2021 NEPP ETW

ARM Radiation Testing Update and Raspberry Pi Guideline

Steven M. Guertin

Jet Propulsion Laboratory / California Institute of Technology

Pasadena, CA

This work was performed at the Jet Propulsion Laboratory, California Institute of Technology,
Under contract with the National Aeronautics and Space Administration (NASA)
This work was funded by the NASA Electronic Parts and Packaging Program (NEPP)

The cost information contained in this document is of a budgetary and planning nature and is intended for informational purposes only.
It does not constitute a commitment on the part of JPL and/or Caltech.
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.

Outline

- ARM Update
  - Approach Overview
  - A5 Testing
  - Future Work
- Raspberry Pi Guideline
  - Overview
  - FY2021 Approach

ARM Processor Testing Overview

- Understanding processor testing for space
  - What’s it going to do with radiation
    - Calculation errors – possible incorrect operation
    - In fact, falling on its face is more likely, requiring reset
    - May permanently fail
  - Test approaches
    - Low-level structures – the old approach, and still used for RHBD devices
    - Application based
- Build collaborations
  - Maximize budget impact by covering more of the space
  - Identify key mission needs – reliability, cost, performance, relevant data
  - Develop better metrics to enable comparison of devices
    - For example, the entire SWAP required to implement a system
    - Dissimilar processor architectures should not always be compared
- Key issues
  - Limited documentation, expensive evaluation equipment, complex system design and complex error modes, potential severely limited hardware options (partner chips, etc.)

Advanced Processors – ARM & Flight/RHBD - collaborative with BAE Systems, HSPC, others

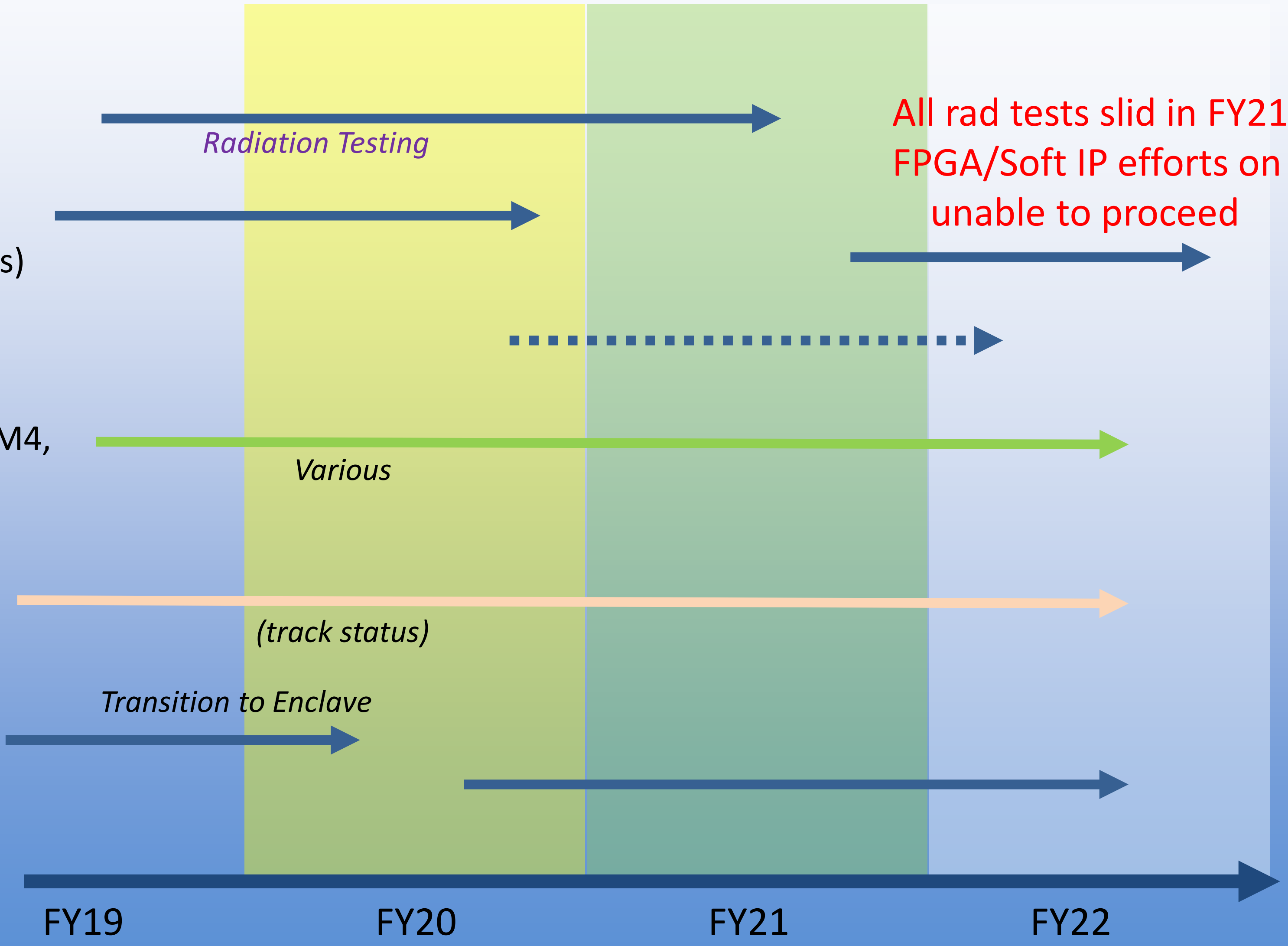
Compare soft- vs. hard-ip ARM devices
- A5 Microcontroller SAMA5D3
- A5 soft IP
- A5x, 7x or 9x (with FT elements)

ARM Fault Tolerance (by design)
- Dual R4 or R5

Additional Vorago/Cobham M0, M4, R5, etc….

High Performance Spaceflight Processor (HPSC)

Snapdragon General
- 855
- 865 (Enclave approach)



Radiation Testing

All rad tests slid in FY21
FPGA/Soft IP efforts on A5
unable to proceed

Various

(track status)

Transition to Enclave

FY19 FY20 FY21 FY22

*FY18-21: ARM SEE Testing

ARM Architecture SEE Fault Handling Across Implementations

Description:

ARM devices are currently being used in many active missions as well as being an architecture favored for future processor designs. NASA and ARM lack understanding of implementation and effectiveness of ARM fault handling. ARM is fab-less, with various design-time configured parameters, each instantiation of ARM IP can have different SEE behavior.

In FY18/19 it was observed that fault-tolerant ARM devices have significant SEFI problems. ARM has indicated this may be due to the hardware manufacturers not implementing fault handling correctly, users not configuring it correctly, or potential flaws in the actual IP. FY20 activities are currently focused on implementing and testing A5 cores in nominal operation (FPGA&. In FY21 this will be expanded to target configuration and operation of ARM features to improve fault handling.

Arrangements with ARM and UFRGS will be finalized in FY20 to support these activities in FY21. Collaboration improves overall test capability, and understanding of test results. Because the A5 is relatively old, there is minimal anticipated risk to making results public.

FY21 Plans:

1. Increase coverage of test capabilities targeting fault handling of A5 core and ARM-supported features (procuring DUTs, etc).
2. Fault handling/SEE injection testing of SAMA5D3 – focused on A5 core (beam charges anticipated ~10k)
3. SEE testing of FPGA implementation of A5 RTL (w/ARM support for FT configuration) (beam charges anticipated ~10k) – This was eliminated due to funding and partner constraints.
4. (Added) Develop system to allow very low rate (<10/cm^2-s) flux using a beam chopper for TAMU and/or LBL

Schedule:

						Month - FY21							FY22	
Task	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Test Plans for Hard & Soft Core A5 IP			🟦	🟦	🟦	🟦									
SEE Tests for Hard & Soft Core A5					🟩	🟩	🟩	🟩	🟩	🟩					
Hard Core A5 Test Report									🟨						
Soft Core A5 Test Report											🟦				
ARM Collaboration	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒
ETW Presentation								🟧	🟧						
End of year Report												🟩	🟩	🟩	🟩

Lead Center/PI: JPL/Guertin
NASA Co-Investigator(s): Andrew Daniel/JPL

Deliverables:

Completion of test set up and test plans (Q2 FY21)
SEE test report for hard silicon devices SAMA5D3 & SAMA5D2C (latter comes from UFRGS collaboration)
SEE test report for FPGA A5 fault tolerant instantiations/options
ETW presentation of preliminary findings (Q3FY21)
Final report of findings (Q1 FY22)

NASA and Non-NASA Organizations/Procurements:

Hardware Procurements Only
Partners: E.Wyrwas/GSFC, P.Rech/UFRGS, R.Jeyapaul/ARM

National Aeronautics
and Space Administration

*FY21: Raspberry Pis for Space

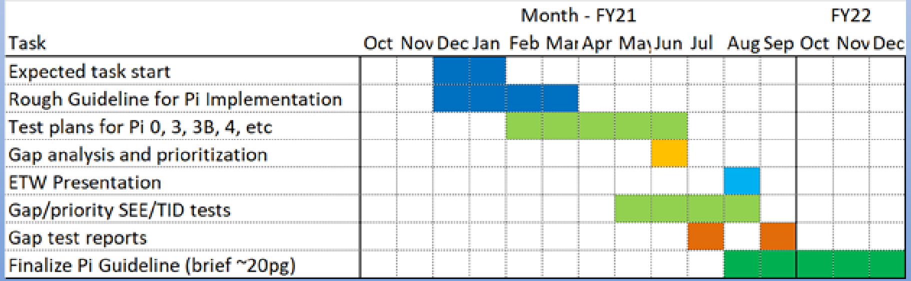
Description:

Raspberry Pis are already available for people to run code using the Astro Pi program through ESA on the ISS (using a hardened Raspberry Pi). This platform was made available in 2015. This has provided an exciting connection between the public and space hardware.
In the meantime the Raspberry Pi 4 and 0 have become available, and there is evidence the Pi 0 can survive to over 200krad(Si), most likely due to it having a very small number of support devices.
The current environment of using more and more commercial equipment in low criticality or shorter-duration missions means that the newer Raspberry Pi models are a useful way to obtain valuable data and engage the public at the same time. Radiation hardened versions of existing Pis can be created relatively cheaply if higher criticality systems become interested as well.

FY21 Plans:

1. Catalog existing Raspberry Pi test data & gap analysis
2. Completion of test set up and test plans (Q2 FY21)
3. Supplemental SEE/TID testing at whatever facilities are available for protons or light heavy ions.

Schedule:



	Month - FY21											FY22		
Task	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Expected task start			🟦	🟦											
Rough Guideline for Pi Implementation			🟦	🟦	🟦	🟦									
Test plans for Pi 0, 3, 3B, 4, etc					🟩	🟩	🟩	🟩	🟩						
Gap analysis and prioritization									🟨						
ETW Presentation											🟦				
Gap/priority SEE/TID tests								🟩	🟩	🟩	🟩				
Gap test reports										🟧		🟧			
Finalize Pi Guideline (brief ~20pg)											🟩	🟩	🟩	🟩	🟩

Lead Center/PI: JPL/Guertin

Deliverables:

Test reports for damaging SEE, TID, system-level high current, and SEFI, as appropriate
ETW presentation (Q3FY21)
Brief Raspberry Pi radiation implementation guideline, focused on coverage and alternates, not depth

NASA and Non-NASA Organizations/Procurements:

Hardware Procurements Only

To be presented by Steven M. Guertin at the 2021 NEPP ETW June 14-17, 2021, NASA GSFC

*Task Partnering

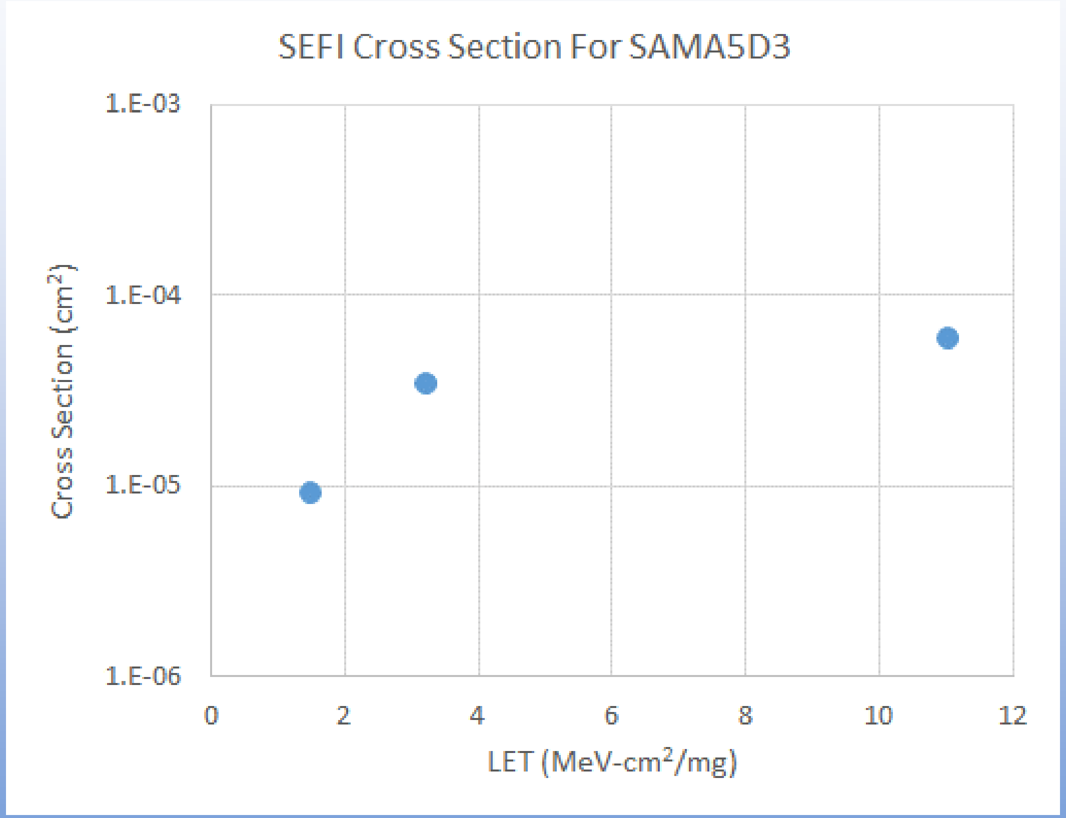
- Engaging in collaborative efforts:
  - NSWC Crane
  - Carl Szabo, Ed Wyrwas, Ted Wilcox, and Ken LaBel, GSFC
  - Larry Clark, ASU
  - Heather Quinn, LANL, and other members of the Microprocessor and FPGA Mitigation Working Group
  - Sergeh Vartanian, Andrew Daniel, and Greg Allen, JPL
  - Vorago Technologies – collaborating on hardware/plans
  - Paolo Rech – GPU/Applications, UFRGS, ARM Collaboration
  - Intel – informally
  - BAE Systems
  - AFRL
  - ARM collaboration – realigning based on A5 efforts

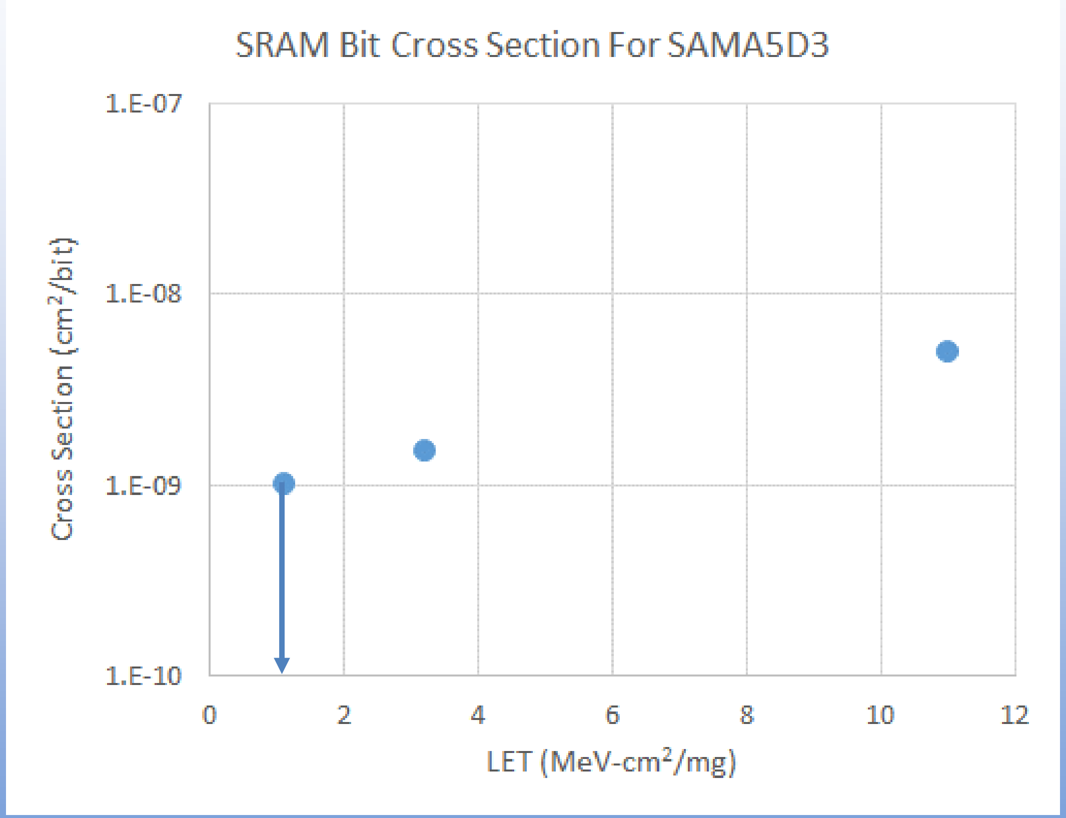
- Looking for additional collaborators
  - Tester side – are you testing processors?
  - Manufacturer side – knowledge or hardware support
  - Application side – specific applications…

ARM A5 General Approach

- A5 is one of the lower end ARM processors, but we have some access to how it works via collaborators
  - Evaluate effectiveness of fault approach
  - Understand SOC-integration impact on fault approach in SAMA5D3 and SAMA5D2 (latter via collaborator)
  - Identify operational mode, OS, and other configuration details that impact fault handling
- First stage is basic error performance of the A5 in SAMA5D3 <- FY19
- Second stage is errors in SAMA5D3 with various FT config <- ~~FY20~~(->FY21)
- ~~Third stage is A5 performance when implemented on an FPGA <- FY21~~
  - Redirecting stage 3 to finalize A5 effort -> forward plan/recommendations for other ARM-based devices.

*Basic Sensitivity (from ETW2020)





- Crashes were more likely to occur than execution errors

- Memory bit errors were easiest to extract with debug tools
- Software crashes limited in-situ data extraction (i.e. memory test programs).

FY21 Effort

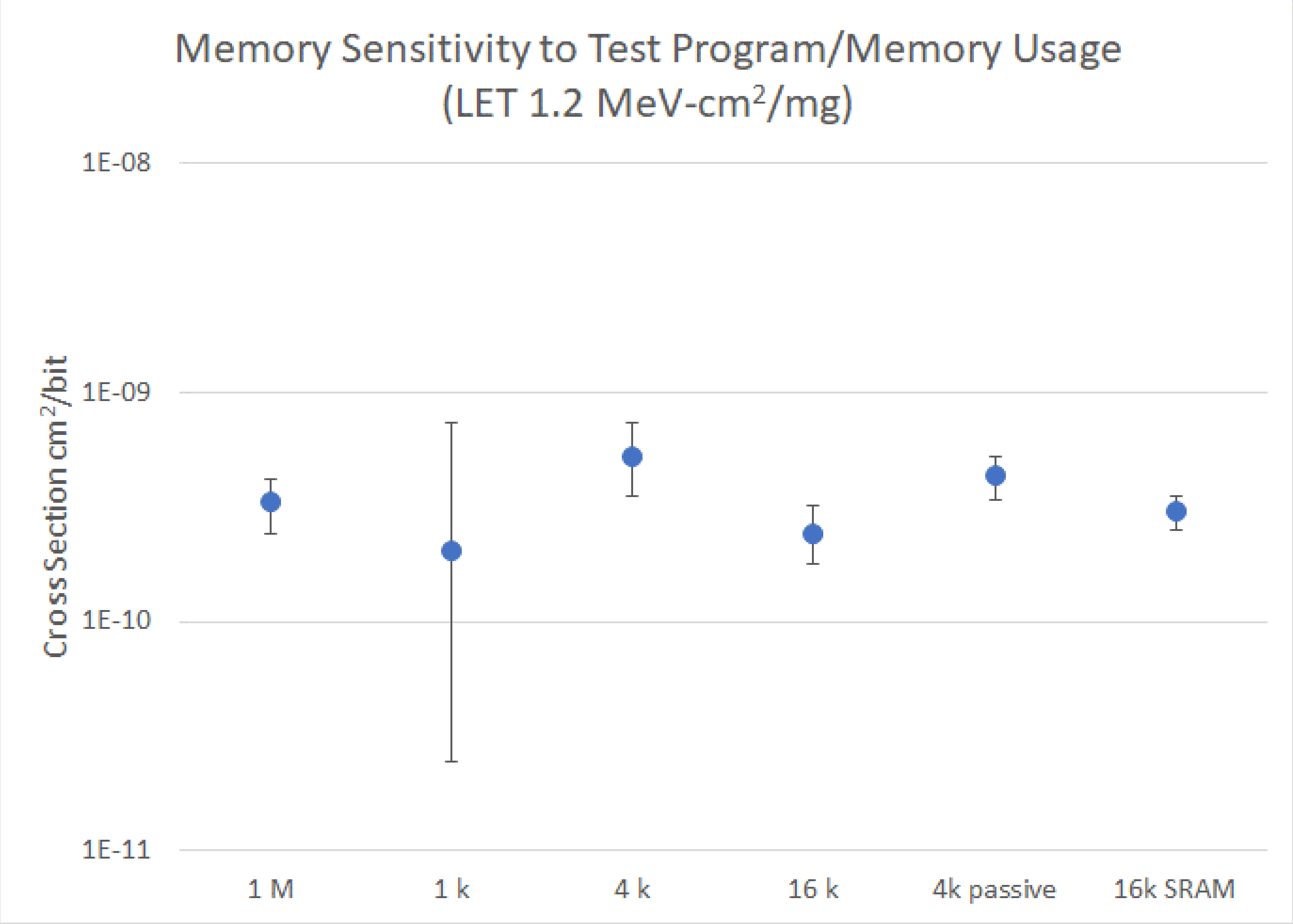
- Primary goal was to show if SEEs in A5 processor in SAMA5D3 were the primary source of observed errors.
  - Codes were designed to operate in the A5 core and provide minimal external communication to avoid radiation sensitivity in peripherals and buses.
- Test codes focused on the sensitivity of internal-to-the A5 core, and external-to-the A5 core.
  - Primary approach was to keep test codes using almost none of the on-chip services. This was achieved by utilizing UART operations within the SAMA5D3 U-Boot architecture.
  - Then codes were designed that would stay “in” or break “out” of the A5 processor.
    - If “break out” results in a higher SEFI rate, then it is very difficult to test the A5 core by itself.
    - Previous SOC tests have shown significantly increased or significantly decreased SEFI rates by doing this.
- Spoiler Alert: Primary result was that test codes that stayed inside the A5 were inherently less stable – which is a good sign for targeted testing the A5.

*Test Codes

- Key was to show system sensitivity to different sizes of programs and use of cache vs. off-chip memory.
- FFT codes (uses recursive architecture) – uses 24 bytes for each size count (3 arrays, complex 32-bit values)
  - 16 – basic, quick FFT – for efficient test of underlying test program
  - 256 – larger (reliably fits inside the data cache)
  - 2048 – definitely requires external memory
- Memory test codes:
  - 4k – test of 4k*4 = 16kB of memory (SAMA5D3's A5 processor has 32k data cache)
  - 1k – smaller version
  - 16k – test of 64kB of memory (definitely larger than on-chip cache)
  - 1M – test of 4096kB of memory (definitely must go off-chip)
  - 4k – dwell version – writes, waits, then reads (previous versions continuously write/read)
  - 16kSRAM – test of 64kB of on-chip SRAM (directly accesses the memory map of one bank of SRAM – device has 2 banks)

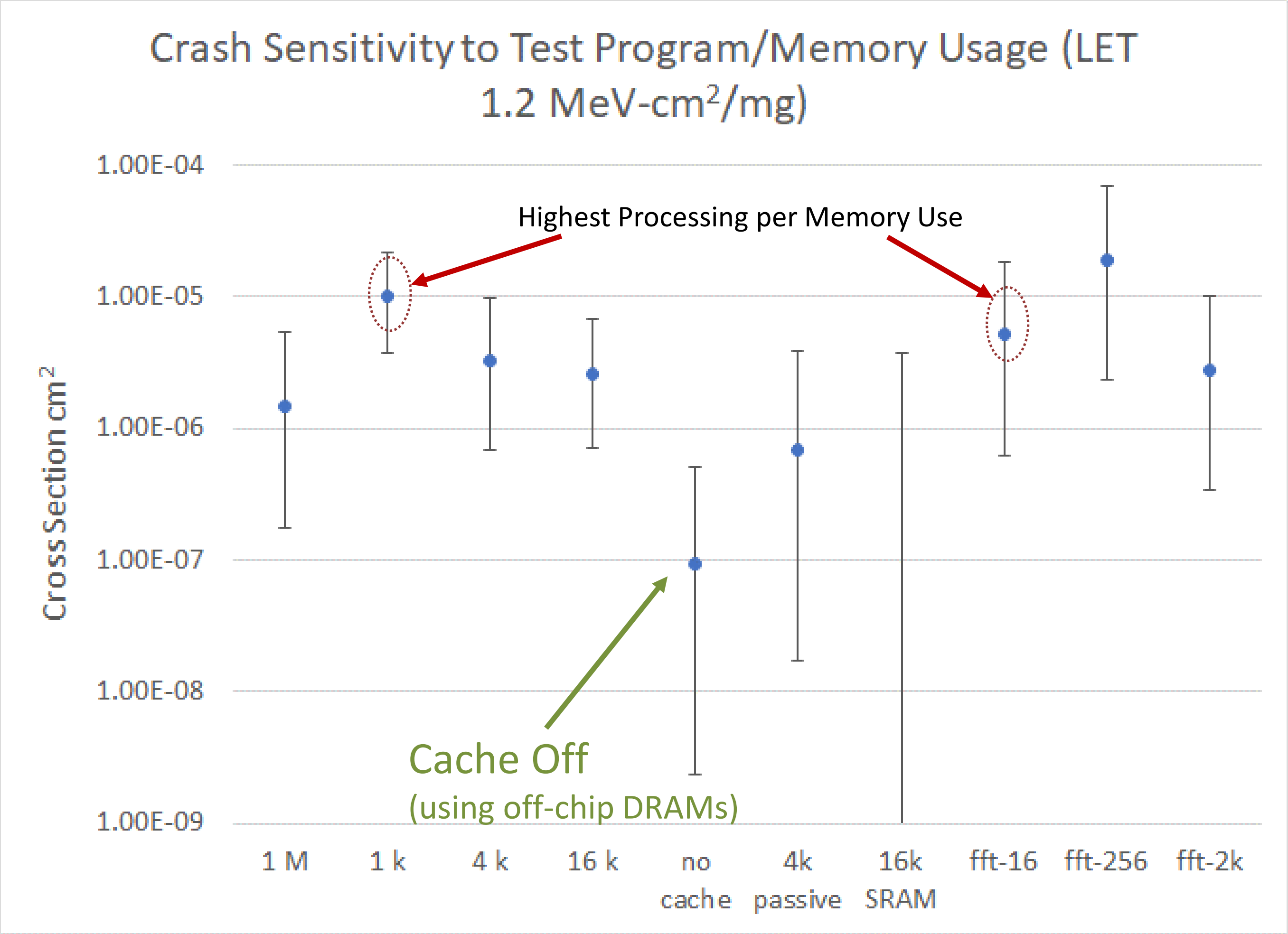
Memory Sensitivity to Use/Test

- All tests showed similar sensitivity to upsets in on-chip SRAM and cache memories.
  - Some tests could be corrected for duty cycle of memory.
  - However, there still remains ~2-4x variation – highly driven by low statistics under some test types.
- Tests scaled for use of caches – both amount, and profile.
  - Off-chip memory access
  - FFT tests have complex profile due to recursion and use of higher-order FFT points.



Tests too big for cache act like they use ~1/2-1/4 of the data cache.

Crash Sensitivity to Memory Usage



- Crash/SEFI sensitivity showed clear correlation to cache use, but not much else.
- With caches enabled, crashes were about 1e-5/cm^2. With them disabled, this drops 30-100x.
  - Performance drops about 10-20x, so the performance benefit is minimal and it takes much longer to get a result.

A5 Conclusion

- Primary goal: show if SEEs in A5 processor in SAMA5D3 were the primary source of observed errors. – They are!
  - So in this chip, it is relatively easy to test the A5 cores.
  - In big SOCs, it is unlikely that an individual processor core can be isolated this way.
- Test codes that stayed inside the A5 were inherently less stable – which is a good sign for testing the A5 with this platform.
  - When the test focus is the A5, the radiation sensitivity is increased.
  - Unfortunately, because the A5 has no cache mitigation, this means test codes did not function very well when isolated to the A5.
  - The SAMA5D3 can be used as a good hardware platform for comparing A5 simulation results to actual hardware.
- The A5, however, is the least fault-tolerant A-series processor offered by ARM.
  - Another platform where a similar exercise makes sense, though possibly much more complex, is the Raspberry Pi
  - Raspberry Pi Compute Module 4 uses a BCM2711 with quad-core ARM A72 processors
    - It is unclear if Raspberry Pis can be used to try to isolate the A72 cores, since they are usually configured to run complex Operating Systems…

Raspberry Pi’s in Space

- Currently evaluating guidelines for:
  - Converting a Raspberry Pi-based design to a flight program various levels of required reliability.
  - Evaluating requirements of computing resources against existing radiation hardened alternatives.
  - Help assess risks of flying commercial units – and methods to ensure results from “other missions” are applicable.
- Ample evidence exists that many small satellite/Cube Sat systems are interested

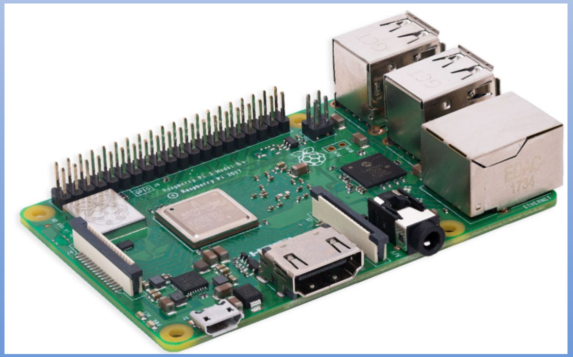
Obligatory photo of Raspberry Pis floating in space!

(Luca Parmitano) https://www.raspberrypi.org

FY2021 Approach

- Primary deliverable is a brief guideline on recommendations for use of Raspberry Pi for flight.
  - Key info for rad hard alternates – depending on use
  - Tailored to key issues of:
    - Environment
    - Applicability of test results and prior history of other Pi's
  - Best practices for configuration (especially fault tolerance of caches to improve on A5-like SEE performance)
- Driven by architecture review and available data on flight use
- Plan is to support this with radiation testing
  - Limited availability of beam facilities during the period of this task
  - High risk that very little data will be gathered before this task times out
  - Possible augmentation with additional testing in the future
    - To support an update of the brief FY21 guideline

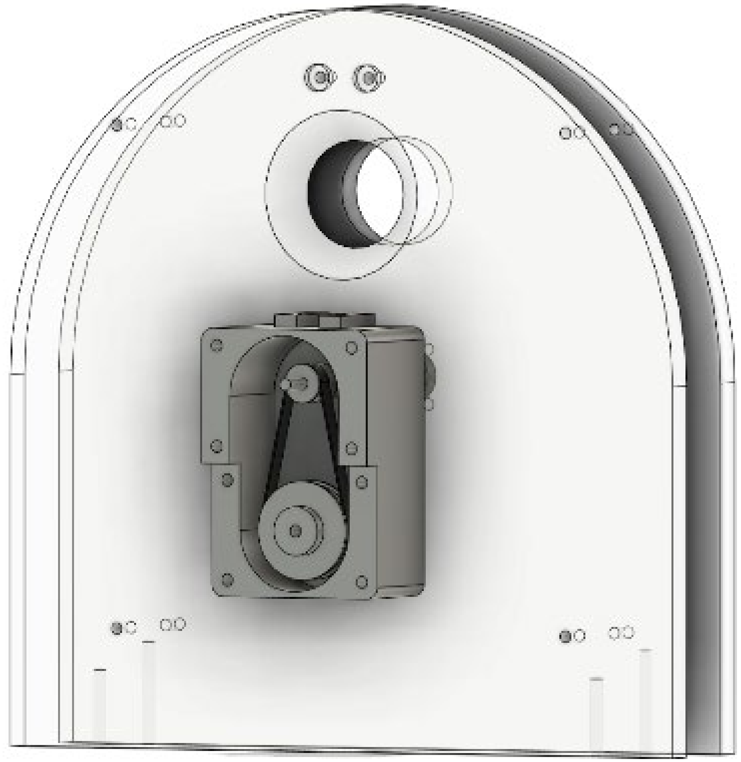
Raspberry Pi Compute Module 4

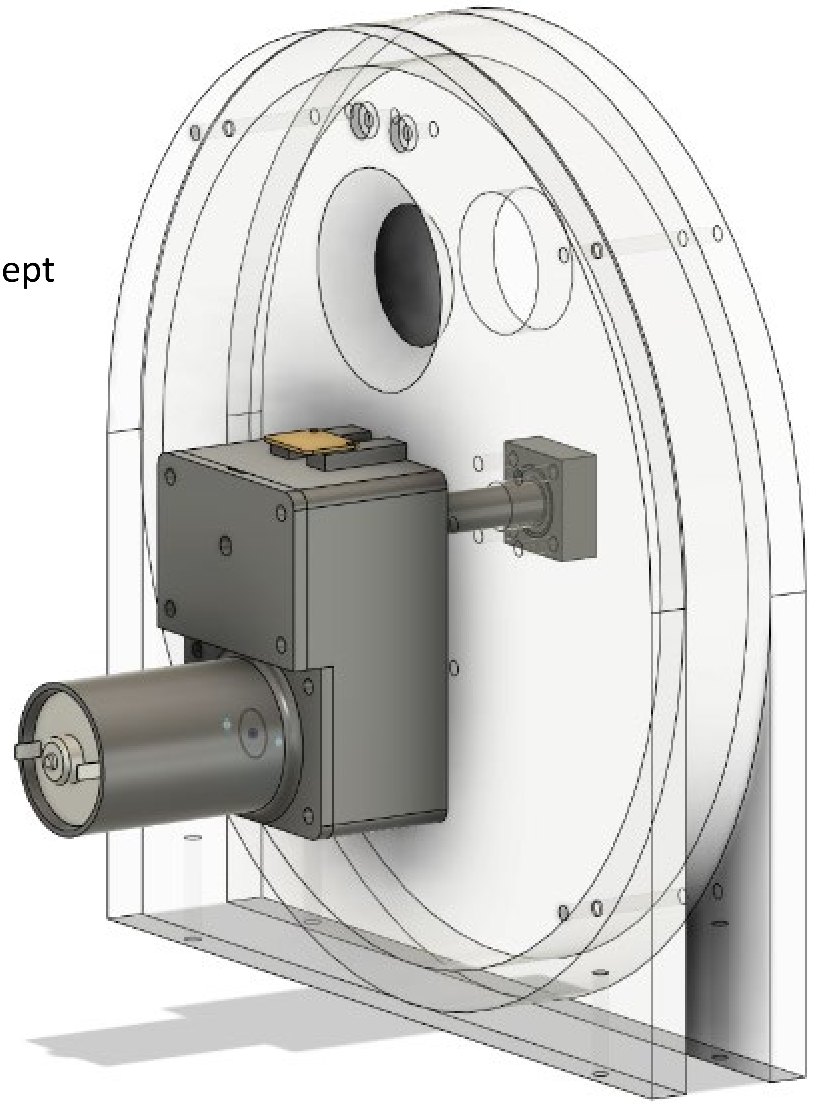


Raspberry Pi 3 B+ (in Astro Pi as of Sept. 17, 2017 upgrade)

Beam Chopper - ~1% Duty Cycle

- Allows “statistically reliable” delivery of ~10 ions/sec
- Fully prototype ready – thanks to Ryan Melendez and intern Sam Delaney!
- Will support lower LET test points with very high cross section
  - The need is limited at higher LETs because devices are likely saturated except for rare destructive events

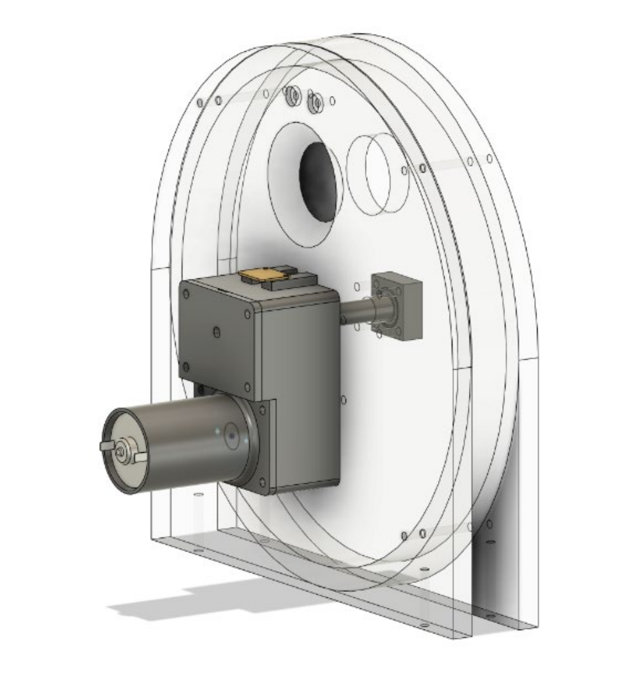




Beam Chopper - Safeties

- Key is to be safe and reliable for facilities to approve use (TAMU, LBNL, …)
  - Operates at end of beamline while beamline safeties disabled
  - Designed for passive safeties (against failure of spinning disc):
    - Large enclosure to contain debris
    - Small output in direction of beamline, with opening toward beamline flared out to allow highest chance debris doesn’t head toward beamline
    - 1 or 2 additional layers of beamline endcap material (depending on facility request and/or beam range)
  - Utilizes 3+ active safeties: (endurance testing will likely tweak these settings/capabilities)
    - Current monitoring window
    - System temperature monitoring
    - RPM measurements (still being refined)
    - Vibration monitoring

- Due to funding priority, this is tabled at “ready to test in lab”
  - Next steps: run at JPL for 24-72 hours solid,
    then run start/stop cycling for a few 100 iterations
  - Send to TAMU to run similar tests in TAMU lab
  - Establish degrading component refresh/stock plan



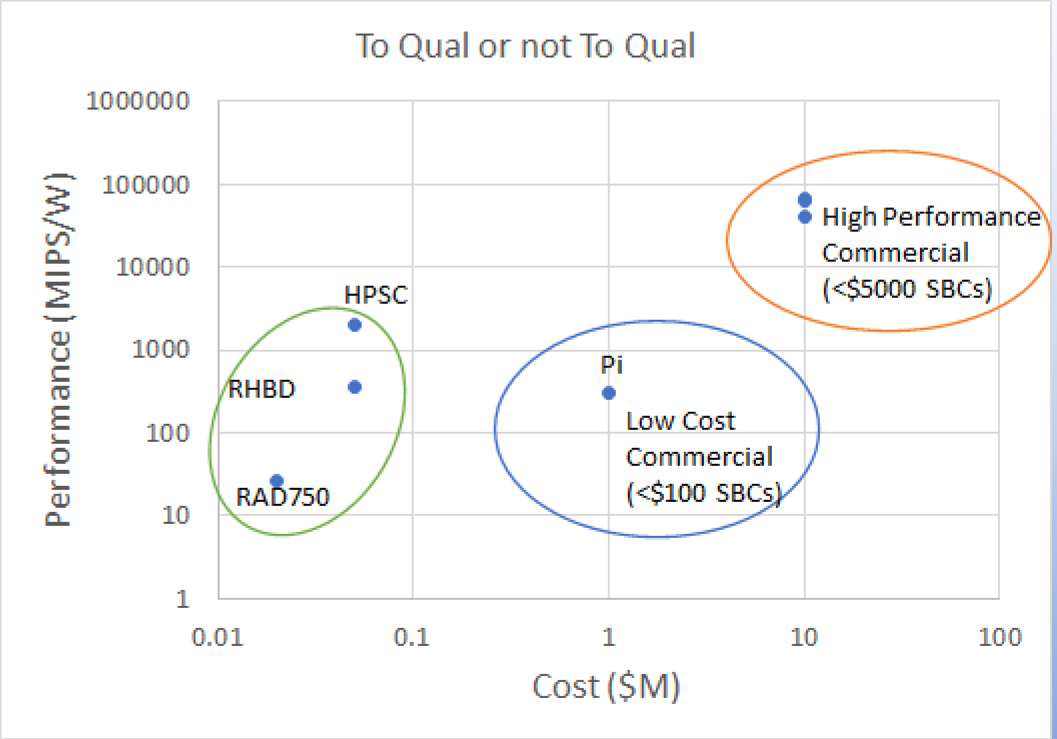
Processor Enclave Biweekly Call

- Please let us know if you are interested in participating
  - Looking for other doing testing of commercial devices and next-generation RHBD devices
  - Primary goal of the call is to try to minimize overlap and maximize testing and effectiveness of testing within NASA and participating government programs
  - Assistance is helpful from: testers, manufacturers (including ARM, RISC V, etc.), and applications designers (what do you guys really need?)
  - Or if you have a program and are looking for data or are interested in helping shape upcoming testing
- Next call is 6/19/2020
  - Planning to review ETW presentations with anyone who wants to call in (contact Ed Wyrwas, Steve Guertin, Jonny Pellish, or your favorite NEPP person)

END

Processor Efforts/Enclave

- In the last 10-15 years there has been a big push to put more COTS hardware in space
  - Achieving normal (flagship) Space Qual ~$10M-$100M
  - On lower-end architectures (Raspberry Pi), this is cheaper, but not really necessary
  - Low criticality (including ISS!), but also non-NASA
- You simply can't get 10000+ MIPS/W in RHBD
  - Can you make your application live in the RHBD range?
  - Do you sacrifice reliability and minimize qual cost?
  - Do you spend a lot of money on a qual that may fail?
- This trade off is not viable for most space users
  - Only option is minimal qual
  - Processor Enclave is seeking to help find reduced qual approaches to achieve some level of reliability
- The good news – 10x-1000x more processing…



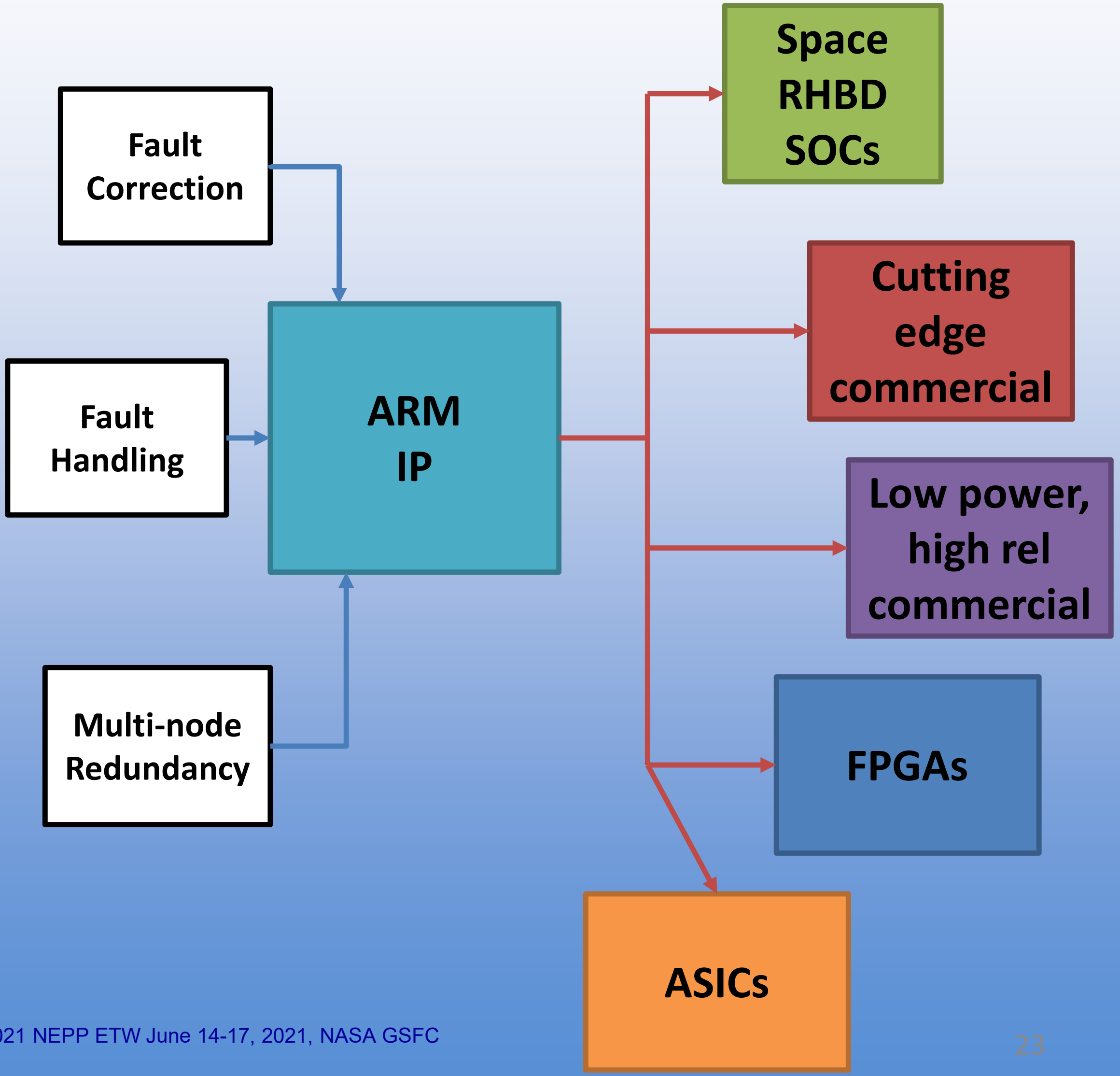
Assumes a "good qual", not just a "quick & dirty" exposure of a few flight lot boards. (this is literally a night & day difference)

Devices of Interest - ARM

- Drone processors
  – like Snapdragon 801
  on Mars Helicopter
- Cell phone processors – like
  Snapdragon 820-855, 865 (5nm!)
- Microchip/Atmel SAMA5D3
  – (ARM A5 devices – collaboration…)
- Xilinx Ultrascale+ MPSoC, similar
- TI TMS570 or similar – for fault tolerant ARM architecture
- (Broadcom, Huawei, Rockchips – esp. RK1808 with neural CPUs…)

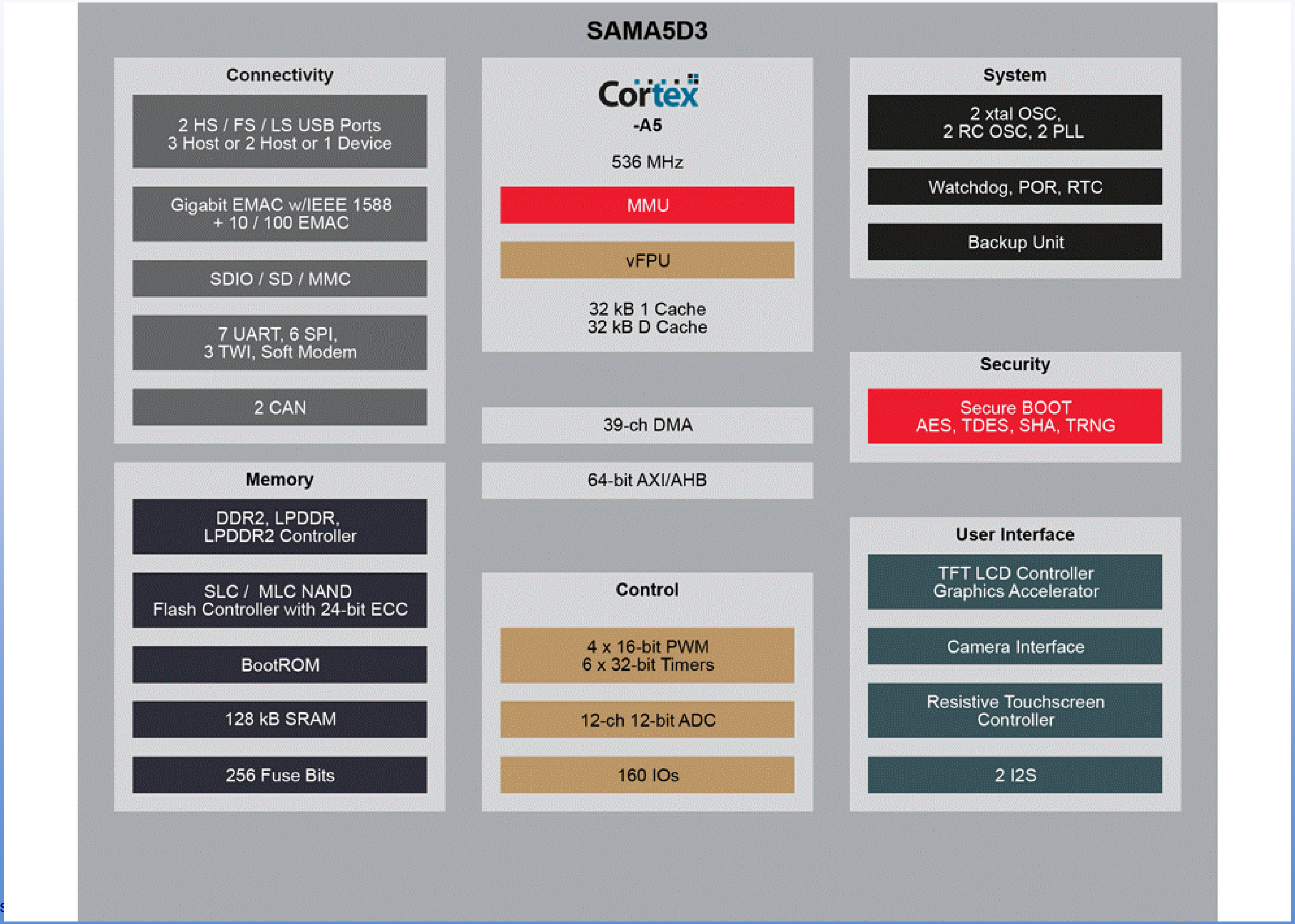
ARM

- Modern space processors are dependent on commercial IP.
- IP-only – puts ARM in a unique position…
  - They don't make hardware.
  - They need to develop and verify fault tolerance works correctly.
  - They need to help licensees properly implement hardware and software.
- Increasing understanding that 10+ heterogenous processor SOCs will stop much more often than they give a wrong answer.
- We can engage ARM to understand:
  - Fault modes; correction methods
  - Fault handling
  - Hardware configuration and handling
    - TMR, DMR, on-chip clusters, proper enabling
  - Software impact



SAMA5D3 - Example

- A5-Based Microcontroller
- Xilinx UltraScale+ MPSoC has dual-core R5s for comparison
  - (And comparison to on-chip quad-core A53)
- Also working on getting A5-IP via collaboration



SAMA5D3

Connectivity

2 HS / FS / LS USB Ports
3 Host or 2 Host or 1 Device

Gigabit EMAC w/IEEE 1588
+ 10 / 100 EMAC

SDIO / SD / MMC

7 UART, 6 SPI,
3 TWI, Soft Modem

2 CAN

Memory

DDR2, LPDDR,
LPDDR2 Controller

SLC / MLC NAND
Flash Controller with 24-bit ECC

BootROM

128 kB SRAM

256 Fuse Bits

Cortex
-A5

536 MHz

MMU

vFPU

32 kB 1 Cache
32 kB D Cache

39-ch DMA

64-bit AXI/AHB

Control

4 x 16-bit PWM
6 x 32-bit Timers

12-ch 12-bit ADC

160 IOs

System

2 xtal OSC,
2 RC OSC, 2 PLL

Watchdog, POR, RTC

Backup Unit

Security

Secure BOOT
AES, TDES, SHA, TRNG

User Interface

TFT LCD Controller
Graphics Accelerator

Camera Interface

Resistive Touchscreen
Controller

2 I2S