National Aeronautics
and Space Administration

NEPP Electronic Technology Workshop
June 17-19, 2014

SOC SEE (Radiation) Guideline Update

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)

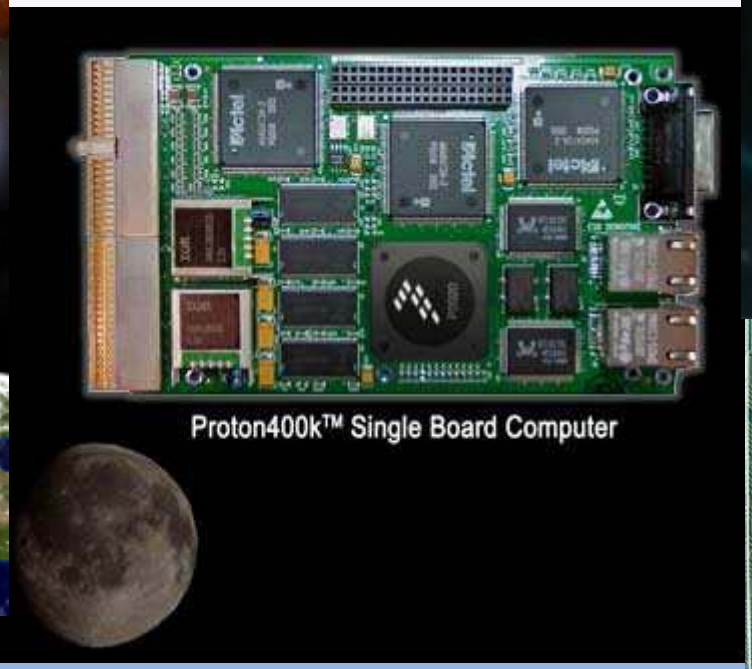
Copyright 2014 California Institute of Technology. Government Sponsorship is acknowledged.

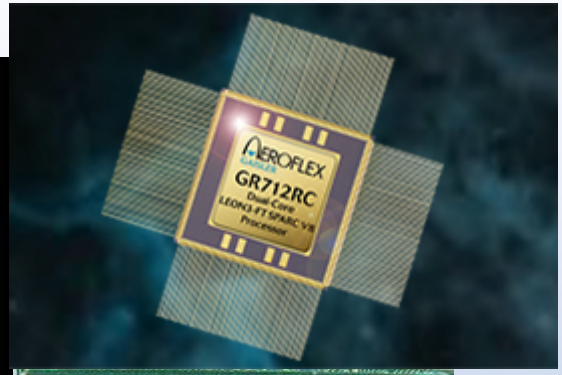
Outline

- SOC SEE Guideline
  - What it is & why
  - Selected items from guideline
    - General structure
    - Test development
    - Test algorithms
    - Testing information and recommendations
- Schedule and Review Effort
- Conclusion

- Discussion of Atom and ARM Efforts

National Aeronautics
and Space Administration





SOC SEE (RADIATION) GUIDELINE

Motivation

- Many space programs are using or designing around modern SOCs – i.e. Freescale P2020
- Space part manufacturers are providing RHBD SOCs – i.e. Aeroflex UT699, BAE RAD5545/RADSPEED microprocessors
- JPL microprocessor SEE guideline needs update for modern SOC devices – and to address NEPP concerns.
  - Very complex devices
  - Multiple elements – do all need to be tested?
  - Manufacturing processes impact on test methods
- Timeline
  - Methodology development ’09-’12
  - Guideline writing ’11/’12
  - External and internal review ’12/’13, Export review ’13/’14

Structure of SOC SEE Guideline

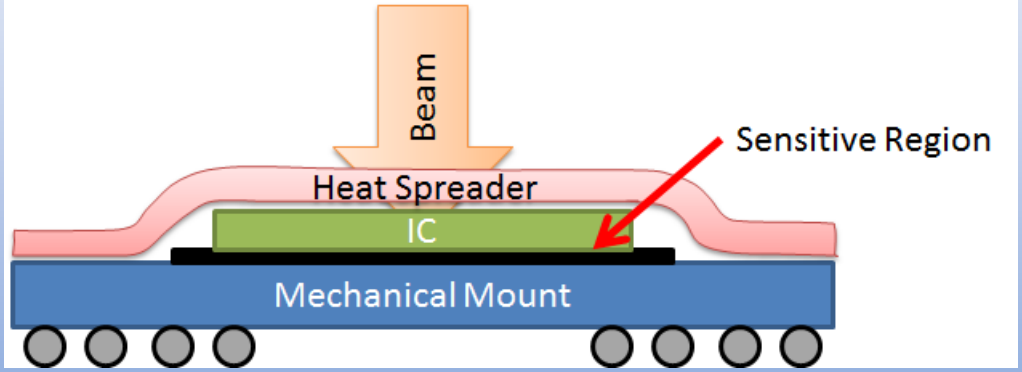
- Covers key areas of test planning, development, and performance:
  - Determining the type of radiation testing needed or possible
  - Development of hardware, software, and test procedures
  - Performance of testing and analysis of data
- Provides relevant examples
  - Freescale P2020 & P5020, Boeing Maestro ITC, Aeroflex UT699
- Provides specific recommendations for various elements of SEE testing of SOCs.

Guideline Coverage

- Details to Consider:
  - Mission environment – see standard guidelines and mission specifications – generally may need proton and heavy ion data
  - SOC details (e.g. RHBD construction)
  - Program usage of the SOC
  - Tool that will be used for rate calculations
- SOC Details
  - Mechanical and thermal information – may impose range requirement
  - Trade study between custom hardware and inexpensive evaluation boards is recommended
  - Modern device feature size may warrant proton direct ionization testing
- Program usage
  - General testing may not be explicit enough for the user
  - But user must provide actual flight usage (unlikely till after launch)

Test Preparation

- Determination of appropriate test facility
  - Summarized: IUCF, UCD, Triumpf, TAMU, BNL, UCB, UCL, RADEF, NSRL
- Establish package materials and determine depackaging…



- Selection of DUT board – custom vs. inexpensive manufacturer evaluation board?
  - 10's of k vs. 0.5-4k
  - Note that lower-end evaluation boards may be sufficient (~0.5k)
  - Might be able to rework mfr board with socket

What and How to Test

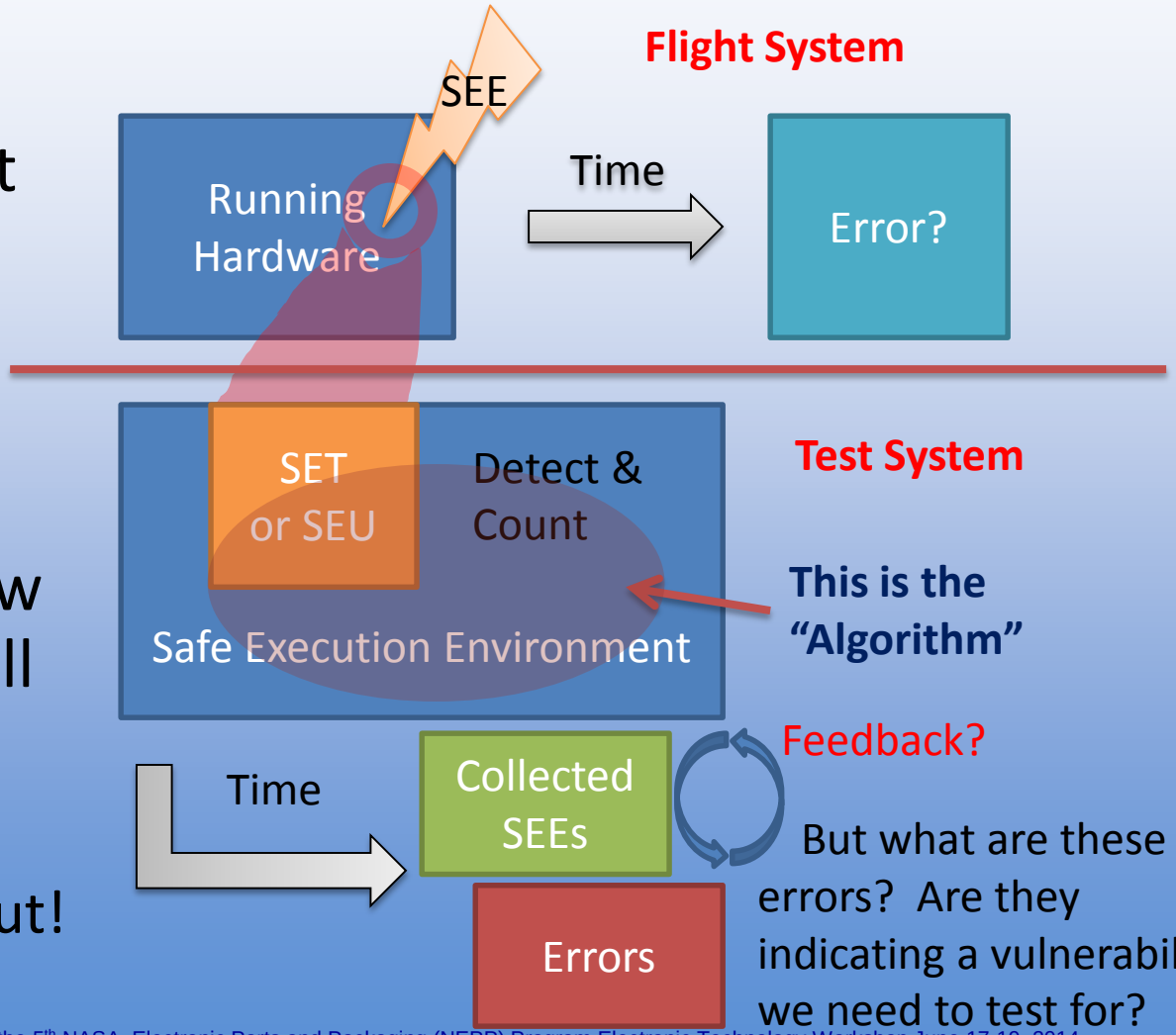
- In a perfect world, you test everything. In reality you have to pick things and build detection.
- The guideline focuses on testing basic structures and systems
  - SRAM-based targets – such as caches and some types of registers
  - Flip-Flop-based targets – generally the other registers and execution unit pipelines (if possible)
  - Exception/Trap/Interrupt systems (especially catching unexpected events)
  - Data flow through memory controllers, communications systems, etc.
- Testing complex devices with multiple subsystems and fault handling is not deterministic.
  - Cycle-by-Cycle comparison to expectation is problematic
  - Simple targets can be easily tested with simple hardware
  - Most everything else seems to require in-situ detection or using debugging hardware to repeat tests during exposure
- Performance benchmarks and manufacturer's hardware test codes must be modified for SEE testing (poor target control and duty cycle)

Keys to Test Algorithms - Structure

- Manufacturer equipment can perform SEE detection
  - Relatively easy to use
  - Limited ability to exercise the DUT
  - Difficult to automate
- Custom Algorithms
  - Recommend Assembly Language
    - Does not hide machine behavior or make assumptions about programming model (for example, C assumes a subroutine call structure and variable storage model)
    - Can make it difficult to debug, and may limit complexity of test algorithms

Anatomy of an Algorithm

- Figure out what sources you want to detect
- Construct a safe environment for SEEs to occur
- Two ways to know what the SEEs will look like
  - Anticipation
  - Test and Find Out!



But what are these errors? Are they indicating a vulnerability we need to test for?

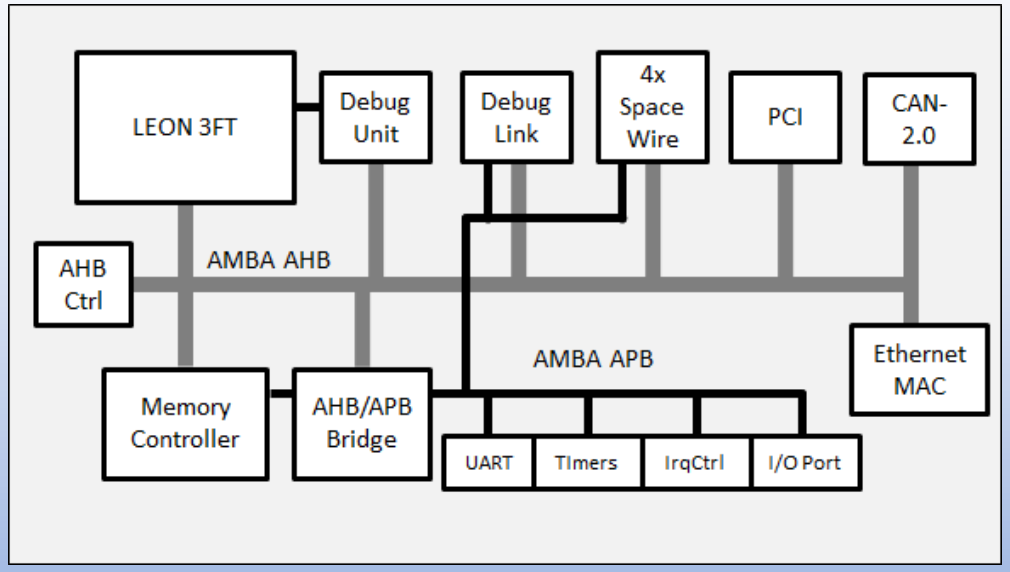
Keys to Test Algorithms - Anomalies

- It is common to want to explore test anomalies because they may be “rare SEEs”. Avoid this want as it leads to much lost time.
- Rely on beam and event statistics. If it doesn’t repeat, put a limit on it and move on. If it does repeat, figure out what it is if possible.
  - If you can’t figure out what it is, that’s ok. Report the rate, but don’t claim a mechanism.
  - Be aware you may be exploring a bug in your code, or even upsets in support or test equipment.
- When possible, use debugging tools to examine these as they enable fast and detailed examination

Testing – Determine Sensitivity

- Because of the nature of SOC testing many different types of events can influence reported data.
- This necessitates determining the sensitivity of the test system for each event type investigated.
  - Determine the system response that establishes a “floor” to detection – for example if the DUT “crashes” (fails to continue proper execution) this generally means that any more rare event is undetectable.
- Sensitivity can be improved by either improving robustness of the test system or by targeting a given event type.

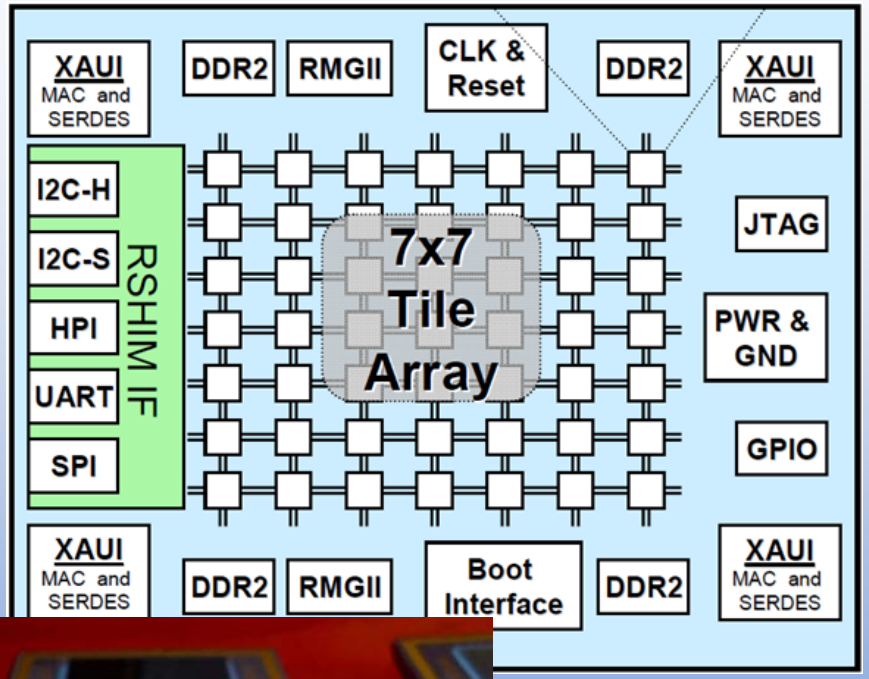
Example from UT699 - 1



- Sparc V8 Leon 3FT core – desired by many space users
- RHBD with built-in FT
- Many different types of components
- Study includes “register partial reset” and Spacewire examination

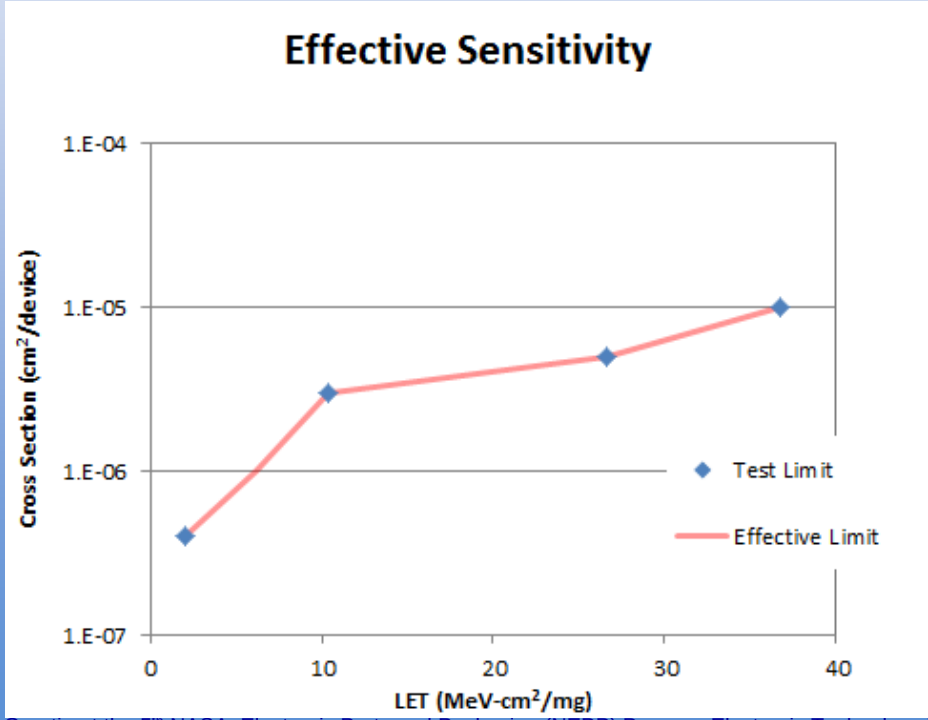
Example from Boeing Maestro ITC

- 49-core tiled microprocessor
- RHBD and FT construction with SRAMs expected to upset readily (but they are EDAC protected)
- Highlights need for custom DUT preparation.



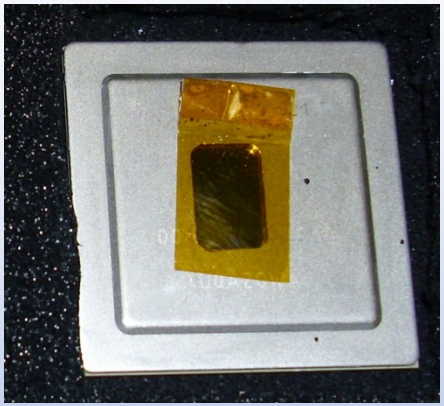
Test Efforts/Results

- Highlight of need to understand beam delivery and algorithm structure (algorithm behavior on multiple tiles potentially confusing).
- Effective sensitivity is a key player because the caches are very sensitive to SEEs and relatively high fluence is required to activate non-cache upsets. But in the meantime the caches are experiencing many SEUs.

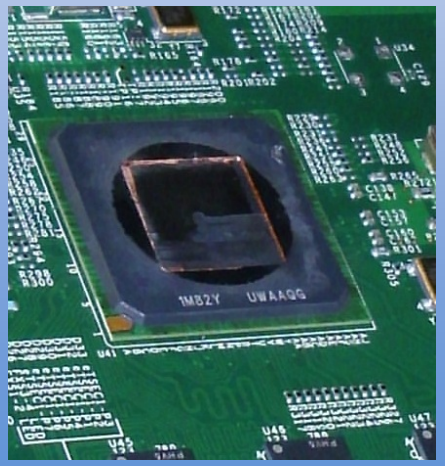


Example from Freescale - 3

- The P2020 and P5020 processors provided the best example of commercial/COTS type devices with the relevant testing issues.
  - Thermal issues and depackaging dominate
  - Cannot reliably test at many angles
- Debugging tools are well-developed
  - Codewarrior tool set is useful for directly observing what is wrong with the processor (provided it can successfully perform).



P5020 – with hole to expose e5500 cores



P2020 with hole in heat spreader

Review & Schedule

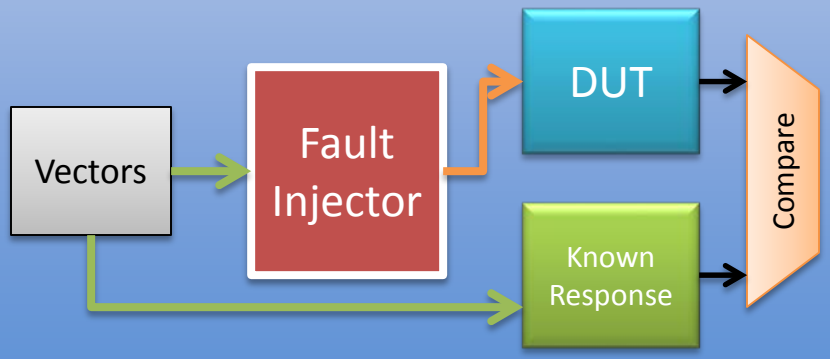
- Initial Reviews: 01/2013-06/2013
  - Heather Quinn, LANL
  - Craig Hafer, Aeroflex
  - Rocky Koga, Aerospace
- Major Update: 07/2013-09/2013
- JPL Release Effort: 10/2013-03/2014
  - ITAR review took considerable time – some references were changed/improved to support
- Current Review Effort:
  - Plan is to release document as NASA document
  - Requesting review responses as soon as possible (original target was mid-June 2014)
  - Final modifications will be finished Sep. (release may be longer)

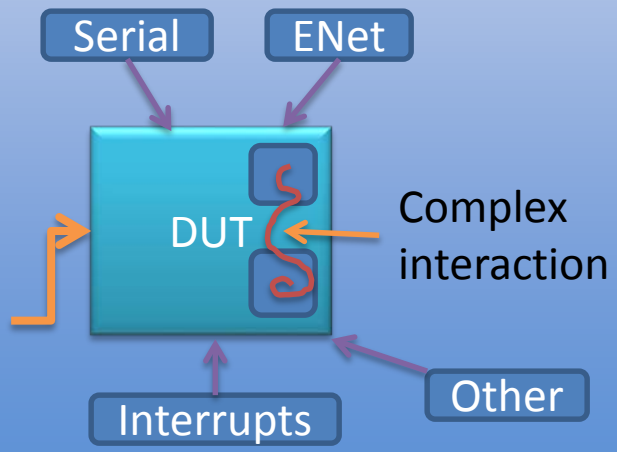
Feedback Topics

- Clarification: Scope of SOC
- What it is: not qualification
- General structure:
  - Executive primer early on, reduce redundancy
- Clarification of sensitivity and sensitizing…
- Prediction of software response
- More detail in translation of existing SEE characterization methods
- Clarification of test conditions – desired vs. limitations imposed by test systems and complexity
- Interplay of TID and SEE (is this already in there?) – improved coverage
- Rad performance at EOL
- Application specific, or TLYF approach
- Clarify position of FPGAs in this guideline – likely will point people to Melanie's FPGA guideline

Software/Application Response

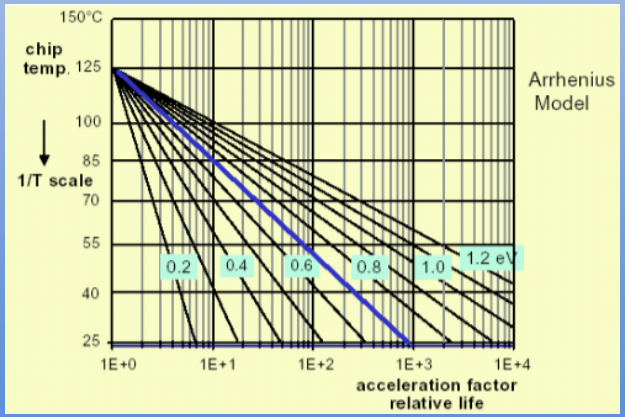
- How do you take the information generated and apply it to a system?
- Three approaches – based on review feedback:
  - Bounding approach – assume a percentage of upsets affect your system: This is the method primarily recommended in the guideline
  - Simulate the system and perform fault injection in-line with SEE behavior – no good recommendation here because of complexity
  - Put flight-like application in the beam to get a general idea what's going on (Test Like You Fly)… adding as useful recommendation for system-level verification





SEE after TID and EoL

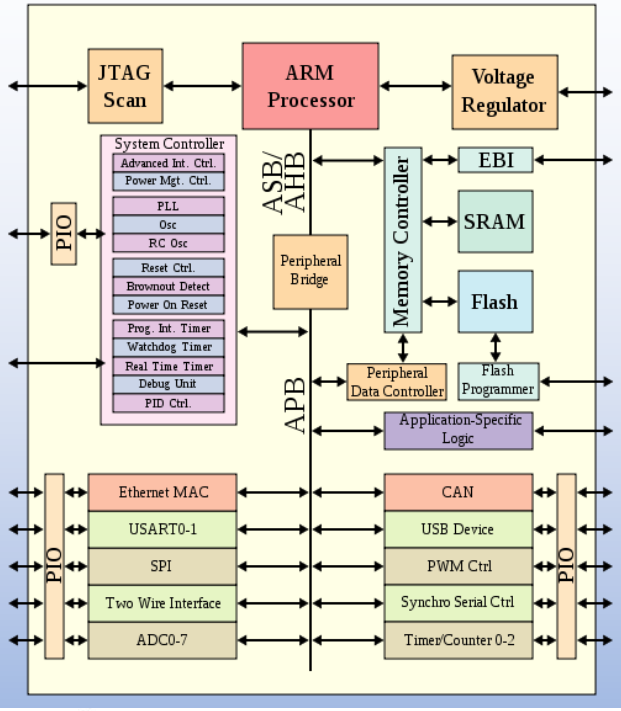
- Will handle this with the Sandia SEE guideline approach for TID
  - Prepare devices with TID exposure – appropriate for the mission
  - Compare with and without TID exposure, if a difference is observed, continue with TID-exposed devices for mission data
  - We don't expect to see much here because TID limits are usually high relative to SOC capability
- End of Life recommendations
  - This topic is probably out of scope for depth in the guideline, but we will recommend life test before exposure on some parts to compare



From Bayle, 2010

SOC vs SOC

- System on a Chip traditionally refers to single chips that include almost all of a computer or DSP circuit
- Modern use, however, is commonly applied to devices where power, memory, and hardware interface elements are exterior to the SOC
  - This appears to be a case of the definition changing, as manufacturers and community are moving to this definition
- For purposes of the audience of the guideline, we will be clear on this distinction, though we may not change titles and headings



Intel CE3100

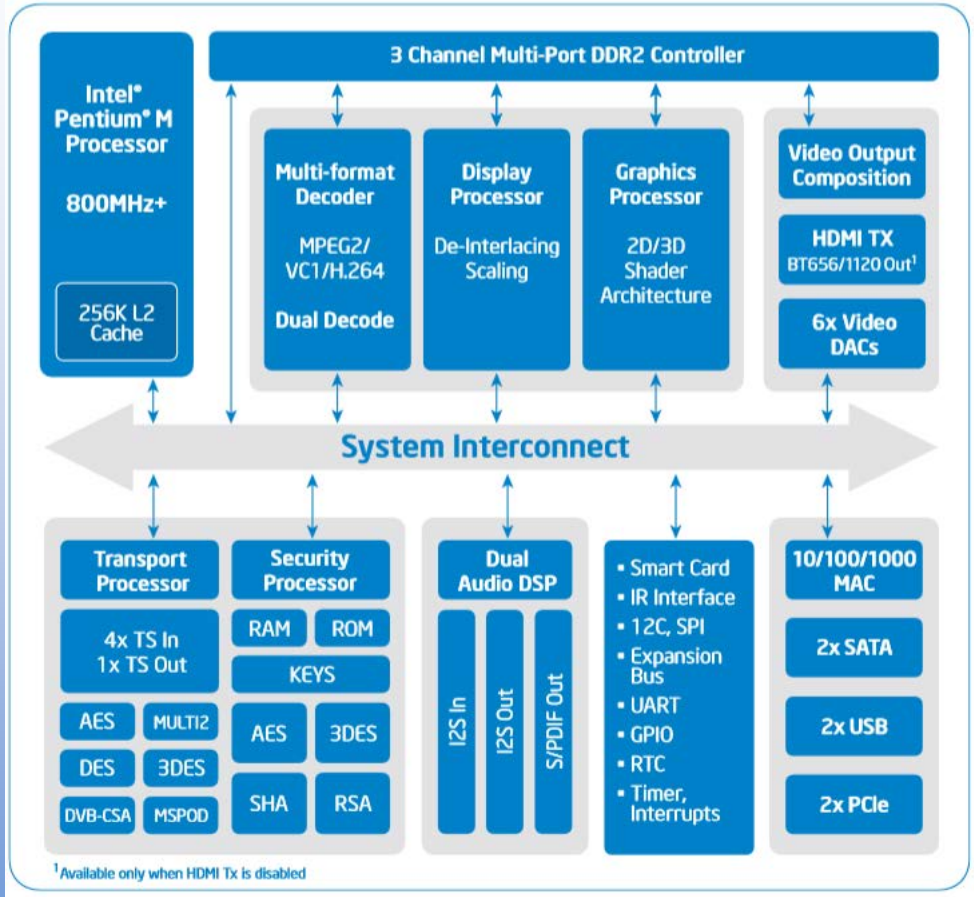


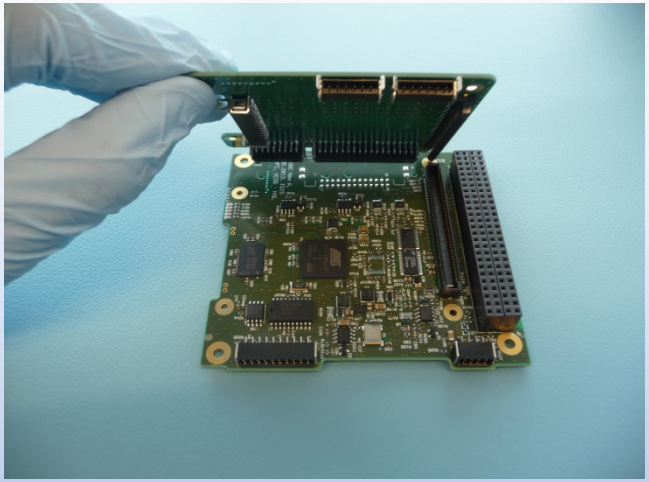
Figure 1 - Intel® Media Processor CE 3100 - Block Diagram

The Intel® Media Processor CE 3100 is a highly integrated system-on-a-chip that combines a high-performance Intel® architecture processor core with leading-edge video decoding and

Conclusion

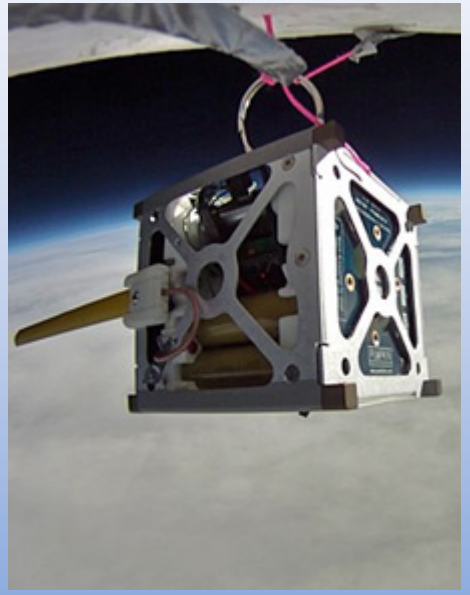
- SEE Guideline completed (pending final changes)
  - Provides detailed information on test planning, hardware and software development, test operations, and sample test review to discuss key concepts
- Review and modification process approaching completion
- Some input from final set of review may require minor modification, but majority of guideline already cleared through JPL release
- NEPP SOC SEE Guideline Document expected to be released before EOY 2014 (name may change slightly)

National Aeronautics
and Space Administration



ARM9-based ISIS board

NASA AMES Phonesat



MOBILE MICROPROCESSORS

(CROSSOVER TOPIC WITH CUBESATS)

To be presented by Steven M. Guertin at the 5th NASA Electronic Parts and Packaging (NEPP) Program Electronic Technology Workshop June 17-19, 2014, NASA GSFC, Greenbelt, MD.

A Packaging Example

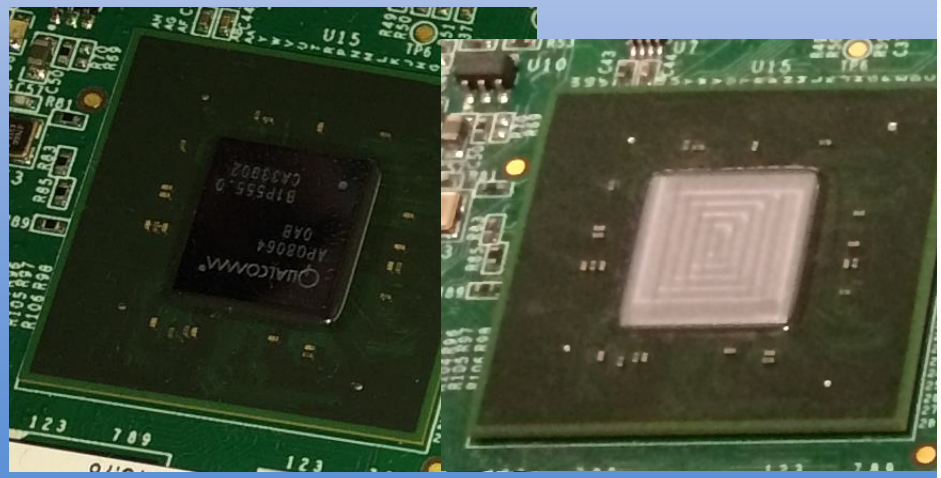
- We are finding some significant problems with some test boards, in terms of preparation for test…

A Packaging Example

- We are finding some significant problems with some test boards, in terms of preparation for test…

Alternate Snapdragon Board

- Another board design – IFC6410 is much easier to work with
- We have established method to thin devices to enable testing with 25 MeV/amu cocktail at TAMU (~100 μm)



SEE Testing of Snapdragon

- Exposed IFC6410 board to Ar @ LET = 7 MeV-cm^2/mg
- Test software was to monitor Android boot
  - Observed through UART error output
  - Provides information about boot behavior for first ~90 seconds
  - Takes ~5 seconds after power up to be activated
  - Provides interrupt/exception reporting
- Total exposure was 3e6 #/cm^2
  - No damage observed
  - No evidence of high current modes (but LET was low)
  - But system had a hard time, and crashed often

National Aeronautics
and Space Administration

L2 Error

- Capture of errors during boot
- LET=6.3, flux ~5e3 #/cm^2
- Using Android boot

Unspecified Exception

- Requires ~10-15 secs of boot to get reports

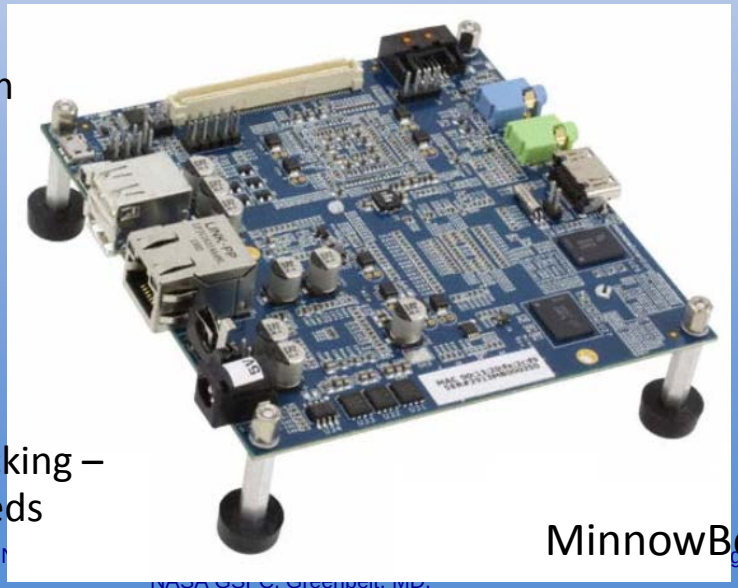
Unspecified Exception

- Usually fails to reach reportable status

```
[    42.074530] Bluetooth: Tx timer expired
[    42.074530]
[    43.814375] L2 Error detected!
[    43.816419]  L2ESR     = 0x00010010
[    43.819807]  L2ESYNR0  = 0x6200080a
[    43.823195]  L2ESYNR1  = 0x04a4977a
[    43.826583]  L2EAR0    = 0x0002c160
[    43.829971]  L2EAR1    = 0x00000000
[    43.833358]  CPU bitmap = 0x1
[    43.836319] L2 data soft error, single-bit
[    43.840378] Kernel panic - not syncing: L2 single-bit error detected
[    43.846787] [<c0014918>] (unwind_backtrace+0x0/0x11c) from [<c07ba978>] (pani
c+0x84/0x1d4)
[    43.855028] [<c07ba978>] (panic+0x84/0x1d4) from [<c007d7f4>] (msm_l2_erp_irq
+0x1fc/0x260)
[    43.863268] [<c007d7f4>] (msm_l2_erp_irq+0x1fc/0x260) from [<c00d2844>] (hand
le_irq_event_percpu+0xb0/0x290)
[    43.873065] [<c00d2844>] (handle_irq_event_percpu+0xb0/0x290) from [<c00d2a60
>] (handle_irq_event+0x3c/0x5c)
[    43.882862] [<c00d2a60>] (handle_irq_event+0x3c/0x5c) from [<c00d554c>] (hand
le_fasteoi_irq+0xdc/0x148)
[    43.892232] [<c00d554c>] (handle_fasteoi_irq+0xdc/0x148) from [<c00d20a8>] (g
eneric_handle_irq+0x30/0x44)
[    43.901785] [<c00d20a8>] (generic_handle_irq+0x30/0x44) from [<c000ee40>] (ha
ndle_IRQ+0x7c/0xc0)
[    43.910544] [<c000ee40>] (handle_IRQ+0x7c/0xc0) from [<c0008620>] (gic_handle
_irq+0x94/0x110)
[    43.919059] [<c0008620>] (gic_handle_irq+0x94/0x110) from [<c07c8cc0>] (__irq
_svc+0x40/0x70)
[    43.927453] Exception stack(0xc0d05e70 to 0xc0d05eb8)
[    43.932488] 5e60:                                     00000000 c0f3cd00 00000
000 00000000
[    43.940668] 5e80: c0d04000 00000000 00000202 c0d05f28 00000012 fa00300c c0d43
75c 0000000a
[    43.948817] 5ea0: c0d4f080 c0d05eb8 c0087d3c c0087710 20000113 ffffffff
[    43.955440] [<c07c8cc0>] (__irq_svc+0x40/0x70) from [<c0087710>] (__do_softir
q+0x4c/0x248)
[    43.963680] [<c0087710>] (__do_softirq+0x4c/0x248) from [<c0087d3c>] (irq_exi
t+0x48/0xa0)
[    43.971860] [<c0087d3c>] (irq_exit+0x48/0xa0) from [<c000ee44>] (handle_IRQ+0
x80/0xc0)
[    43.979734] [<c000ee44>] (handle_IRQ+0x80/0xc0) from [<c0008620>] (gic_handle
_irq+0x94/0x110)
[    43.988249] [<c0008620>] (gic_handle_irq+0x94/0x110) from [<c07c8cc0>] (__irq
_svc+0x40/0x70)
[    43.996642] Exception stack(0xc0d05f28 to 0xc0d05f70)
[    44.001678] 5f20:                   00000000 00000000 00000001 00000003 00000
003 00000003
[    44.009858] 5f40: c0d65924 00000003 00000000 c0d65924 00000000 00000000 00000
002 c0d05f70
[    44.018007] 5f60: c07c873c c005bf7c 60000013 ffffffff
[    44.023073] [<c07c8cc0>] (__irq_svc+0x40/0x70) from [<c005bf7c>] (msm_cpuidle
_enter+0x70/0x78)
[    44.031680] [<c005bf7c>] (msm_cpuidle_enter+0x70/0x78) from [<c056911c>] (cpu
idle_enter+0x14/0x18)
[    44.040592] [<c056911c>] (cpuidle_enter+0x14/0x18) from [<c0569690>] (cpuidle
_idle_call+0x1e0/0x3c0)
```

Atom Test Devices

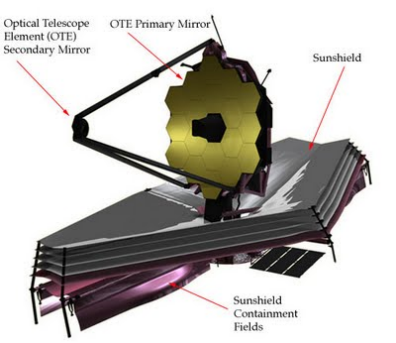
- Two types of boards:
  - MinnowBoard
  - Conga board
- Conga boards
  - Use E620 embedded Atom processors
  - 45nm parts
- MinnowBoards
  - Use E640 embedded Atom processors
  - 45nm parts
- Hobbyist Boards
  - Strong community for hacking – will be helpful for test needs

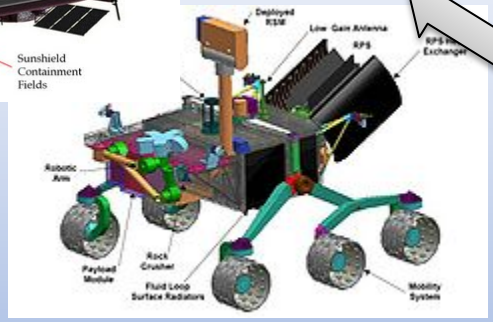


MinnowBoard

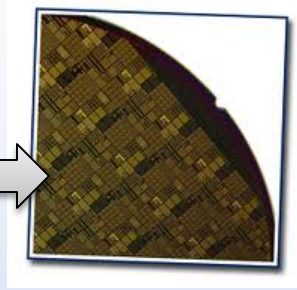
BACKUP

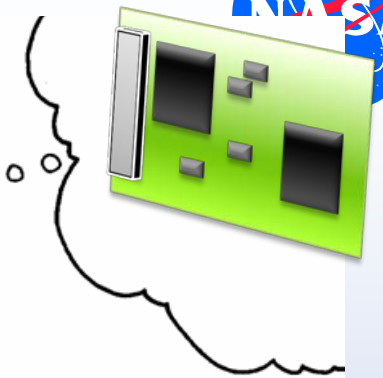
NEPP Approach





Users





Manufacturers

Testers

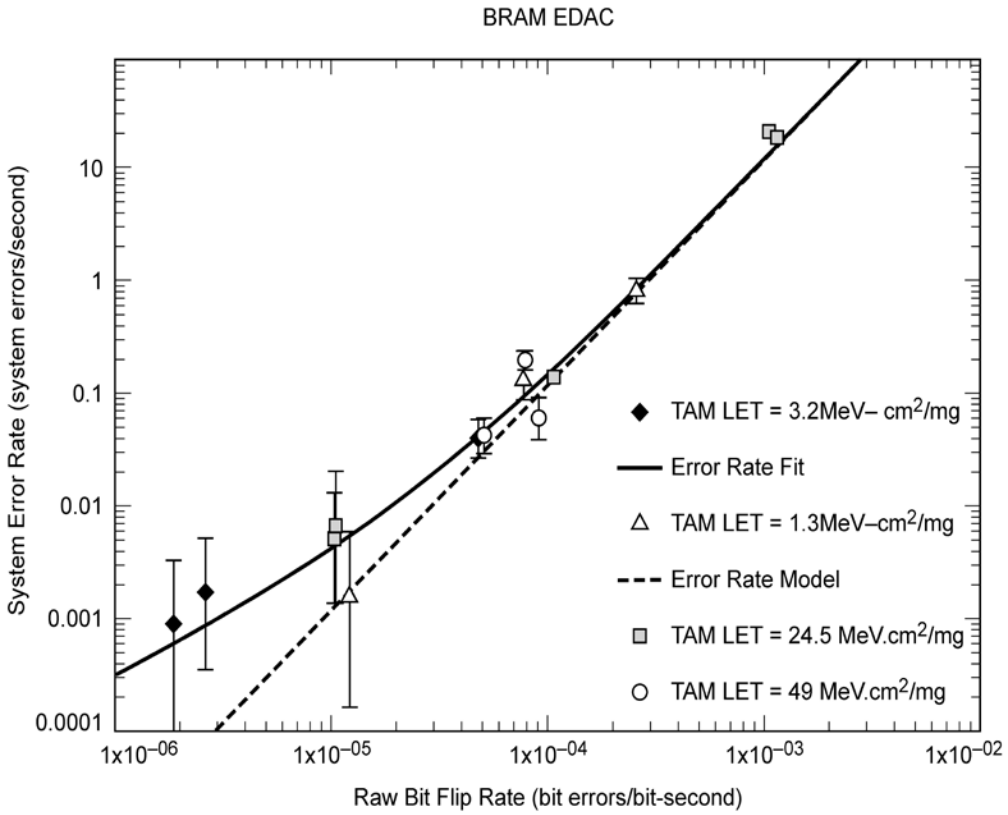
- Devices are too complex for full characterization
- Test methods must target basic information needed for manufacturers and provide useful information for users to be aware of and/or mitigate radiation effects
- This approach seeks to provide the information necessary to understand the most important radiation effects for a given system, utilizing manufacturer assistance, and targeting actual application needs.

Focus Areas for Guideline

- Collaboration with Manufacturers and Users
- On-Chip Peripheral Approach/Prioritization
- Fault Tolerant Device Test Approaches
- RHBD Device Challenges to Test Development
- Multicore Device Unique Challenges
- General Test Methods
- Collecting Results from Sample Testing

Key Data to Collect

- DUT Preparation
  - Hardware modifications
  - Parameters of operation (V, T, I, f)
- # of observed events
  - Description of event
  - Details of code or test equipment that enables detection (machine-description if possible)
  - Details of event – algorithm dependence, throughput, FT/EDAC, etc
- # of incident particles
  - Species, energy/LET, angle of incidence
  - Structure of beam delivery (constant, Poisson, pulsed)



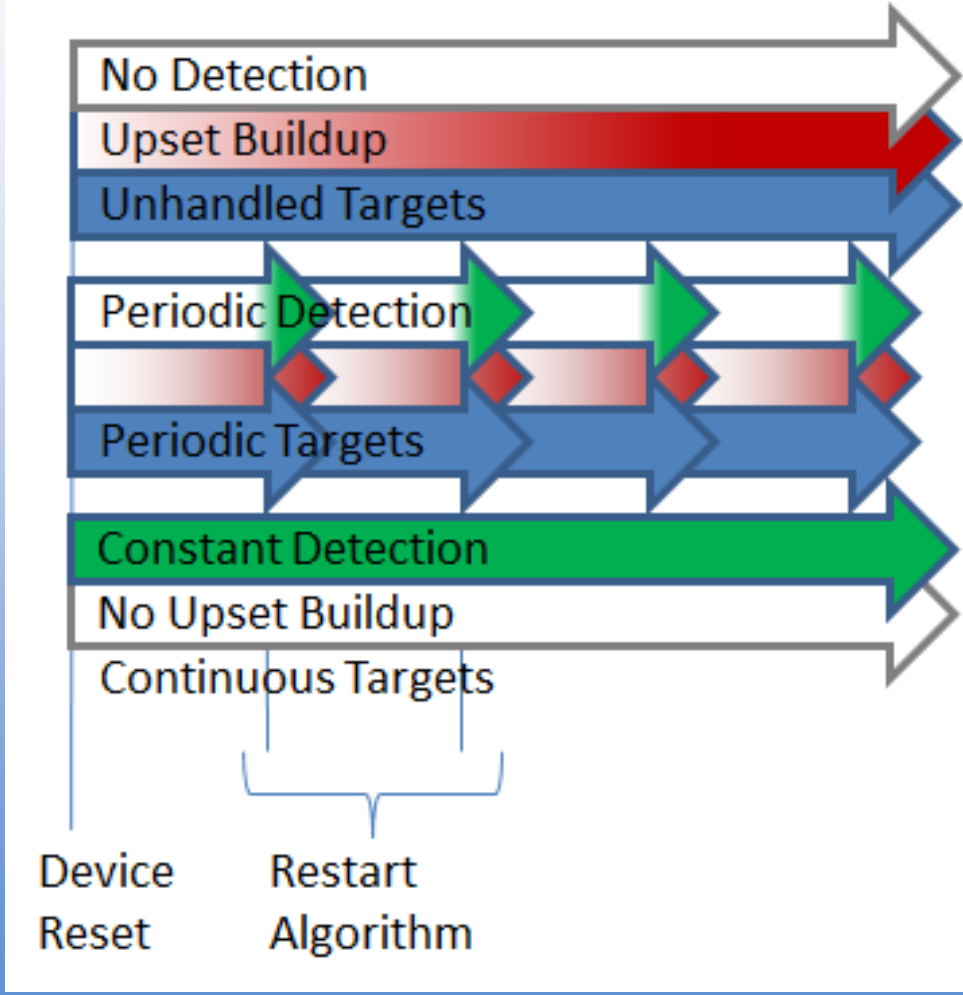
True FT protected element error rate follows dashed line. Real structure may have non-FT component leading to flattening.

Testing – Avoiding Redundancy

- Many modern SOCs are constructed from SRAM or similar elements that are easily tested.
- But these elements are known to be weak to SEE, so they are protected with FT or EDAC. Thus characterization is largely unnecessary.
- It is also important to verify to FT, EDAC or other protection works.
- And it is important to avoid measuring sensitivity of these elements many times (though they can be good for establishing a baseline and test-to-test consistency).

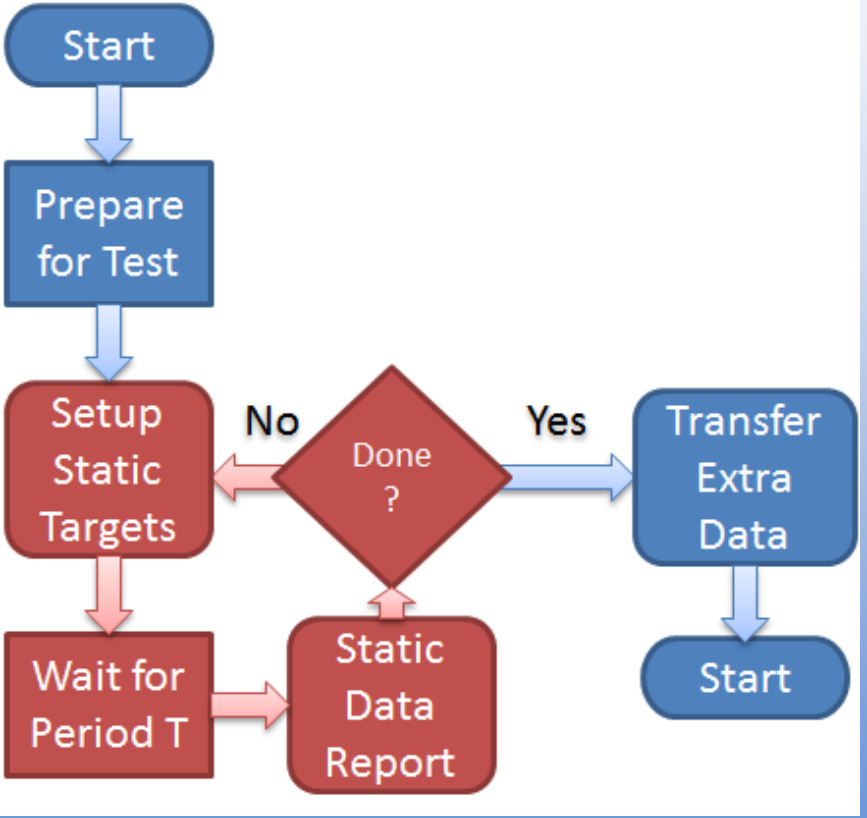
Test Algorithm Time Models

- Unhandled targets build upsets during exposure.
- Periodic targets build upsets during an integration phase
- Constant detection



Standard Static Soak Test

- Test algorithms are generally based on repeated sub-tests that may report on each loop iteration…
- Test algorithms should periodically report results to enable immediate detection of loss of operation.
- Periodic reporting enables use of partial runs.



Preparation for the Test Trip

- Where possible, test SEE detection code by planting simulated SEEs (or using debugging tool to alter things)
- Consider use of laser facilities for fault injection and checkout (but be wary that laser can inject non-beam-type faults)
- Coordinate with test facility regarding requirements for thermal management
- Perform standard DUT preparation and equipment checkout for use at SEE test facilities
  - Long cables (power and/or communication), noise
  - Enable remote power-cycling and manipulation of key buttons (such as power and reset)

Testing – Recording the Test

- When possible, collect the following:
  - Facility test record
  - Test engineer test log
  - Power supply log files
  - All I/O logs between the DUT and support equipment (including key transfers to enable verification of commands)
  - SEE records stored during test and reported after the test completes

Interest in SOCs

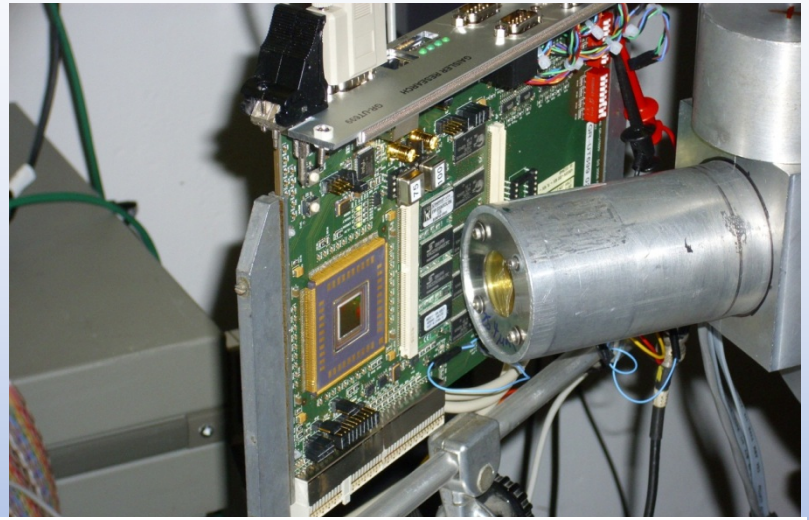
- SOCs gain popularity and will likely fly soon
  - UT699 is a controller on MISSE-7
  - Several programs interested in flying Maestro
  - SpaceMicro building Proton400k-L with Freescale P2020
- Single Event Effects Test methods unclear
  - Very complex devices
  - Multiple elements – do all need to be tested?
  - Hardware simulation of SEEs?
  - Manufacturing processes impact test methods
    - RHBD
    - Fault Tolerance
    - Multicore

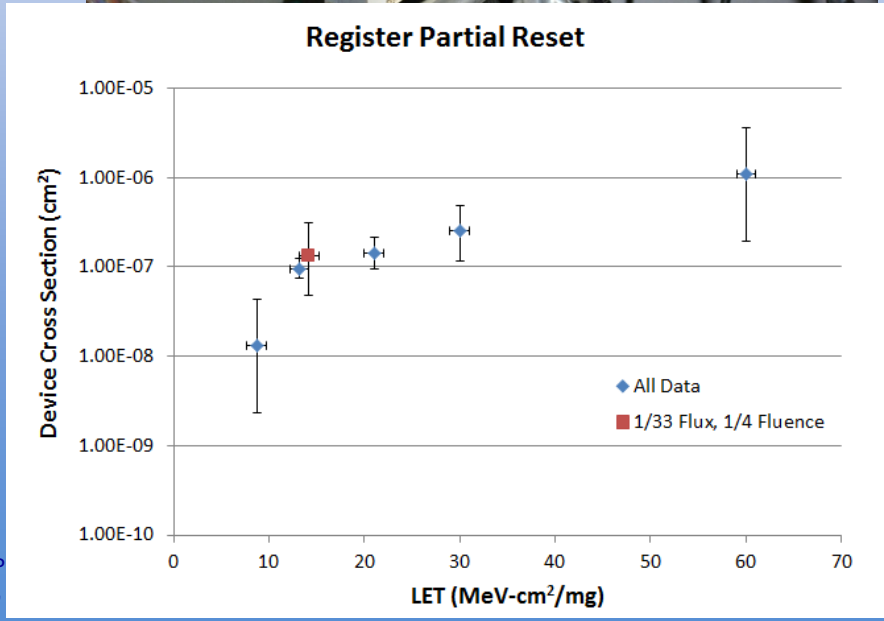
Other Relevant Study

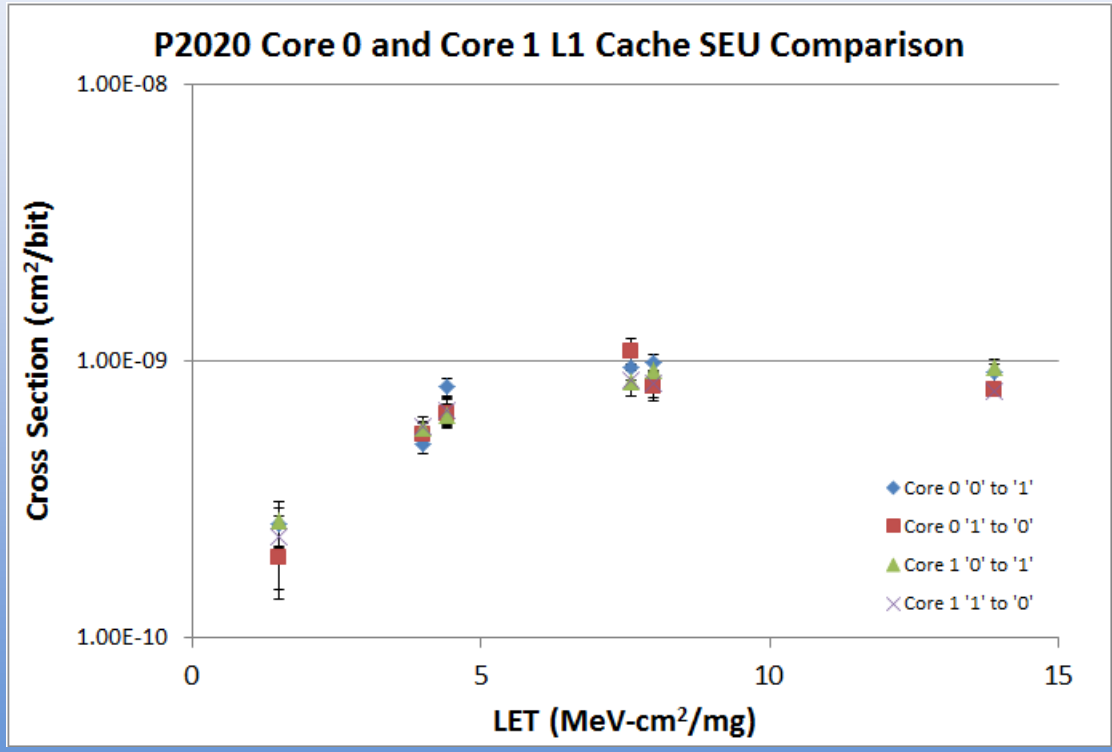
- Frequency Dependence is potentially important and is explored, but recent results indicate the sensitivity is a low priority in commercial devices where frequency is high – mostly because the sensitivity is already very high..
- Cross section vs. Feature Size has stayed within a factor of 10 from 180 to 45nm for Freescale/Motorola devices.
- Voltage is important but largely untestable on modern devices due to lack of control of point-of-load converters on demonstration boards.
- TID enhancement of SEE – it is recommended to be prepared for this, but it is unclear how to give general guidance on this case.
- Angular study of RHBD cells (such as DICE latches) is recommended but it is also pointed out that high angles of incidence may be infeasible.

Testing/Results

- Manufacturer involvement ensured successful DUT preparation and understanding of events
- Explored different types of flux and fluence dependence to lock-in on the register partial reset
- Spacewire results showed effective sensitivity was not sufficient to observe SEE: Spacewire is robust…





P2020 Provides Dual Core Environment…

- Multicore testing will gain in importance…
- P2020 provides cache-coherency for communicating data between processors
- Testing with both cores active shows cache sensitivity is essentially the same.