Rad-Hard Volatile memory: 256 Mbit
Enh	2018	Rad-Hard Non-volatile memory: 64 Mbit
Med	2022	AD Converter : 8-10 bits, 1.25 giga-sample per sec
Enh	2022	DC-DC converter family: > 90% efficiency, 1-10 Volts

*2015 and 2017 CFSP changes dates and computing performance characteristics

- Reviewed prior “market research”
 - BAE processing analysis conducted for SMC/EN (2010)
 - AFRL processing Request for Information (RFI) (2011)
 - NASA “High Performance Processing Study” (2012)
- Some evidence of a possible disconnect...more to the story?
- Study needed to ID next-gen computing needs & architecture options
- Keep in mind
 - AFRL is a laboratory (i.e., long-term focus)
 - Developmental timelines require we look beyond CFSP (e.g. 2018)

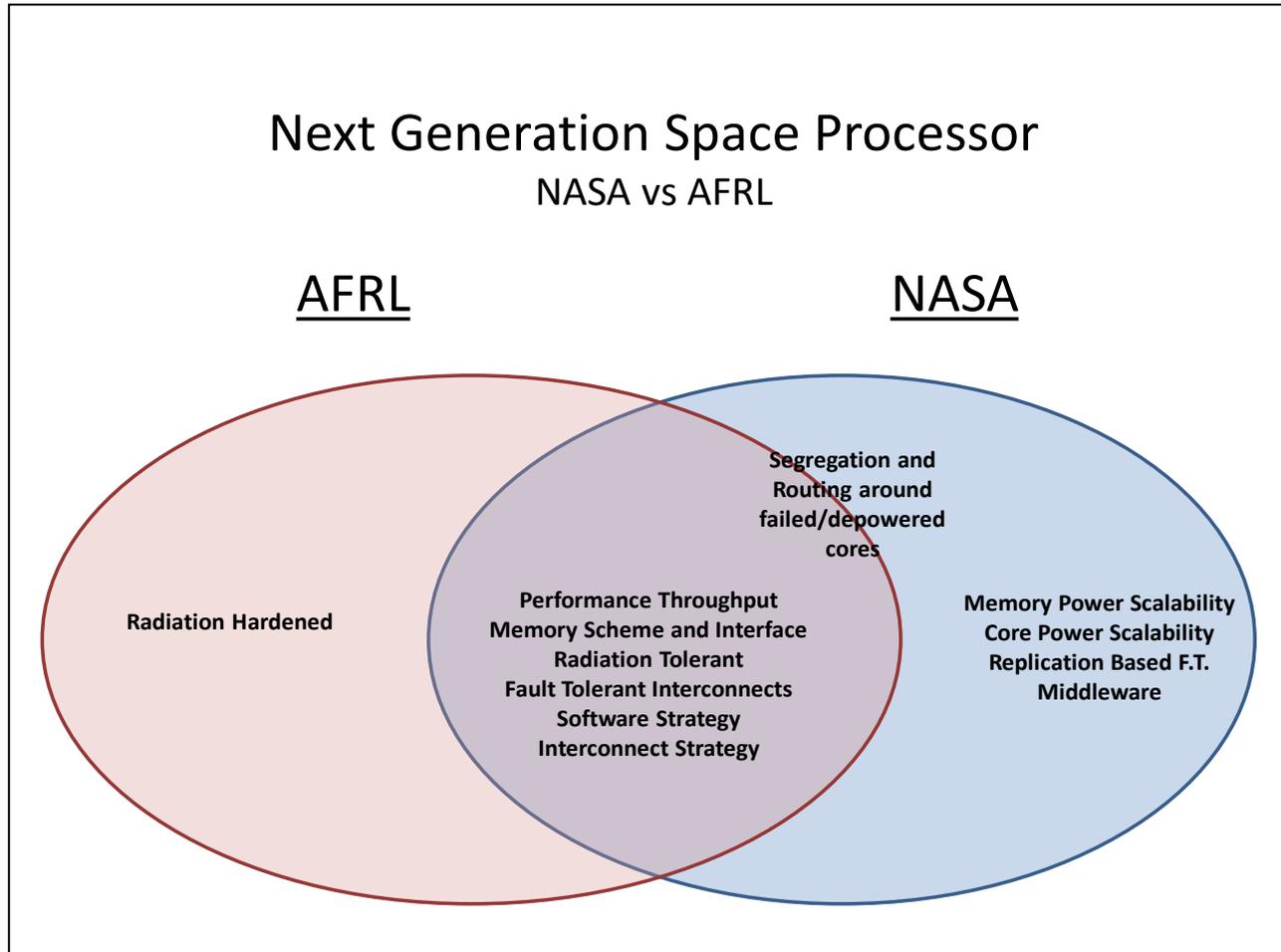
1. MIPS: Million Instructions Per Second
2. GIPS: Giga Instructions Per Second
3. FPGA: Field Programmable Gate Array

AFRL Perspective Study

How We Are Doing the Trade

- Experience over past two decades: Answer won't be cheap
 - What can we leverage?
 - Who could we partner with?
- NASA agreed to cost share (50/50) the study

AFRL vs. NASA Primary Needs and Concerns



There was sufficient common ground to justify a joint AFRL-NASA program.

AFRL Issued, Joint, Innovation Phase BAA

- Two Part Program:
 - Part 1 – Study – Generate USAF future Space High Performance Processing Requirements
 - Culminates in Technology Interchange Meeting (TIM) to review USAF Requirements
 - NASA and AFRL will issue a joint requirements document
 - Part 2 – Refine Processor Architecture and Computer Design
 - Generate Model/Simulation of Proposed Architecture
 - Run Benchmarks (CFE) to Evaluate Performance of Proposed Design
 - Define Implementation Approach
 - IP, IP Modifications Required, Foundry, End to End Design Flow, Software Stack, V&V

AFRL BAA-RVKV-2013-02
Proposal Due Date 5/29/13

BAA Requirements

Part 2 Requirements – A Starting Point for Purposes the BAA

- Provide a minimum of 24 processor cores to support both highly parallel applications and to provide a high degree of granularity for power management, fault tolerance and program unit distribution.
- Providing > 10GFLOPS, multiple 10Gb/s I/O and DDR 3 Memory Ports at 7 Watts
- 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 (<<1W)
- Based on commercially available hardware and software IP (processing cores, external I/O and memory interfaces, software stack and development environment)
- Able to reset individual cores or a cluster of cores, as determined by smallest unit of granularity
- Radiation hard to at least 1 Mrad TID, Latch up Immune to an LET of at least 90, with a hardware-uncorrected SEE rate of not greater than 0.01 event/day in Adams 90% worst case GEO environment
- Interoperable with other high performance computing architectures, e.g., FPGAs

*****Proposers encouraged to offer alternatives *****

AFRL BAA-RVKV-2013-02

Proposal Due Date 5/29/13

BAA Requirements

- Identification of process and/or RHBD library to be used, along with test data to substantiate claims of radiation hardness
 - RHBD technology with test data showing TID tolerance to 300krad, SEL immunity to 70MeV
- Simulation/model results for a set of NASA-defined benchmarks (FFT, search algorithms, etc..)
 - Performance thresholds will be specific to each benchmark
- Simulate/model results of fault response and power management
 - Demonstrated ability to operate through faults and restore correct operation
 - Demonstrated ability to dynamically manage power under software control
- Management/ Development plan that makes a credible case that TRL 5-6 can be achieved 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 (including margin)
 - Detailed development cost estimate (including reserves)
 - Risks identified and risk management approach provided

AFRL BAA-RVKV-2013-02

Proposal Due Date 5/29/13

And the Winners Were: BAE, Boeing, Honeywell

Part A Approach: Ask the People Who Build the Systems

- Assemble a team of space computing experts



- Leverage substantial prior NASA analysis
- Determine AF needs:

Formulate the Right Questions

- Performance
- Radiation Mitigation
- Power / Thermal
- Memory
- I/O, etc.

Interview the Right People

- SMC
- Primes
- NASA
- Others (NRO, NRL, etc..)

- Develop reference architecture and benchmark against established needs (blueprint for part 2)

Part A Approach: Ask the Right Questions

- Gov't gave vendor teams overarching guidance
 - Timeframe 2020-2030
 - ID applications that drive space computing
- Ktrs developed detailed questions
 - Gain insight into computing trends
 - Result: 50/50 mix of needs vice enabled capabilities
- Independently surveyed stakeholders





Part A Guidelines from AFRL



- **What are the near and long-term (through 2030) Air Force space-based applications that drive spacecraft computing? Opportunities for tech insertion?**
- **What are the considerations imposed on space computing by these applications?**
- **What are the current or near-term computing systems offerings, and will they provide the required capabilities?**
- **Which computing architecture(s) provide the maximum return on investment for the Air Force?**
- **How can the Air Force most effectively invest its limited \$\$ to effect the development of the required tech?**
- **What requirements are imposed on wafer fabrication, mask making, third-party intellectual property, and other factors by the requirement for trust as discussed in DoDI 5200.44?**



Part A Approach: Ask the Right People

- Deliberately sought engagement from SMC and S/C Primes
 - SMC/EN & XR facilitated interactions with SMC SPOs
 - Contractors surveyed primes directly

Organization	Mission System
Northrop Grumman	AEHF Payload; STSS Demo; SBIRS
Lockheed Martin	GPS IIR,M; AEHF Bus; SBIRS
Boeing	GPS IIF, AEHF (proposer), SBSS, WGS
Raytheon	STSS Demo, AEHF & GPS Ground; AFRL HTI Demo
Excelis	GPS
Ball Aerospace	SBSS

- NASA coordinated across their centers
 - Many needs in common (e.g. autonomy, image processing, mission assurance)
- Engaged NRO (multiple points), NRL, and others

Part A Results

- Two different methodologies
 - Direct, open ended questions about performance needs
 - Multiple choice polling with consolidated results
- Common themes emerged
 - Hybrid computing architectures needed
 - **No current or planned solution meets all surveyed mission needs**

Key Takeaways of Part A (Reqs Study)

- The full scope of mission needs call for a variety of architectures
 - RAD5545a appears to meet stated current needs
 - Future needs are more complex
 - **No existing/emerging space processor meets all surveyed needs**
- Planned missions drive hybrid computing architectures
- Several enhancing capabilities enabled, e.g., autonomy, adaptive nulling, and software defined radio schemes
- Informed objectives for Part B: Architecture Concepts... (see next slide)

Part B: Architecture Concept and Sim/Modeling Phase

- Objective
 - Reduce technical risk via the **development of multiple hardware architecture designs** for a rad-hard, “general-purpose” multicore flight computing system
 - **Evaluate those designs against** a set of **benchmarks** representative of the future NASA applications

Part B: Design Objectives (informed by Part A and NASA HPSC Study)

Computational Throughput	Power
20 GOPS + 10 GFLOPS	10W (threshold)/ 7 W (goal)
1 GOPS + 20 GFLOPS	10W (threshold)/ 7 W (goal)
40 GOPS + 0 GFLOPS	10W (threshold)/ 7 W (goal)
0.3 GOPS + 0.1 GFLOPS	1 W or less

- Dynamically power scalable at core level granularity without disrupting system ops
- Fault tolerant interconnects between cores and to external I/O and memory devices
- Built in self test with faulty core bypass and power up into known good state
- Natively supports N-Modular Redundancy where N=2, 3
- Segregation and routing around failed and depowered cores
- Radiation tolerant to 1MRad, Latch-up Immune
- Interoperable with other high performance computing architectures, e.g., reconfigurable computing FPGAs
- Real time support; System software and application development environment

Baseline design a guideline only: encouraged trades with corresponding reasoning

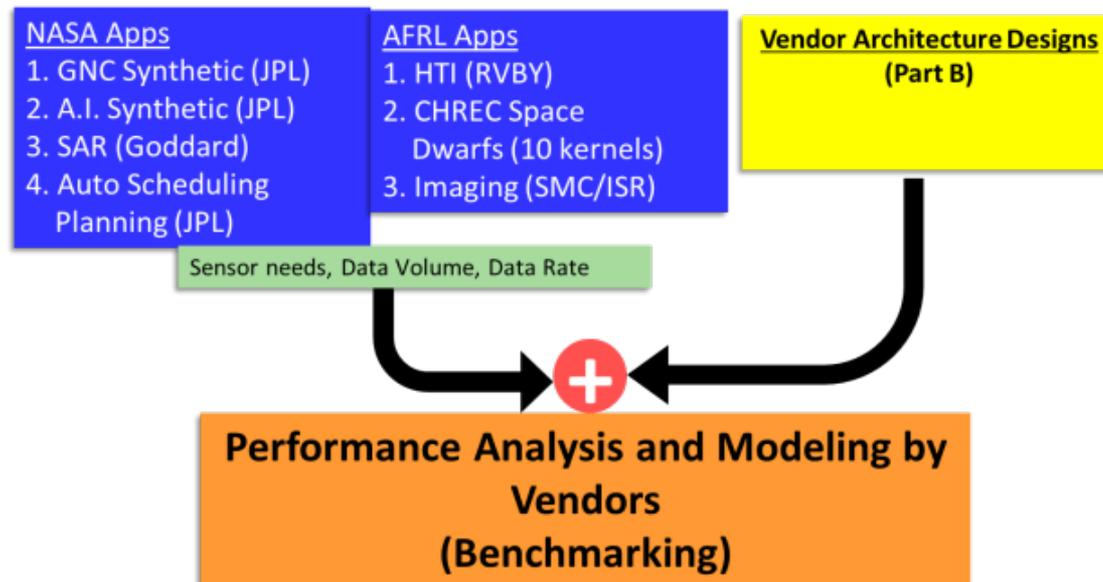
Part B: Concept Designs

- Each vendor using a “many-core” ARM-based architecture
 - Hybrid approaches based on *big.LITTLE* concept from ARM
- Conducting trades with respect to core sizes, interconnect pipelines, cache hierarchy, memory controller, co-processor integration, etc..

Part B: Sim/Modeling Phase

“Benchmarking”

- Application Benchmarks provided as the most reliable means to assess real-world performance
- Scope of benchmarking activity was aggressive to glean as much as possible about the design concepts against real world application information.
- Completion of benchmark activity, in part or whole, was valuable to assessing performance capabilities



In aggregate, the level of benchmarking achieved gave considerable performance information about the various design concepts devised

Benchmark Result Summary

- Enabled appropriate sizing of cache memory
- Showed efficacy of a baseline fault tolerance feature set
- Showed efficacy of a many-core architecture
- Showed efficacy of an ARM based compute engine core to enable a broad range of NASA and USAF apps
 - Power efficiency, power management, performance capability
 - Showed an array of ARM A53 cores is optimal
- Showed cost & performance benefits/capabilities of 5 different architectures
- Conclusions:
 - Provides cost-effective tech options for a next-gen space processing architecture that meet future capabilities
 - Extensible by providing allowances for more ARM cores
 - Portable to advanced semiconductor process/logic libraries

Government Team Conclusions and Non-Advocate Review

HPSC – Non Advocate Review (NAR) Panel Conclusions

1. The NAR panel concurs with the need for a new, more capable, space processor
2. Three independently developed vendor architectures were proposed, all met government team goals except for budget
3. The vendors all proposed large chip formats dominated by ARM A53 clusters – apparently unaware their designs were so similar. The vendors thus cross-validated each other on the following points, for which the NAR panel concurs:
 - a. The applications studies show the emergent architecture is effective and has limited risk
 - b. Secondary differences in the architectures are not critical, such as GPUs and accelerators
 - c. Implementation as a single large chip will be too expensive
4. The government team proposed adding an additional feature level to the vendors' common direction by partitioning it into a modular, scalable, "Chiplet" implementation. This includes transitioning some hardware features to middleware. The Chiplet approach has the following advantages:
 - a. Enables use of smaller chips, which are proportionally less expensive and have less development risk
 - b. Increases flexibility by supporting configurations that are either scaled up or scaled down to meet needs of different missions
 - c. System flexibility would allow increased use by other organizations
 - d. The risk reduction of the innovation phase will carry forward to the Chiplet approach
5. The NAR panel concurs with the direction of the NGSP government team proposal that relies on the Chiplet approach
6. The NAR panel report will have more detail

HPSC Chiplet Procurement

HPSC Chiplet Procurement

NASA GSFC RFP

- HPSC Synopsis Release – 9/15
- RFI Release – 10/15
- Industry Day – 11/15
- RFI Inputs Received – 11/15
- Draft RFP Release – 4/16
- RFP Release – 6/16
- Proposals Received – 7/16
- Contract Award – 3/31/17
- Kickoff – 5/3/17