

70 YEARS OF CREATING TOMORROW



Los Alamos
NATIONAL LABORATORY

Challenges in Testing Complex Systems



Heather Quinn

July 8, 2013

UNCLASSIFIED



Author's Excuse for Not Making the First NSREC



Author's mother in 1963 on Girl Scout trip to Mexico in 1963



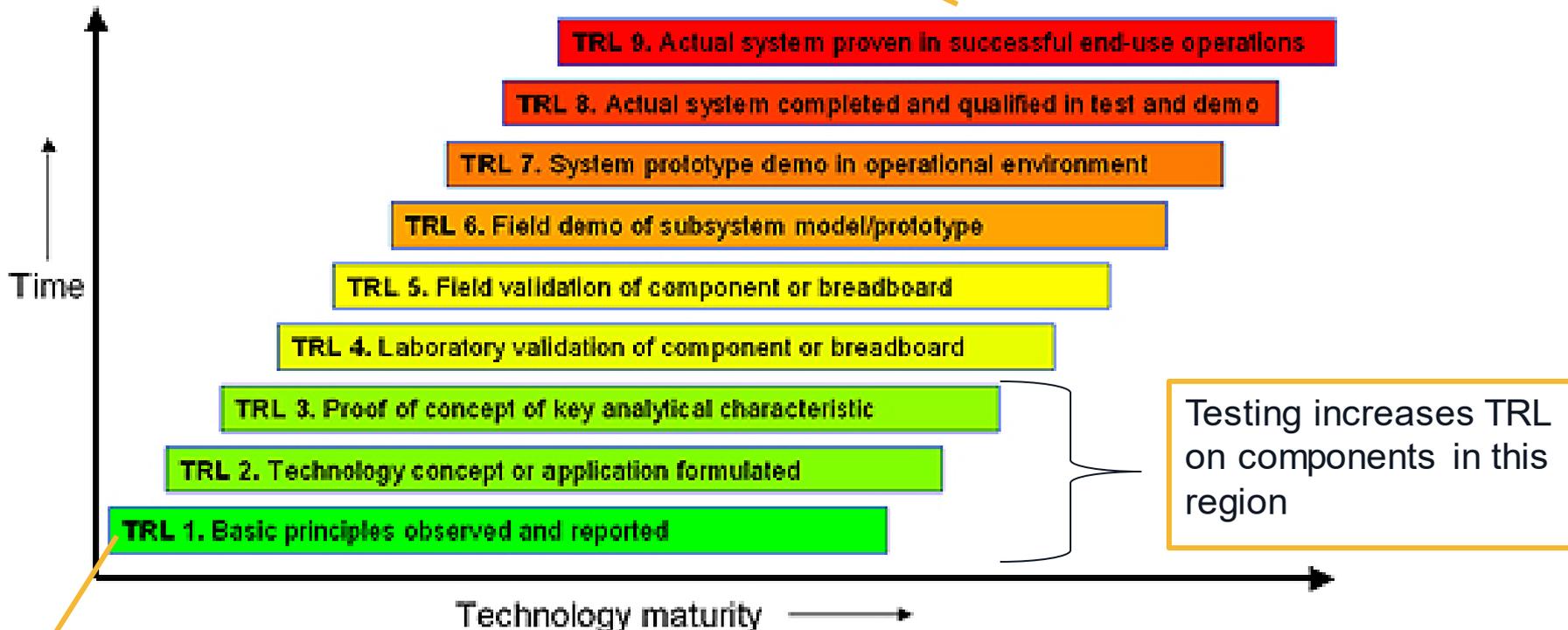
Has Testing Changed?

- The basics of testing have not changed drastically over the years:
 - Much of testing boils down to counting and functional monitoring
- But many other things have changed:
 - We are counting more ions
 - We are doing SEE testing
 - We use (arguably) more complicated test fixtures
 - We are testing a wide range of commercial components
 - We are able to test more full systems



The Reason for Component Testing Has Not Changed

Testing allows components in this region to be applied to different environments



<http://ge.geglobalresearch.com/blog/commercializing-rfid-technology-with-avery-dennison/>

Technology Readiness Level (TRL)



Types of Radiation Testing Have Not Changed

- Total dose testing to ensure the components can withstand the exposure to ionizing radiation typical for the orbit and mission length
- Single-Event Effect (SEE) testing to ensure the components do not have destructive SEEs and that the non-destructive SEEs are tolerable or “fixable”
 - Destructive: SEL, SEGR, SEB
 - Non-Destructive: SET, SEU, SEFI
- Prompt dose and displacement damage (DD) testing which is optional for many missions, except for the case of DD in photodetectors



Test Burden Has Not Changed

Radiation Effect	Components Types
Total Ionizing Dose (TID)	Discretes and Integrated Circuits (ICs)
Enhanced Low Dose Rate Sensitivity (ELDRS)	Bipolar, Bi-CMOS
Single-Event Upset (SEU)	Digital ICs
Single-Event Transient (SET)	Analog ICs, Digital ICs, Photodetectors
Single-Event Functional Interrupt (SEFI)	ICs
Single-Event Latchup (SEL)	ICs
Single-Event Gate Rupture (SEGR)	Power MOSFETs, Flash
Displacement Damage (DD)	Photodetectors

Without collaboration, most of us have too many components with not enough time or budget to completely test.

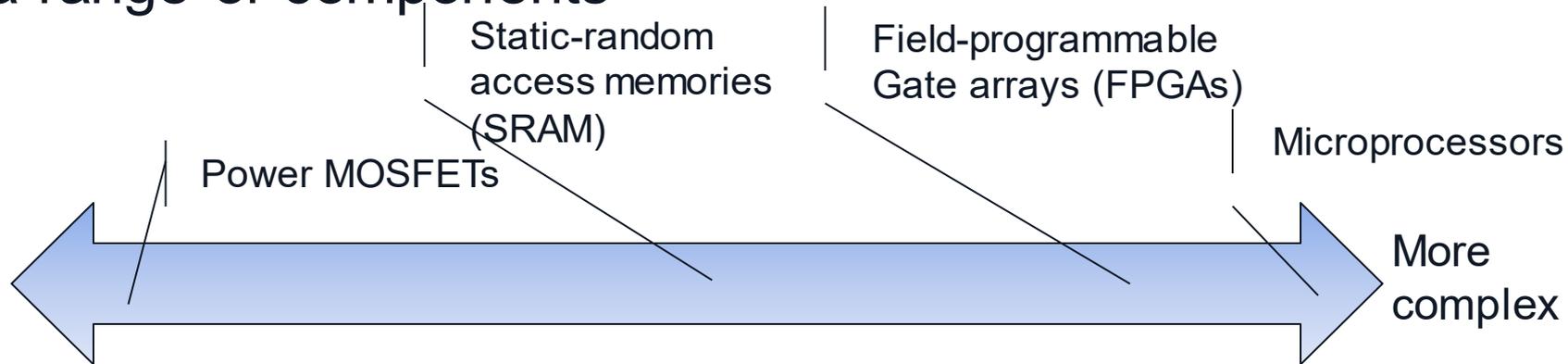


In This Short Course

- We will discuss:



- During this short course we will discuss these steps for a range of components





Basics of Testing



Designing Tests

- Determine what you are testing and how
 - Which components
 - Which radiation effects
 - What conditions



Designing Tests: Test standards

- Test standards are useful as a guide – the test standards will help you create repeatable tests
 - The test standard for total dose (Mil883) is designed to provide information about the worst possible radiation exposure scenario (total amount of radiation, temperature, voltage, etc.)
 - The test standard for SEE (heavy ion, and neutron) is designed to provide worst possible SEE conditions – radiation, angles, voltage and temperature
 - There are also test standards and guidelines for testing specific types of components, such as microprocessors and linears.
- Without test standards it would be hard to do component-to-component comparisons
- The test standards are an excellent place to start the test design process



Designing Tests: Defining Performance Requirements

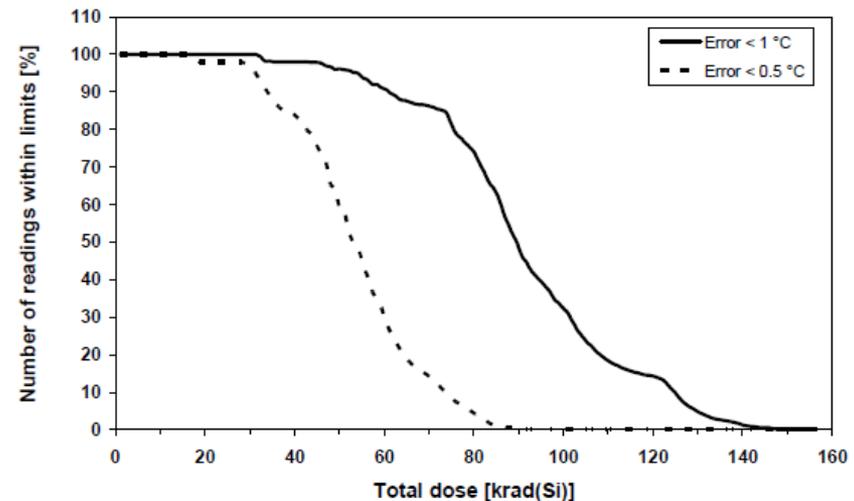
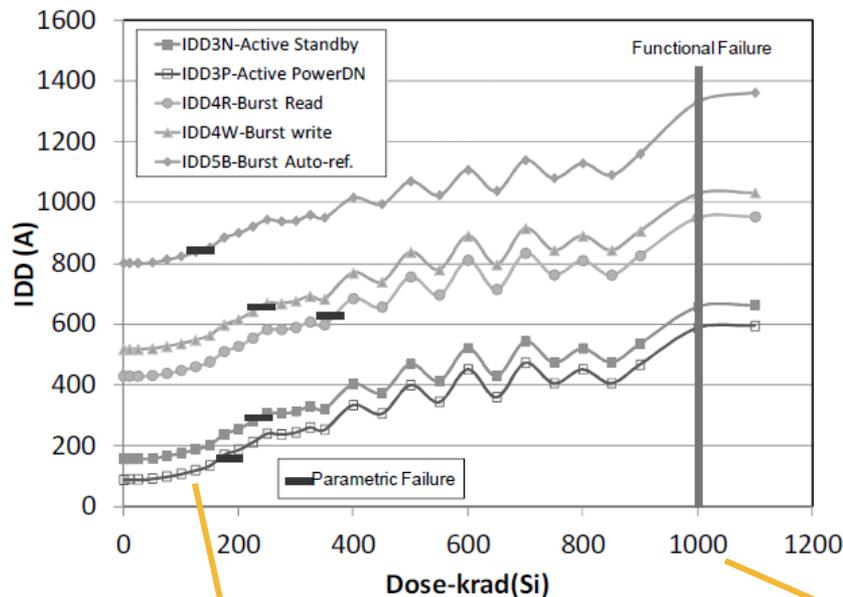
- Examples of performance requirements:
 - TID > 50 krad(Si) + margin
 - Onset of destructive SEEs > 70 MeV-cm²/mg
 - Rate for non-destructive SEEs < 1 x 10⁻¹¹ SEE/device-day
- Performance requirements are necessary on two fronts:
 - Informs the testers when to stop the test
 - Determines criteria for judging the quality of the components
- Ideally, these performance requirements are defined within the mission's documentation, but may have to be determined through interaction with the design and radiation teams
 - See Buchner's International School on the Effects of Radiation on Embedded Systems for Space Applications (SERESSA) Talk



Designing Tests: TID Tests

Cochran, D.J.; Boutte, A.J.; Dakai Chen; Pellish, J.A.; Ladbury, R.L.; Casey, M.C.; Campola, M.J.; Wilcox, E.P.; O'Bryan, M.V.; LaBel, K.A.; Lauenstein, J.; Batchelor, D.A.; Oldham, T.R.; "Compendium of Total Ionizing Dose and Displacement Damage for Candidate Spacecraft Electronics for NASA," *Radiation Effects Data Workshop (REDW)*, 2012 IEEE, vol., no., pp.1-9, 16-20 July 2012

Hofman, J.; Sharp, R.; "Measurement Methods for Total Ionizing Dose Testing: In-Situ versus Standard Practice," *Radiation Effects Data Workshop (REDW)*, 2012 IEEE, vol., no., pp.1-4, 16-20 July 2012



Monitor the performance

Monitor the amount of radiation





Designing Tests: SEE Tests

- There are many different types of SEE tests
- SEE tests are event-based – you need to keep a count of the radiation and the number of events.
- General requirements for all SEE tests:
 - SEE tests require the part to be biased at all times during the test
 - If the functionality needs to be determined, then inputs and outputs will be needed
 - Depending on the type of test, it might be necessary to clock or not clock
 - Static tests: not actively functioning
 - Dynamic tests: actively functioning
 - “Semi-static” and “semi-dynamic” from JPL Microprocessor Guideline
 - Define the facility – not all testing is appropriate at all facilities



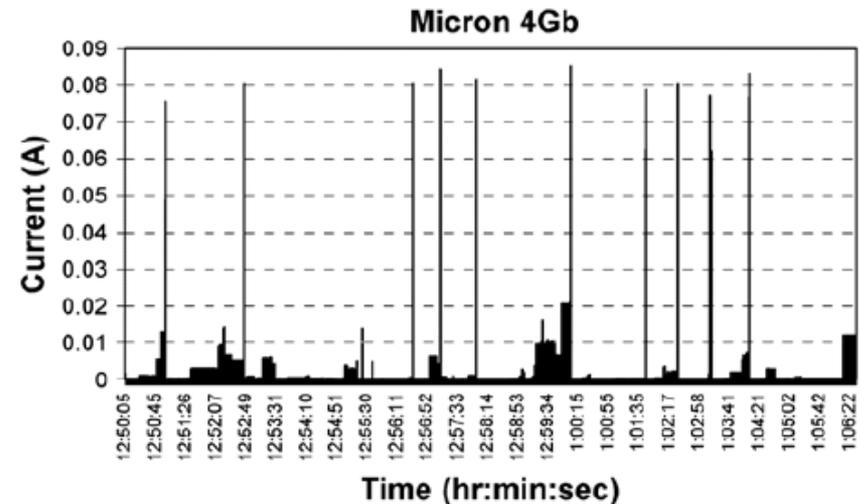
Test Fixture Design

- Most test fixtures need some very basic characteristics, including the ability to independently...
 - Clock
 - Bias
 - Write input values
 - Read output values
- It is necessary to be able to independently bias the component
 - So that problems with power regulators or other components do not affect the test
 - So that current consumption can be monitored
- Input and output vectors are only needed for tests that need functionality verified
 - Need a method for detecting functional errors



Test Fixture Design: Monitoring Current

- Need to be able to bias the component without using power regulators and/or a wall wart
- Ideally, you will need a power supply that will allow you to both power the component and log the current
- Many power supplies have:
 - Support a range of voltages
 - Provide over-current protection
 - Allow logging over General Purpose Interface Bus (GPIB)

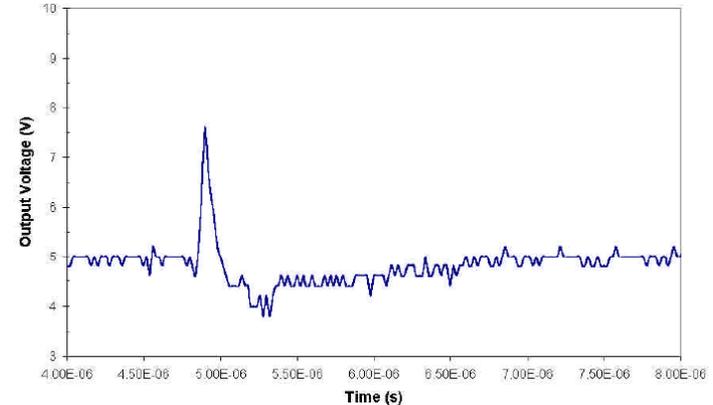


Oldham, T.R.; Berg, M.; Friendlich, M.; Wilcox, T.; Seidleck, C.; LaBel, K.A.; Irom, F.; Buchner, S.P.; McMorro, D.; Mavis, D.G.; Eaton, P.H.; Castillo, J.; , "Investigation of Current Spike Phenomena during Heavy Ion Irradiation of NAND Flash Memories," *Radiation Effects Data Workshop (REDW), 2011 IEEE* , vol., no., pp.1-9, 25-29 July 2011

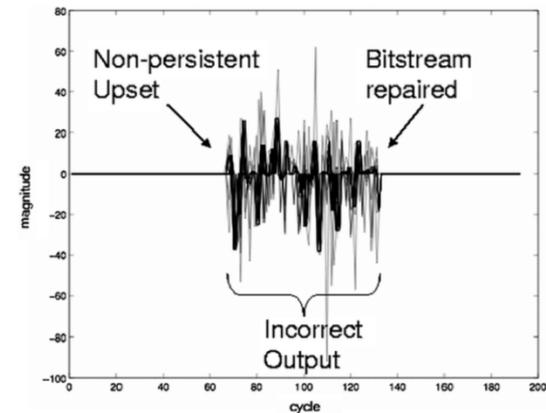


Test Fixture Design: Checking Functionality

- Need a quick way to detect when the output vectors are incorrect
- For analog tests, often times only transient detection is needed
 - An oscilloscope with the ability to trigger on transients and save screenshots is a reasonable solution
- For many components it is often necessary to determine whether one of many output vectors are not correct
 - Because errors can be input-vector dependent, it is possible that no, some or all output vectors are in error
 - As clock speeds increase, detecting incorrect output values at speed is challenging



http://radhome.gsfc.nasa.gov/radhome/papers/2003_Linear.pdf



Morgan, K.; Caffrey, M.; Graham, P.; Johnson, E.; Pratt, B.; Wirthlin, M.; , "SEU-induced persistent error propagation in FPGAs," *Nuclear Science, IEEE Transactions on* , vol.52, no.6, pp. 2438- 2445, Dec. 2005

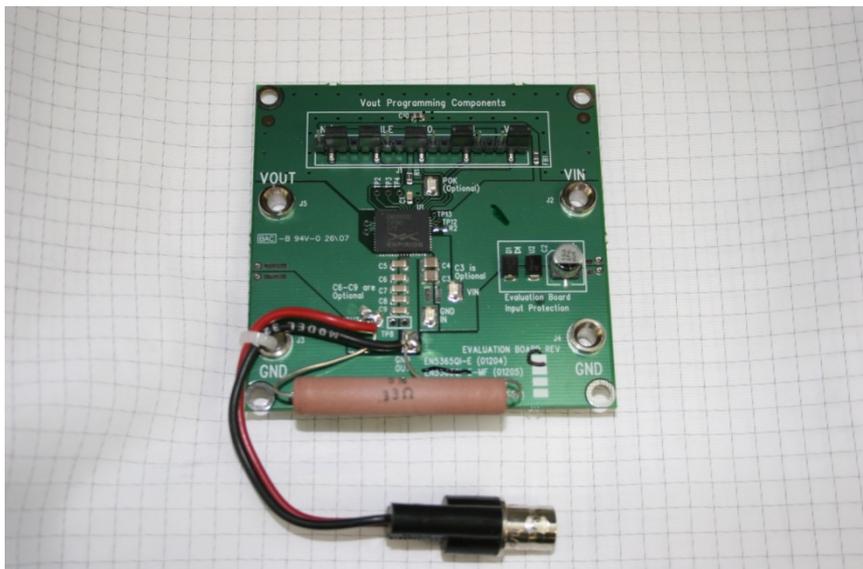


Test Fixture Design: Test Setups for Functional Verification

- Simpler parts, such as analog-to-digital converters (ADCs), digital-to-analog converters (DACs), point-of-load (POL) converters, operational amplifiers (Op-Amps), can often be tested using power supplies, functional generators, waveform generators and/or oscilloscopes
- Memories, ADC, DACs, and FPGAs can be tested using FPGA-based test fixtures
- Some complex components, such as microprocessors, can use secondary computers and sophisticated software tools



Test Fixture Design: Simple Test Setups

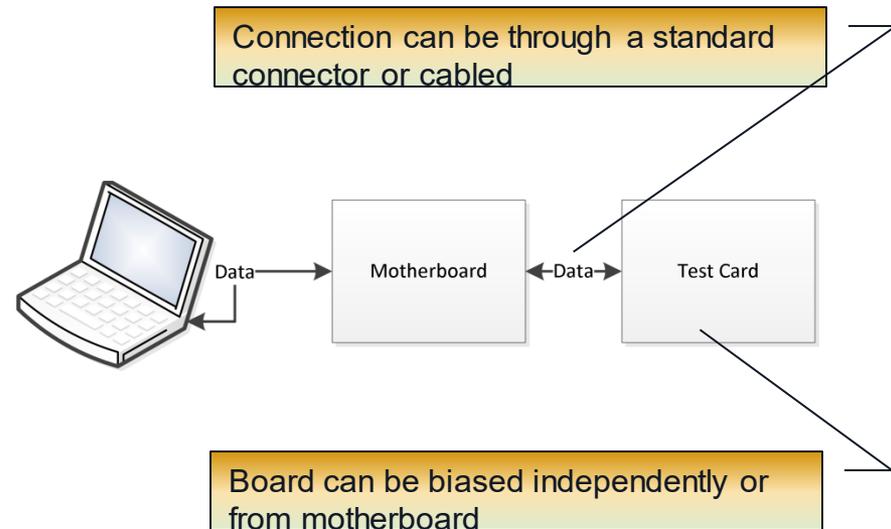


- Use evaluation boards when possible
- Modify boards as necessary:
 - Bypass power setup on board to allow the use of a programmable power supply
 - Add loads as necessary
 - Attach test points to an oscilloscope



Test Fixture Design: Motherboard-Daughter Card Test Fixtures

- There can be a lot of recurring engineering costs in test fixture design, such as
 - Designing the interface between the test board and the logging system
 - Designing systems for providing error detection and/or correction
- In recent years, many organizations have created motherboard-daughter card systems, where the test fixture is split into two separate systems

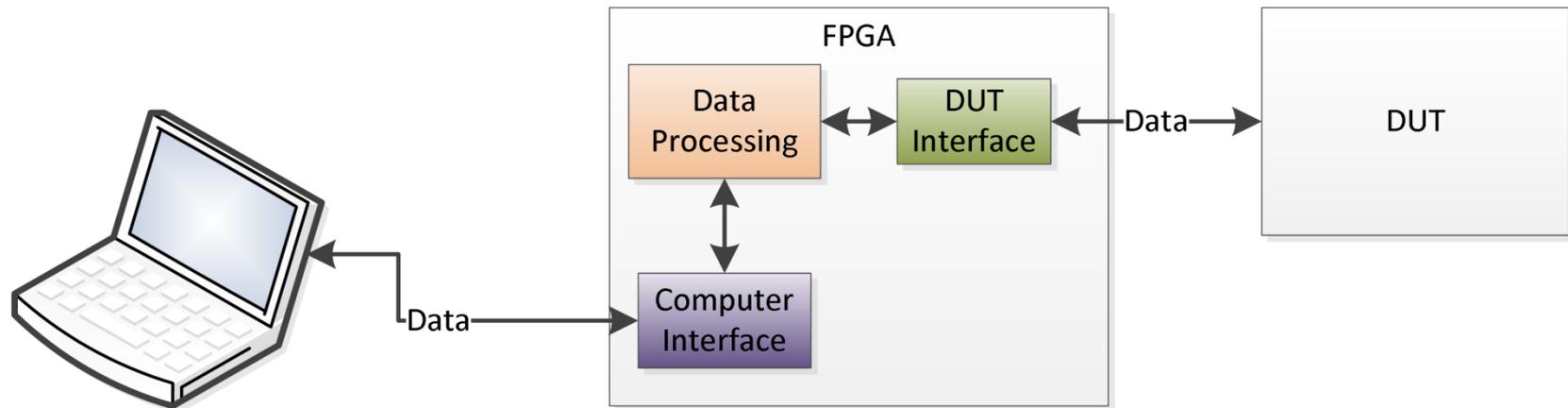




Test Fixture Design:

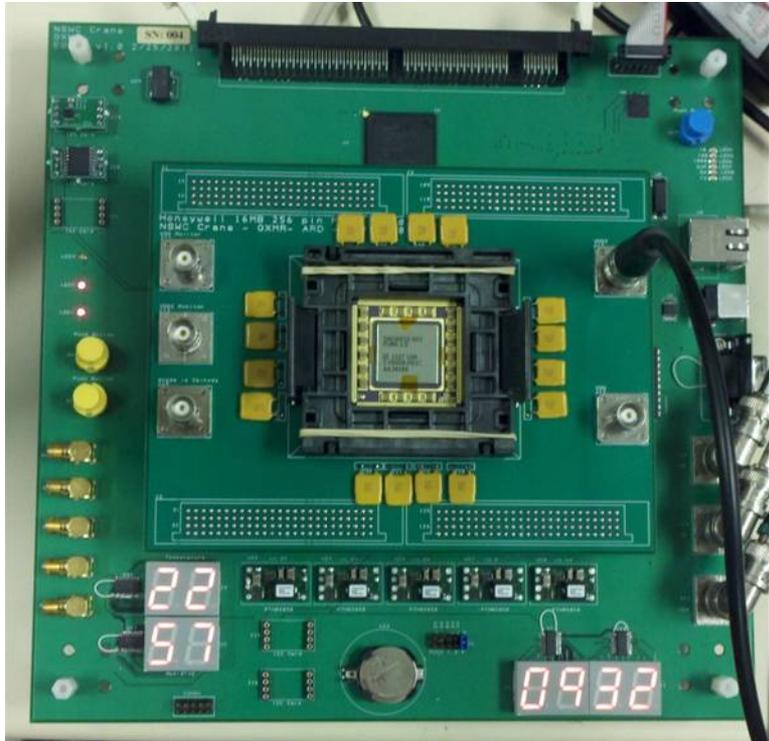
FPGA-based Motherboard-Daughter Card Test Fixtures

- Because FPGAs allow designers to create custom circuits, they are particularly useful for creating custom interfaces to:
 - ADCs
 - DACs
 - SRAM
 - Dynamic Random Access Memory (DRAM)
 - FPGAs

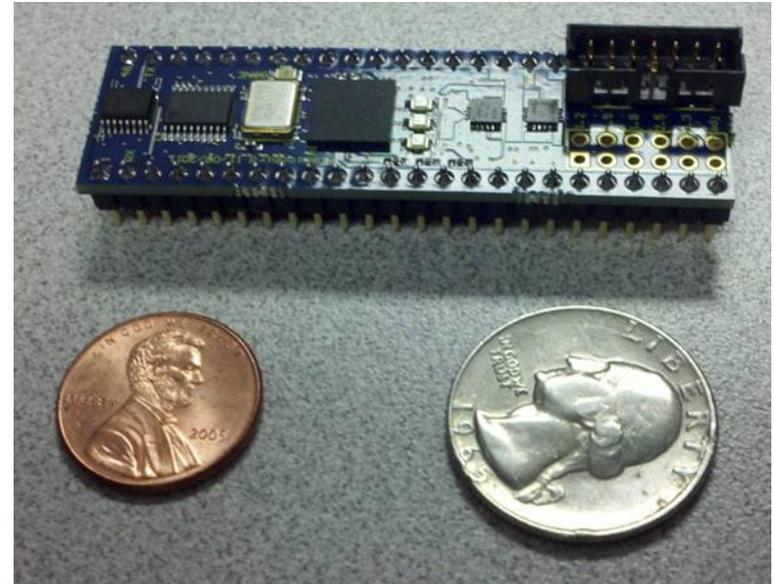




Test Fixture Design: Crane Egret and Hummingbird



Used by permission

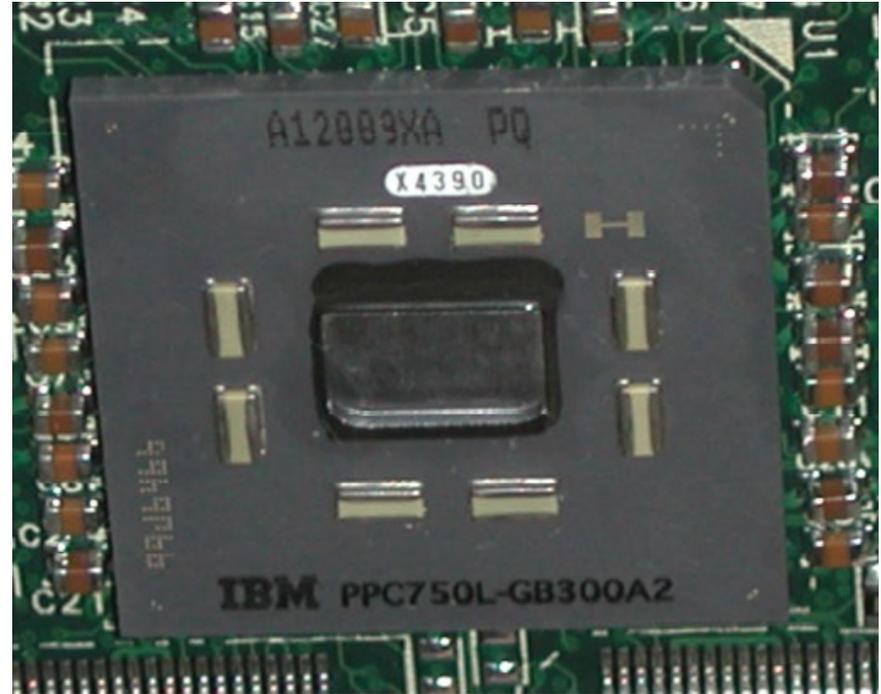


Used by permission



Test Fixture Design: Preparing Components for Testing

- For heavy ion testing, it is necessary to de-lid and thin components – most likely an outsourced activity for most organizations
 - If testing at angles, might need to be thinner
- Delidded parts are fragile and are easily damaged during shipping
 - Putting kapton tape over the surface can help



<http://trs-new.jpl.nasa.gov/dspace/bitstream/2014/40790/1/08-13.pdf>



Test Fixture Design: Minimizing Equipment in the Cave

- The test fixture needs to be split between what is in the radiation environment and what is not:
 - Ideally only the component with a minimalistic printed circuit board (PCB) are in the cave
 - The power supplies are usually in the cave
 - Anything needed for real-time monitoring needs to be outside – get the data out of the cave
 - We had a lot of problems with hard drives in proton and neutron environments
- If you need to leave equipment in the cave:
 - In heavy ion: generally not a problem
 - Proton: pack the equipment in Boron or lead
 - Neutron: pack the equipment in concrete, steel or polyethylene, although nothing is going to help you much except distance from the beam
 - Prompt: just get it out



Test Fixture Design: Cables

- Make certain you know the minimum length of the cables from the cave to the user facility
 - Coax, serial and Ethernet are useful for facilities where the minimum length is over 20'
 - Universal serial bus (USB) extenders provide the capability to drive USB longer distances
- Test cables and the test fixture before leaving for the test.





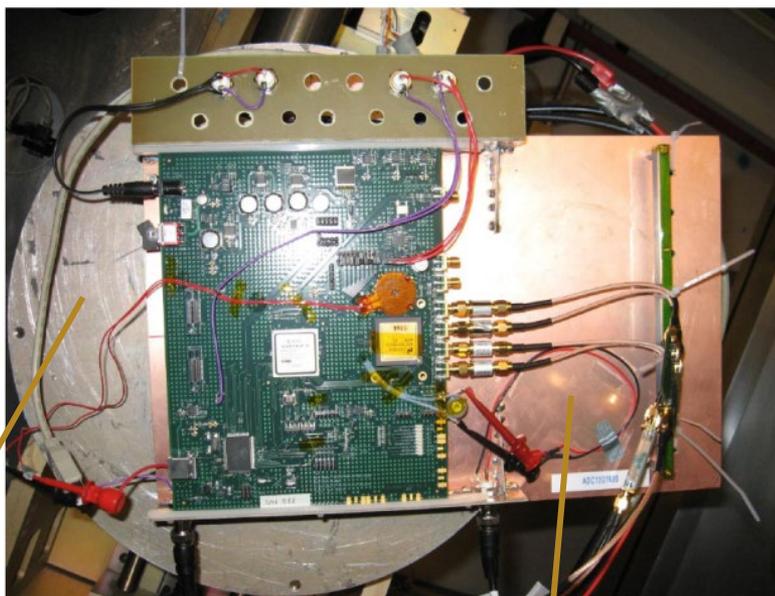
Test Fixture Design:

Cables and Connectors Exist to Break Your Heart

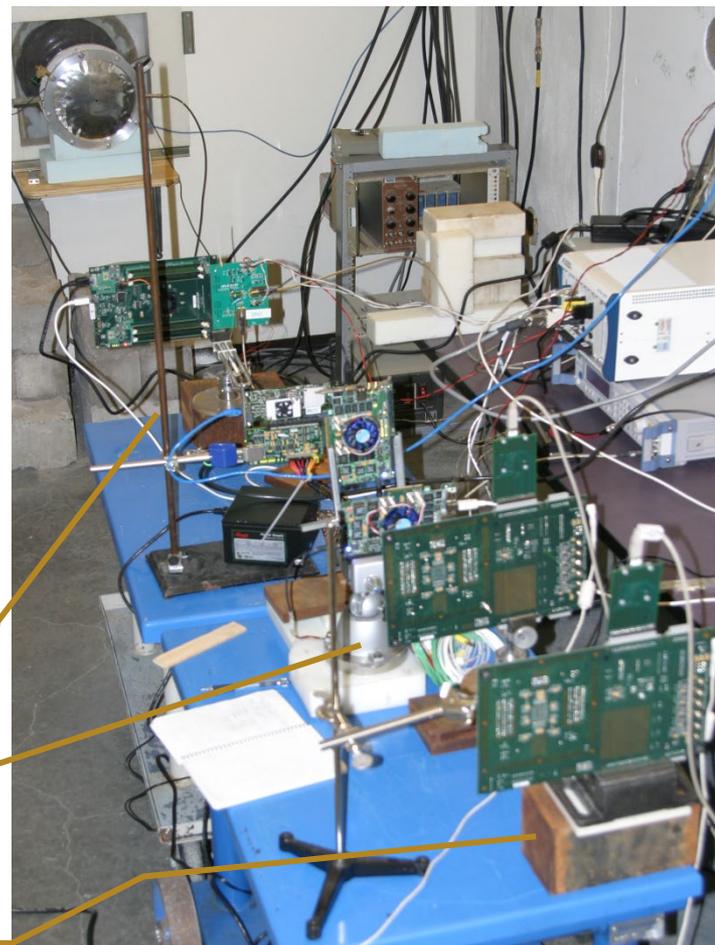
- Cables break, get lost, get tangled
 - Be careful pulling on the cables – they break
 - Use tie wraps to keep the weight of the cable from pulling over the test fixture
 - Cables can turn into giant antennas during prompt dose testing
 - Buy good cables – you do not want bad cables to affect your data
- Some connectors can often only be used a finite number of times
- Broken or faulty cables and connectors can be painful to diagnose



Test Fixture Design: The Mechanical Test Fixture



Single Event Effects Test Report: Texas Instruments
ADC12D1600QML-SP 12 Bit, Dual Channel 800/1600/3200 GSPS
Analog-to-Digital Converter, photo by Thang Trinh



LBL Flange/Mount

Lab stand

Mount

Vise

Assortment of steel and poly blocks
found around the accelerator





Test and Test Fixture Design: Documentation of the Test Design and Setup

- Documenting the test design and test fixture is necessary
- Before:
 - Writing about the test design and test fixture will sort your thoughts
 - Document how the test fixture works on the bench, what cables you used, and how it was set up
 - Information about what to pack and the procedure for the test
- During:
 - Take photos of the test setup
- After:
 - Immediately, you will need the information for the test report
 - Years later when you have completely forgotten the test, documentation on what you did, decisions made, what hardware used will be useful
 - If a test ever needs to be repeated, then it should be easy to pull the test fixture back together – as long as you have not cut any cables, lost the hardware, salvaged the power supply or cannot buy missing equipment any more



Test Execution

- There are three phases to a test:
 - Setup
 - Execution
 - Breakdown



Test Execution: Setup

- During setup, you need to find your equipment, unpack it, and set it up
- This part should be the easy part, as long as everything worked when you left and as your equipment survived the trip to the facility.
- There is a lot of last-minute rushing around for some people – the test fixture wasn't completely working when they shipped it, something broke in shipping
 - Bring backup tests in case something breaks.



Test Execution: Execution

- Most of testing is about counting – counting the radiation and counting the events
 - Make certain you keep track of the dosimetry – most facilities provide an electronic record but not always
 - Make certain you get the logs out of the cave as quickly as possible – always make backups of the logs so that you do not lose your data
 - Make certain any data you are writing down by hand is accurate – often times it won't be
 - Take more notes than you think you need. You never know what you need until later



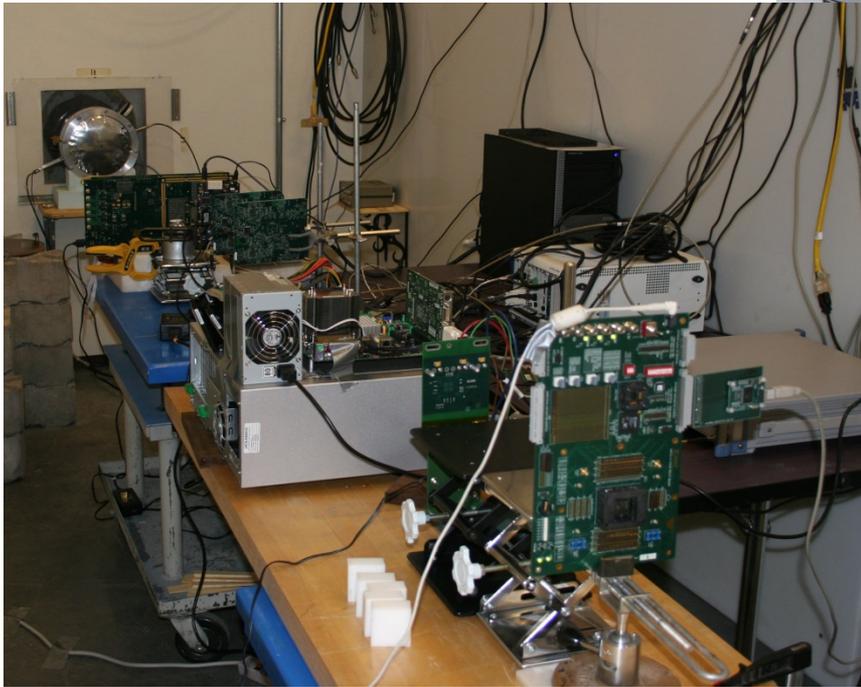
Test Execution: Dosimetry and Beam Uniformity

- Besides ions the most important things a facility gives you is dosimetry and uniformity
 - Check to make sure you are getting the right dosimetry – golden components are useful for checking dosimetry
 - Check uniformity using the software controlling user access to the beam (Beam Monster)
- If you do not have a good idea what is going on with the beam, the data might be useless



There Is A Giant Pile Of Cookies Behind Jeff Barton

Eight tests in the beam...



...Monitoring it in the user facility



Test Execution: A Few Thoughts on Safety

- There are a number of hazards at accelerators:
 - Tripping hazards
 - Radioactivity
 - Sleeplessness
- There is a fine line between a funny story and an accident
 - Be careful running up and down the stairs
 - Be careful handling lead/steel bricks
 - Be careful handling “hot” (radioactively or thermally) equipment
 - Be careful running into proton caves too quickly



Test Execution: Breakdown

- Most of this part of the test will focus on gathering all of your equipment and repacking it
- You will also need to work with the facility to see what their rules are for removing equipment from the cave
 - In some facilities, any equipment in the beam line or the cave must be surveyed before being removed
 - Keep in mind: some equipment might be activated and you will leave it behind for days, weeks, months
 - In the meantime, did you need that equipment for another test?
- Most people just stuff equipment hither and tither – try to be more organized as you don't want your equipment to break on the way back



Analysis

- After the test, the test data are analyzed
- For SEE tests need to:
 - Calculate cross-sections
 - Calculate the linear energy transfer (LET) in heavy ion to determine the LET at the active region using Stopping and Range of Ions in Matter (SRIM) tool
 - The LET changes as it moves through material (air, Silicon, etc.) – SRIM will help you determine what the LET is when you get to the active region
 - Fit data to curves
 - Calculate errors rates
 - Determine whether it meets performance requirements
- For TID tests need to:
 - Determine highest level of TID where functionality is maintained
 - Determine whether it meets performance requirements, including any margining
- There are many good sources on the basics of analyzing data



Analysis: Event Categorization

- For some components, SEU, SET and SEFI data sets will be intermingled
 - Intermingled data sets could skew cross sections
 - The data set needs to be separated into separate data sets by event type
- Keep in mind that SEFIs generally affect the functionality of the component
 - Might need to throw out data around the SEFI event
 - Will need to adjust the fluence if you do throw out the data
- Possible options for separating data sets:
 - Applying thresholds
 - Correlating physical locations
 - Jackknifing



Analysis: Jackknifing

- For more subtle classifications, Jackknifing might be useful [1]
- Statistically this process is used to remove statistical outliers from a set of data, but can also be used to categorize data sets into subsets
- Jackknifing examines small windows of data to determine if there are statistical outliers inside the window:
 - A series of means and standard deviations are calculated for the window of data with one member of the window removed
 - Then calculate a mean and a standard of deviation for the series
 - If removing one member of the data set can be removed such that the mean of means can be reduced, then the member is removed from the window
- Keep in mind:
 - Ask yourself this question: is there a reason for statistical outliers?

[1] B. Efron and R. Tibshirani, An Introduction to the Bootstrap. Chapman and Hall/CRC, 1993.



Analysis: Jackknifing

Examining subsets

Why is this data point so large?

Comparing the means of the subsets

Window of data {1, 2, 3, 70}	mean =	19	std dev =	34.0098
------------------------------	--------	----	-----------	---------

Subset 1	{2, 3, 70}	mean =	25	std dev =	38.97435
Subset 2	{1, 3, 70}	mean =	24.66667	std dev =	39.27255
Subset 3	{1, 2, 70}	mean =	24.33333	std dev =	39.55165
Subset 4	{1, 2, 3,}	mean =	2	std dev =	1

mean of means (MOM)	19	std dev of means =	11.3366
---------------------	----	--------------------	---------

comparison of subset 1	mean =	25	MOM - mean =	6
comparison of subset 2	mean =	24.66667	MOM - mean =	5.666667
comparison of subset 3	mean =	24.33333	MOM - mean =	5.333333
comparison of subset 4	mean =	2	MOM - mean =	17

Subsets are fine

Comparing each subset to the means of means

For this subset (MOM-means) > std dev of means – the removed element is skewing the data set. Might want to separate this window into two different categories and determine what is going on with all of the data



Analysis: Cross Section

- After categorizing events, cross sections for each category are calculated using this formula:

$$\sigma(LET) = \frac{Events}{Fluence}$$

- If the data point was collected when the test fixture was angled, then the cross section equation is:

$$\sigma(LET, \theta) = \frac{Events}{Fluence * \cos(\theta)}$$

- LET also changes with angle:

$$Effective\ LET(\theta) = \frac{LET(0^\circ)}{\cos(\theta)}$$

- The inverse cosine law is “breaking down” for some components – keep track of your angles and not just Effective LET when testing



Analysis: Error Bars

- It is important that all experimental data include error bars
 - Error bars provide context for the collected data – small error bars mean that there is a lot of data behind the data point
 - Error bars help us compare data points – when error bars overlap between data points it could mean that the only difference is test error
 - By convention this data is expressed as Events \pm Error Bars
- As a convention, the radiation effects community calculates error bars for cross-sections based on this formula:

$$\frac{2 * \sqrt{events}}{fluence}$$

- These are 95% confidence intervals, which means that with repeated tests 1 in 20 tests will be outside of the bounds of:

$$\left(-\frac{2 * \sqrt{events}}{fluence} , \frac{2 * \sqrt{events}}{fluence} \right)$$



Analysis: Sparse Data

- The previous recommendation on error bars assumes that there are more than 50 events collected
- Sometimes it is not possible to collect 50 events on a particular type of fault:
 - Destructive events tend to kill components
 - SEFIs can be rare
 - SEUs are dependent on the number of memory bits
- If there is less than 50 events, the Poisson error bars for 95% confidence intervals that were calculated in the 1930s are used
 - For these error bars, the error bars are not symmetric, so there is both an upper and lower error bar
 - In this case, the data will be expressed as Events (lower error bar, upper error bar)



Analysis: Null Data

- Occasionally there will be a null data set:
 - Below the onset for the effect
 - Effect might not have occurred for that data set
- By convention there are two ways to deal with null data:
 - Place data points with downward arrows on the graph
 - Calculate the cross-section based on the formula below and using the error bars for 0

$$\frac{1}{\textit{fluence}}$$

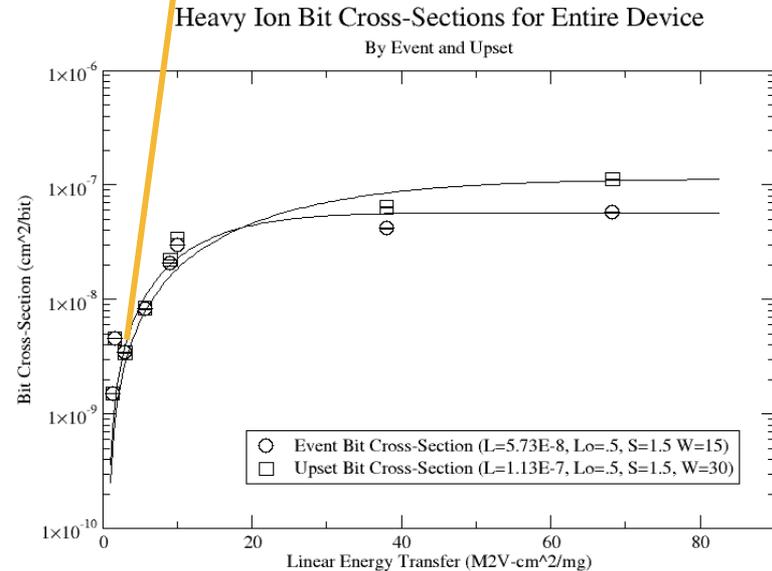
- We prefer the second to the first, because the reader can determine if the test did not run long enough (too little fluence) or whether the effect is truly absent for that data point



Analysis: Be Wary of What Your Data Tells You

- While log-linear is the usual way to plot data points, looking at the data in linear-linear might reveal outliers or unusualness
- Plot your data while testing to see if anything odd is occurring
 - While rare, there could be a dosimetry problem during the test that is affecting the data
 - Chase down an outlier – maybe the outlier is failure mode you did not expect

What is wrong with this data point?



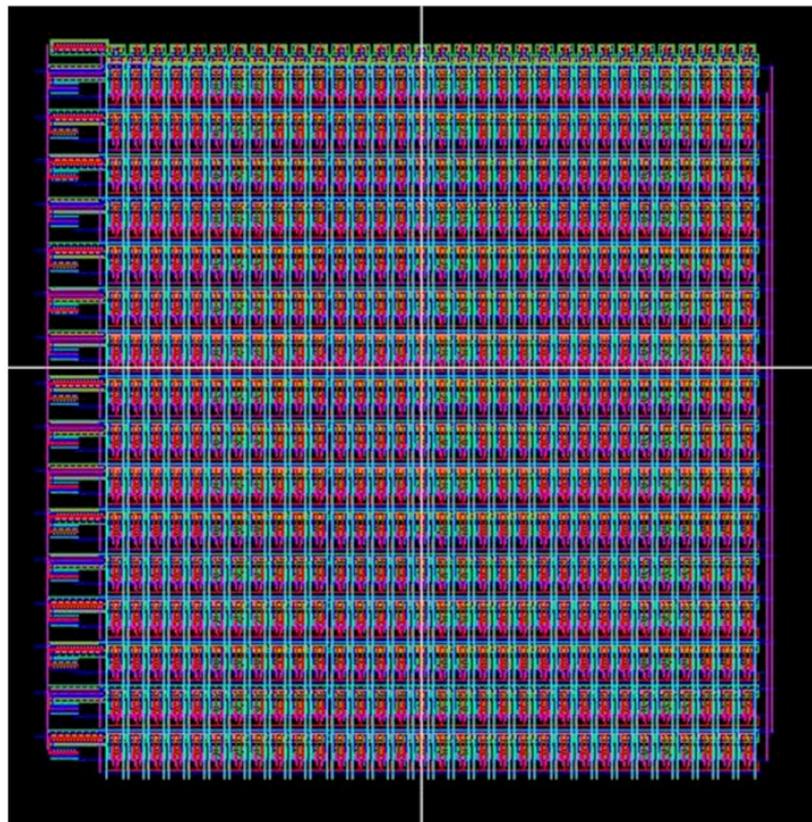
H. Quinn, K. Morgan, P. Graham, J. Krone, and M. Caffrey, "Static Proton and Heavy Ion Testing of the Xilinx Virtex-5 Device," in *The Proceedings of Data Workshop for Nuclear and Space Radiation Effects Conference, 2007*, pp. 177-184.



Case Study: SRAM



SRAM Case Study



http://s.eeweb.com/members/jessica_shoemaker/projects/2011/04/01/image21-1301665634.png

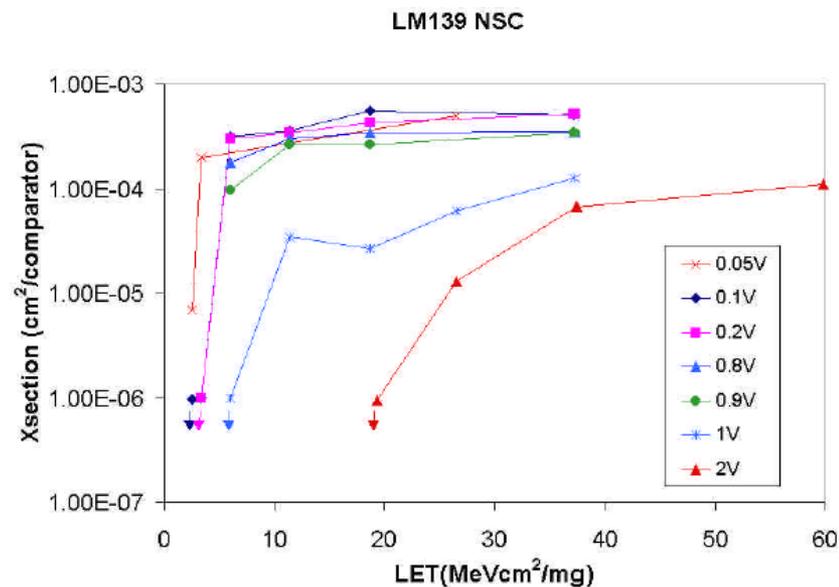
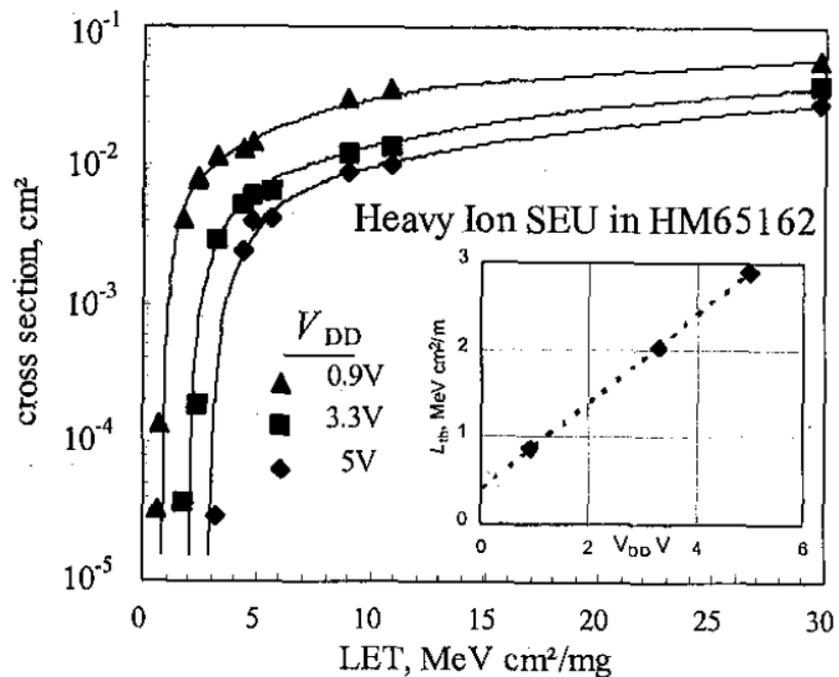


SRAM Case Study: Complications of Testing

- Sensitive to:
 - Voltage variations
 - Temperature variations
 - Usage conditions
 - Multiple-bit upsets (MBUs)
 - Pitch and roll angles
- Need to test a range of the above variations to get an idea of how the SRAM will perform when deployed



SRAM Case Study: Biasing

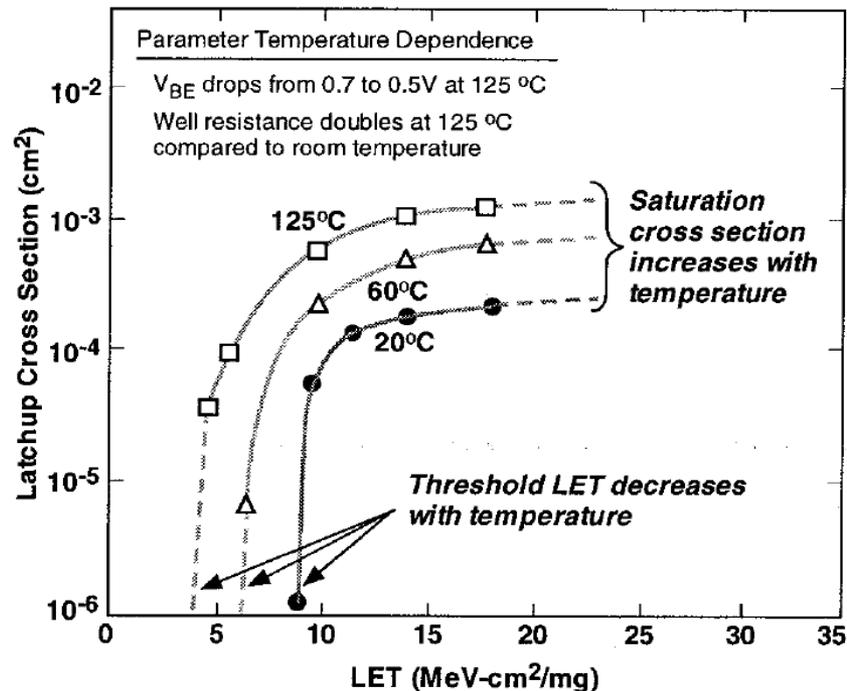


J. Barak, J. Levinson, A. Akkerman, E. Adler, A. Zentner, D. David, Y. Lifshitz, M. Hass, B. E. Fischer, M. Schlogl, M. Victoria, and W. Hajdas, "Scaling of SEU mapping and cross section, and proton induced SEU at reduced supply voltage," *IEEE Transactions on Nuclear Science*, vol. 46, pp. 1342-1353, Dec 1999.

C. Poivey, S. Buchner, J. Howard, and K. LaBel, "Testing Guidelines for Single Event Transient (SET) Testing of Linear Devices," http://radhome.gsfc.nasa.gov/radhome/papers/2003_Linear.pdf2003.



SRAM Case Study: Temperature



T. F. Miyahira, A. H. Johnston, H. N. Becker, S. D. LaLumondiere, and S. C. Moss, "Catastrophic latchup in CMOS analog-to-digital converters," *IEEE Transactions on Nuclear Science*, vol. 48, pp. 1833-1840, Dec 2001.
T. F. Miyahira and A. H. Johnston, "Latchup in CMOS Analog-to-Digital Converters," <http://trs-new.jpl.nasa.gov/dspace/bitstream/2014/36809/1/01-1287.pdf>

Schw ank, J.R.; Shaneyfelt, M.R.; Dodd, P.E., "Radiation Hardness Assurance Testing of Microelectronic Devices and Integrated Circuits: Test Guideline for Proton and Heavy Ion Single-Event Effects," *Nuclear Science, IEEE Transactions on*, vol.60, no.3, pp.2101,2118, June 2013.

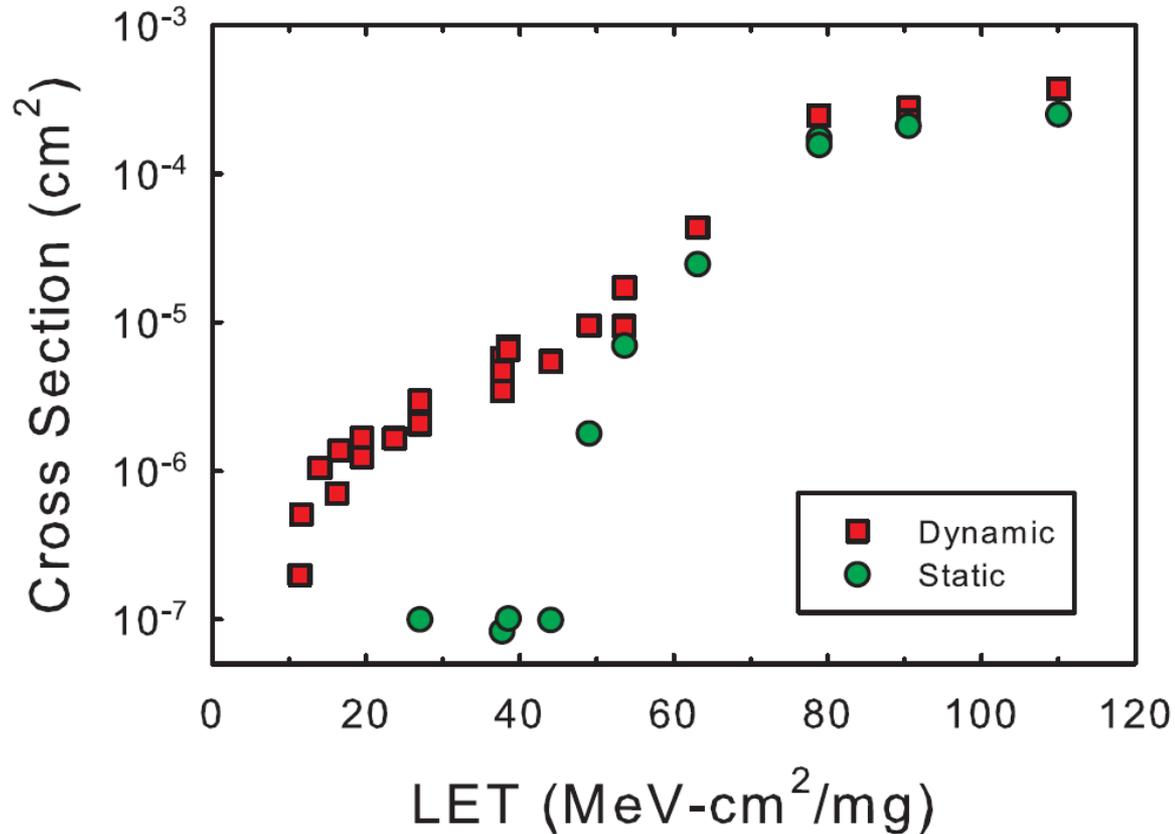


SRAM Case Study: Angles

- It is common in heavy ion testing to test at angles:
 - There are a limited number of ions at each beam, which translates into a limited number of LETs that can be tested
 - Due to the inverse cosine law, ions that come in at an angle will have a higher LET than normal incidence ions
 - It is best to use angling in the LET ranges where the cross-section is quickly changing, such as around the onset and around the knee
 - It is important for error rate calculations to get cross-section calculations in the lower energy and LET ranges correct, as it is often the most significant source of error in the error rate prediction
- Some types of layouts have to be tested at angle:
 - DICE latches should only be sensitive to upsets at an angle – necessary to test at a wide range of angles to cause faults



SRAM Case Study: Static vs. Dynamic Testing

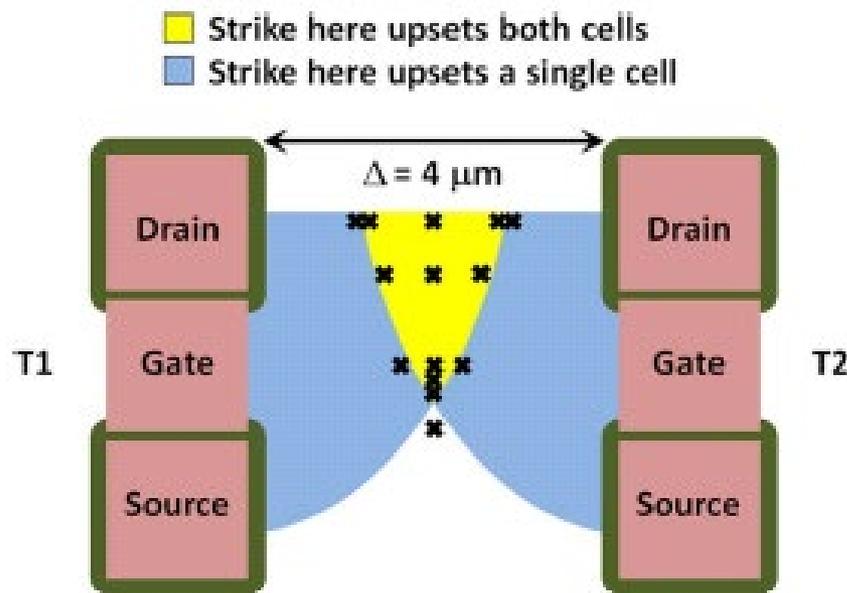


J. R. Schwank, M. R. Shaneyfelt, and P. E. Dodd, "Radiation Hardness Assurance Testing of Microelectronic Devices and Integrated Circuits: Radiation Environments, Physical Mechanisms, and Foundations for Hardness Assurance," to be published in the *IEEE Transactions on Nuclear Science*, 2013.



SRAM Case Study: Multiple-bit and Multiple-cell Upsets

- One of the reasons why researchers have been turning toward dynamic testing of SRAMs has been an increase in MBUs and MCUs
- If you want to collect information about the MBUs, you need to keep the number of upsets/read down so that you do not construct MBUs/MCUs:
 - As a general rule of thumb, the probability of constructing MBUs increases at 8x the rate of percentage of upset bits in the SRAM [1]

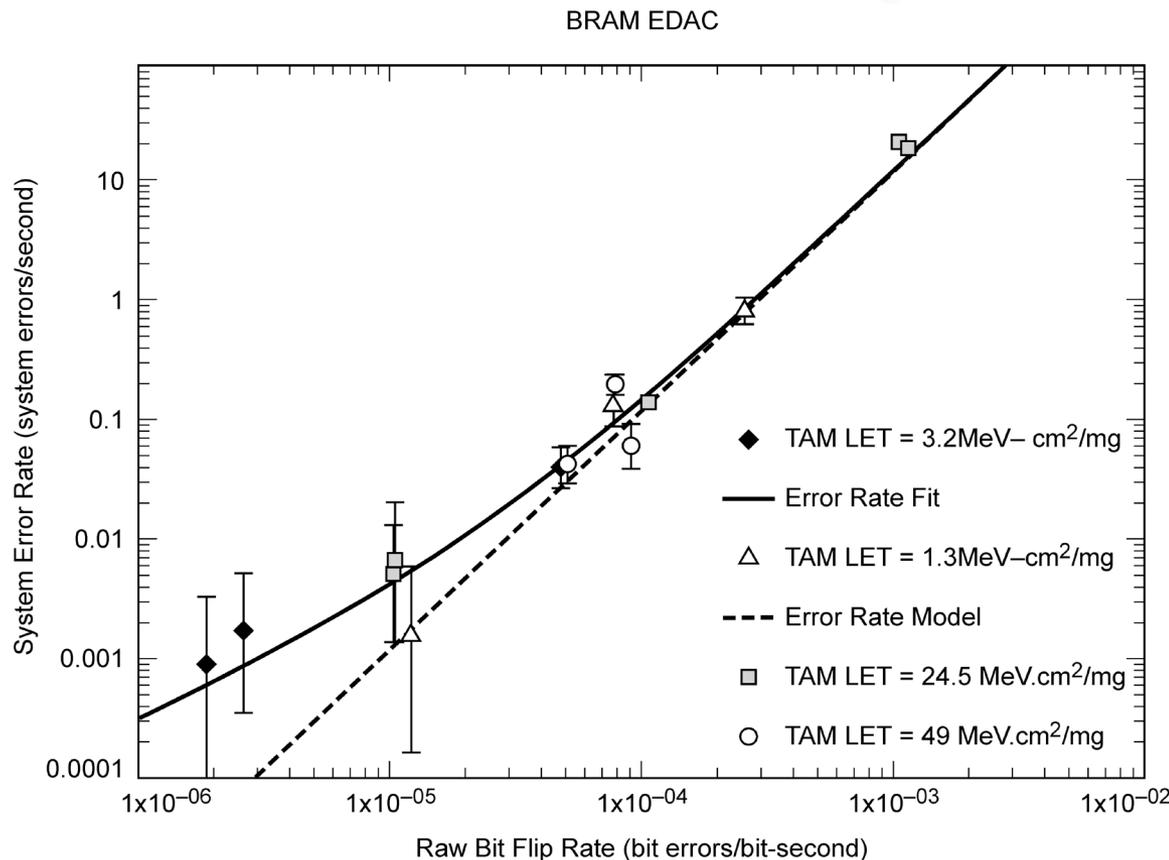


Black, J.D.; Dodd, P.E.; Warren, K.M., "Physics of Multiple-Node Charge Collection and Impacts on Single-Event Characterization and Soft Error Rate Prediction," *Nuclear Science, IEEE Transactions on*, vol.60, no.3, pp.1836,1851, June 2013

[1] Quinn, H.M.; Graham, P.S.; Wirthlin, M.J.; Pratt, B.; Morgan, K.S.; Caffrey, M.P.; Krone, J.B., "A Test Methodology for Determining Space Readiness of Xilinx SRAM-Based FPGA Devices and Designs," *Instrumentation and Measurement, IEEE Transactions on*, vol.58, no.10, pp.3380,3395, Oct. 2009



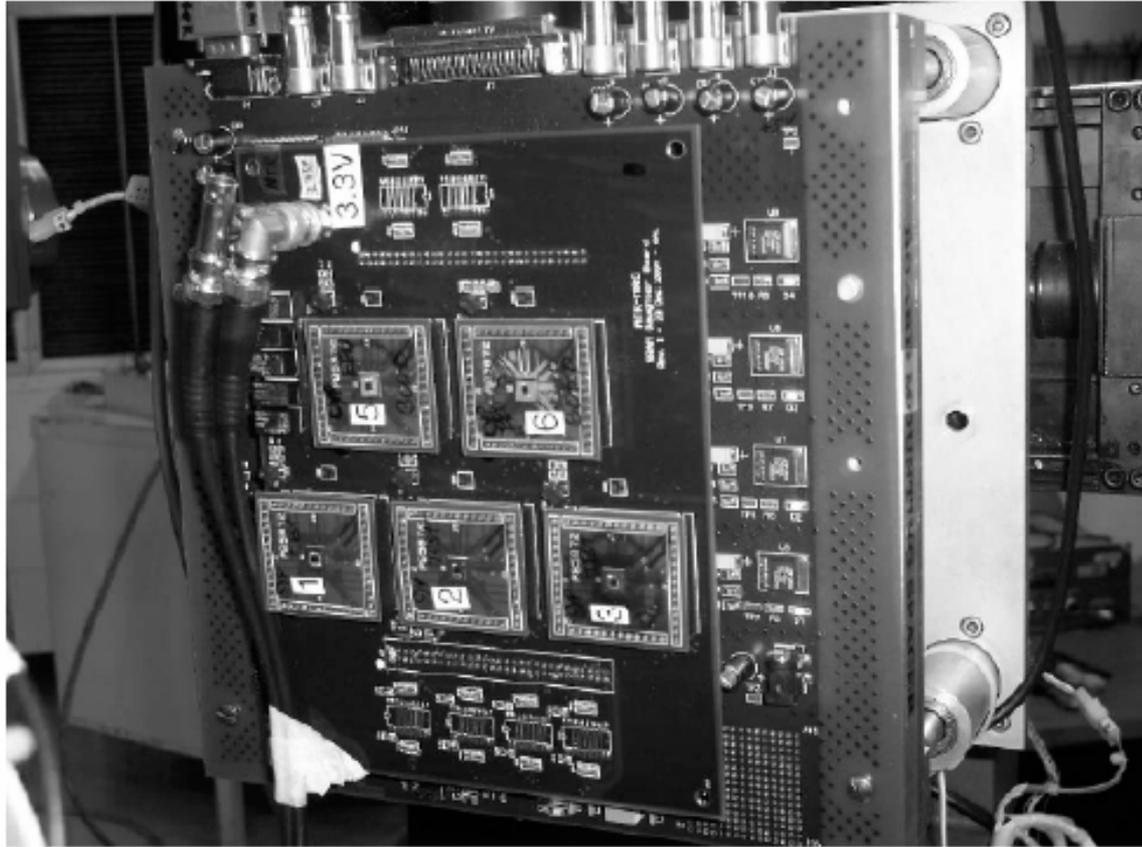
SRAM Case Study: Internal Fault-tolerance Techniques



S. M. Guertin, "SOC SEE Test Guideline Development," presented at the Single-Event Effects Symposium, San Deigo, CA, 2013.



SRAM Case Study: SRAM Test Fixtures

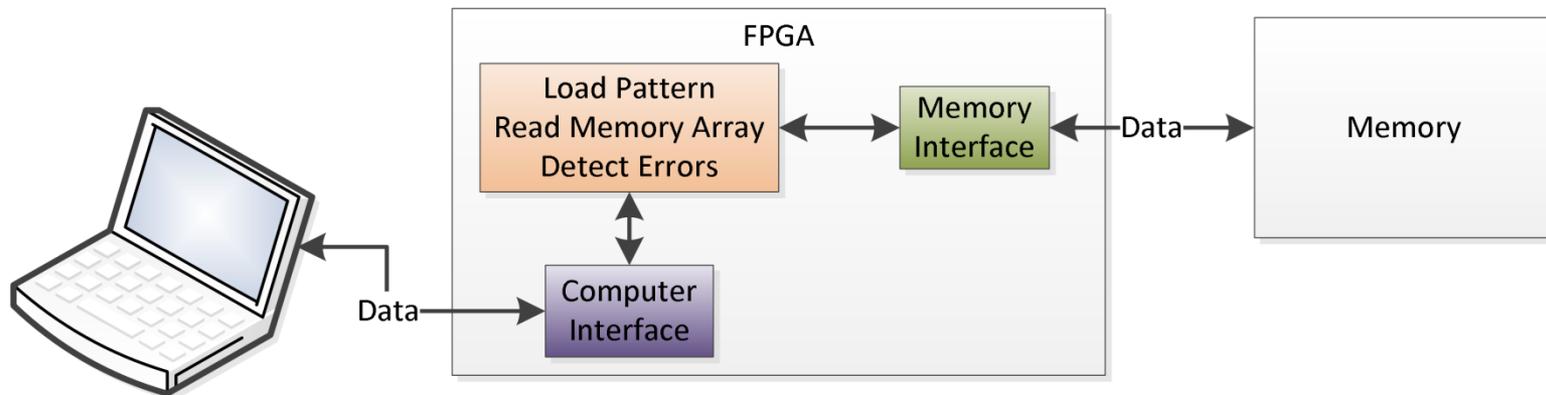


Page, T.E., Jr.; Benedetto, J.M., "Extreme latchup susceptibility in modern commercial-off-the-shelf (COTS) monolithic 1M and 4M CMOS static random-access memory (SRAM) devices," Radiation Effects Data Workshop, 2005. IEEE , vol., no., pp.1,7, 11-15 July 2005



SRAM Case Study: FPGA-based Test Fixtures for SRAM

- FPGA can
 - Perform memory reads and writes,
 - Determine what memory values are in error
 - Keep statistics, and
 - Return data to a computer
- Likely necessary to change the FPGA user circuit for the memory interface for each memory
 - Xilinx synthesis toolset includes a memory interface generator that can make this process simpler



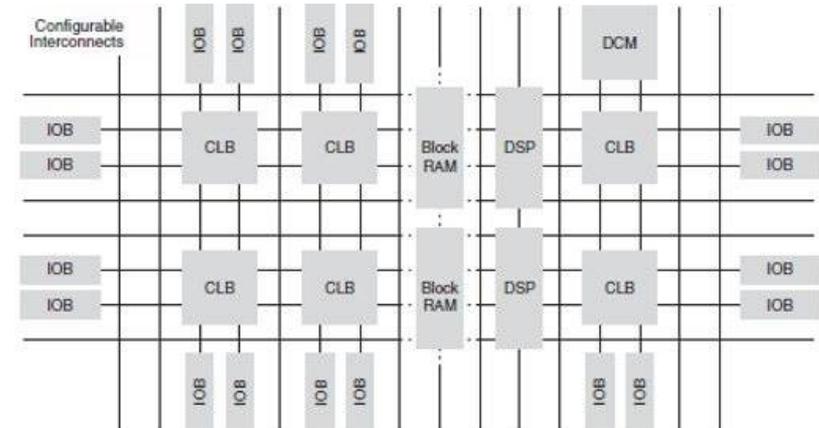


Case Study: SRAM-Based FPGAs



SRAM-based FPGA Case Study

- FPGAs are a reconfigurable technology that allows designers to develop custom hardware without the cost of application-specific integrated circuits
 - “Programmed” or “described” in a similar process to designing software
- The user circuits are programmed onto fabric:
 - Logic is converted to lookup tables
 - Wiring is converted to programmable routing where signals trace from point to point through switches
 - The major difference between different types of FPGAs is whether the logic and routing are implemented in SRAM, flash or fuses



<http://embedded.communities.intel.com/community/en/hardware/blog/2011/03/07/roving-reporter-processors-and-fpgas-match-well-in-data-intensive-applications>



SRAM-based FPGA Case Study:

Basics of Testing

- While the strength of the FPGA is “blankness” of the reconfigurable fabric that allows designers to develop custom hardware circuits, it is not actually true that FPGAs are just a homogenous sea of programmable logic and routing
 - There are specialized members on the fabric
 - There are many embedded “hard” (not reconfigurable) cores within the fabric
- There is also a factor in the inherent reliability to how the user circuit to mask errors, which is design dependent

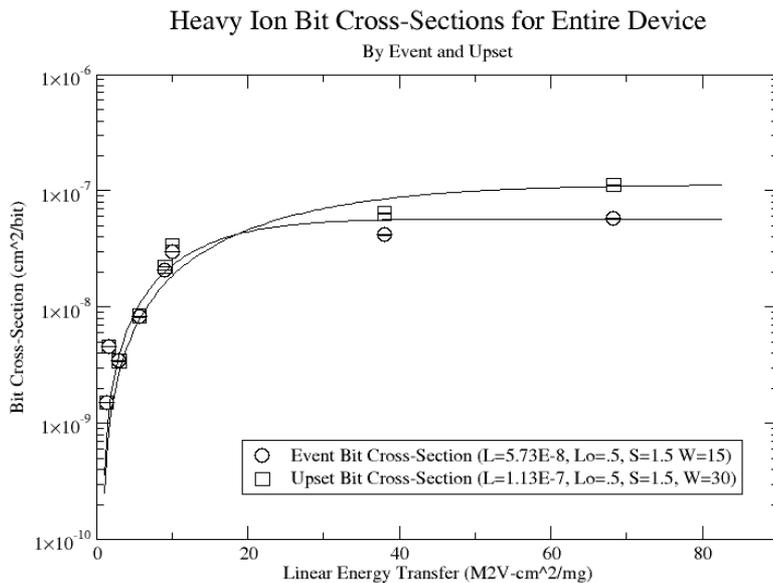


SRAM-based FPGA Case Study: Static vs. Dynamic Test Design

- Static testing of FPGAs focuses on determining the basic radiation sensitivity for the fabric – SEUs
 - The static cross-section for the components are often an upper bound on the radiation sensitivity, as the user circuit cannot use all of the bits in the component
 - Monitor only SEUs in the fabric through the configuration ports
- Dynamic testing of FPGAs focuses on SEUs affect the functionality of user circuits and the fabric:
 - The user circuit will not use all of the fabric – most SEUs in the unused portions will not affect the user circuit
 - The user circuit might also mask an error in the used portions – specific input combination might be needed to “trigger” the error
 - Monitor both SEUs in the fabric through configuration ports and monitor outputs for incorrect output data



SRAM-based FPGA Case Study: Virtex-5 Static Testing

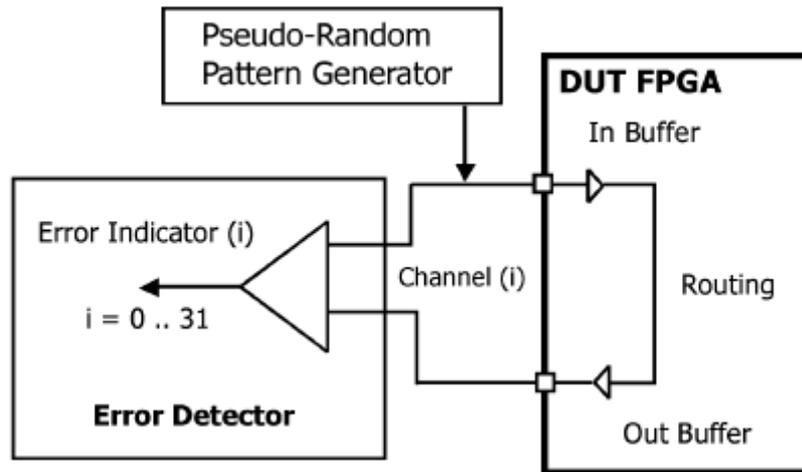


H. Quinn, K. Morgan, P. Graham, J. Krone, and M. Caffrey, "Static Proton and Heavy Ion Testing of the Xilinx Virtex-5 Device," in *The Proceedings of Data Workshop for Nuclear and Space Radiation Effects Conference, 2007*, pp. 177-184.

- The Virtex-5 test focused on just the SEU characteristics of the FPGA:
 - The entire configuration memory was read from the device several times a second
 - The entire configuration memory was overwritten after reading
 - Component was tested at both normal incidence and multiple angles
- Test show the SEU cross section is $1 \times 10^{-7} \text{ cm}^2$ with an onset below $1 \text{ MeV-cm}^2/\text{mg}$
- Test showed an energy- and angle-dependence on MBUs



SRAM-based FPGA Case Study: Dynamic Testing of Input/Output Buffers



Swift, G.M.; Rezgui, S.; George, J.; Carmichael, C.; Napier, M.; Maksymowicz, J.; Moore, J.; Lesea, A.; Koga, R.; Wrobel, T.F.; , "Dynamic testing of Xilinx Virtex-II field programmable gate array (FPGA) input/output blocks (IOBs)," *Nuclear Science, IEEE Transactions on*, vol.51, no.6, pp. 3469- 3474, Dec. 2004

- In this test, the testers were attempting to determine if SEUs in the configuration of the input/output blocks (IOBs) caused observable output data errors
- The test was designed to specifically isolate the IOBs so that no other aspect of the user circuit would affect the results
- The test design established that SEUs in the IOB configuration could cause observable output errors, and determined a cross section for output errors



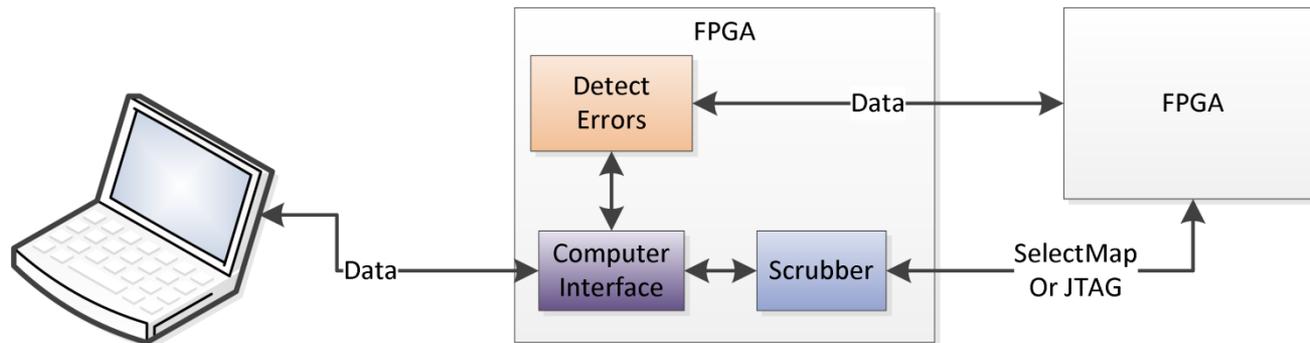
SRAM-based FPGA Case Study: Test Fixtures

- One of the issues with SRAM-based FPGAs is that they can accumulate SEUs really quickly in accelerated radiation environments and most successful tests need to find a way to remove SEUs quickly:
 - Off-line Reconfiguration: Completely overwrite the programming data, lose all of the current processing
 - On-line Reconfiguration (scrubbing): Completely or partially overwrite the programming data, do not lose current processing



SRAM-based FPGA Case Study: FPGA-based Test Fixtures

- It is common to use FPGAs while testing FPGAs, as it is possible to configure the auxiliary FPGA to:
 - Reconfigure/remove SEUs (scrub) the FPGA under test,
 - Detect/correct SEFIs, and
 - Detect/correct functional failures
- The auxiliary FPGA can perform these actions, as well as interface the data to a computer for logging



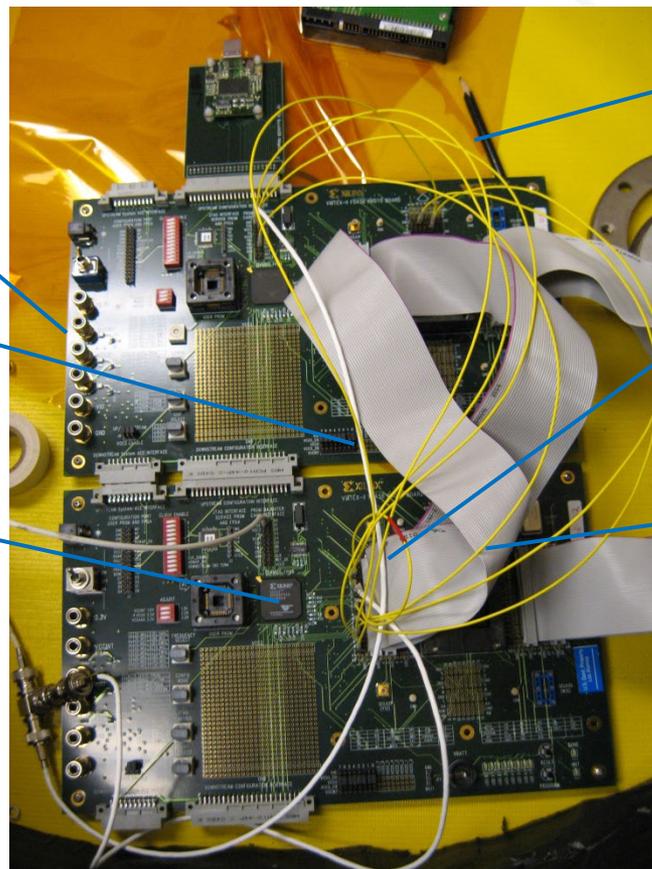


SRAM-based FPGA Case Study: A Cautionary Tale

Independently
bias components

Independently
clock components

Scrubbing and
functional monitoring



Jumper cables are
hard to set up
correctly, easy to
fall off

Clock cables
connect to fragile
connectors

Cables are
matched length

H. Quinn, K. Morgan, P. Graham, J. Krone, M. Caffrey, and K. Lundgreen,
"Domain Crossing Errors: Limitations on Single Device Triple-Modular
Redundancy Circuits in Xilinx FPGAs," *IEEE Transactions on Nuclear Science*,
vol. 54, pp. 2037-43, 2007.



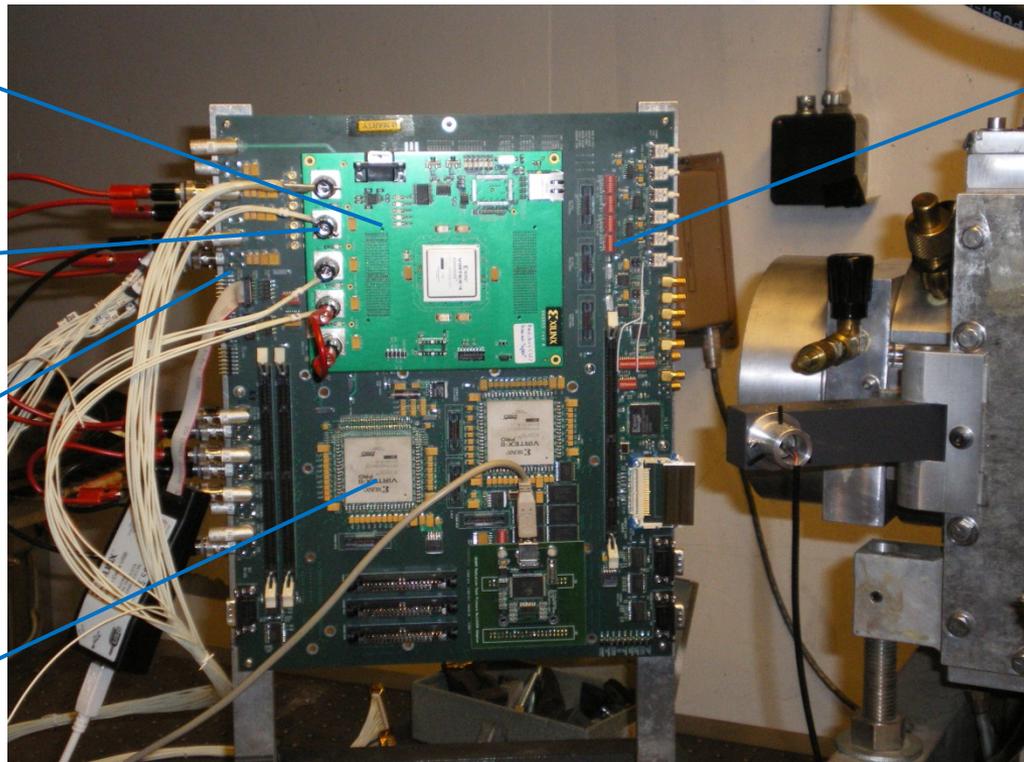
SRAM-based FPGA Case Study: SEAKR Board

Can test any component with a daughter card

Independently bias components

Independently clock components

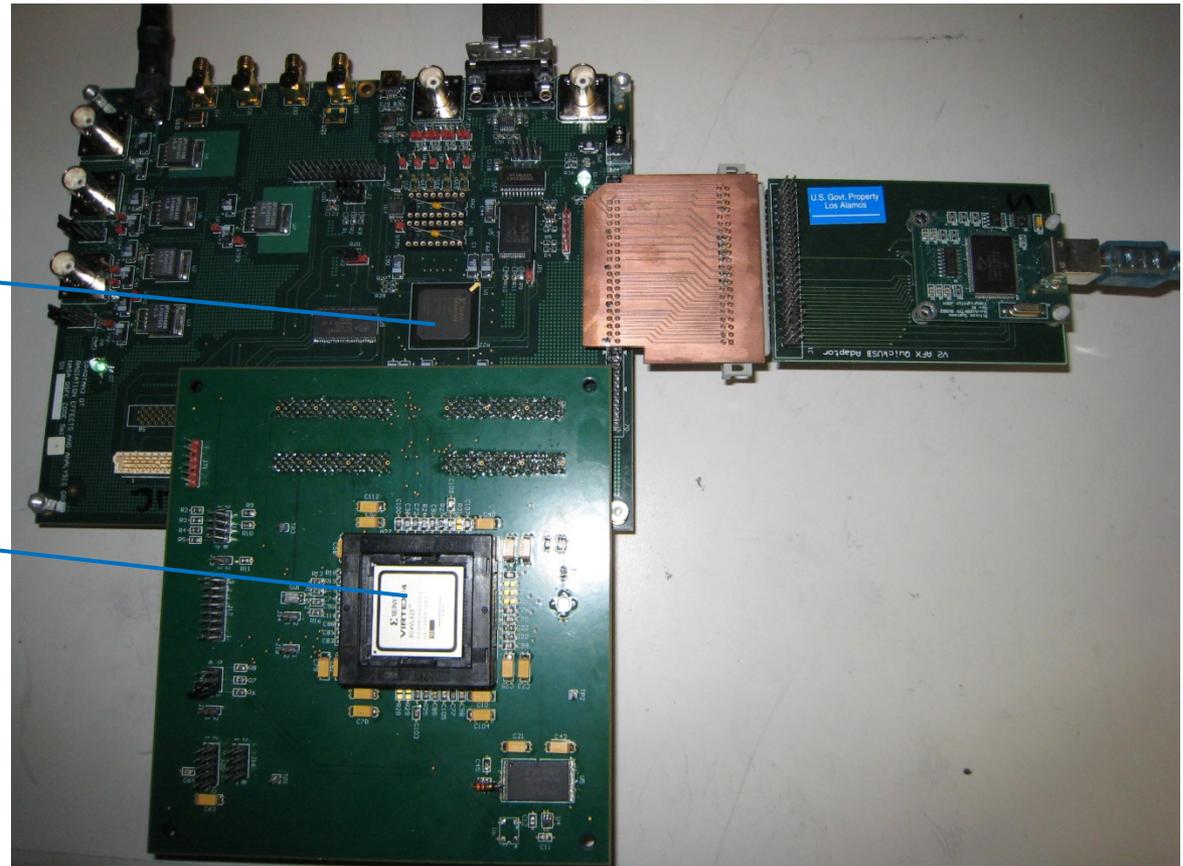
Scrubbing and functional monitoring



Difficult to assemble



SRAM-based FPGA Case Study: NASA Boards



Scrubbing and functional monitoring

Can test any component with a daughter card

M. Berg, "A Comparative Study of Field Programmable Gate Array Error Cross Sections: Putting Data into Perspective," presented at the Military and Aerospace FPGA and Applications (MAFA) Meeting, 2007.





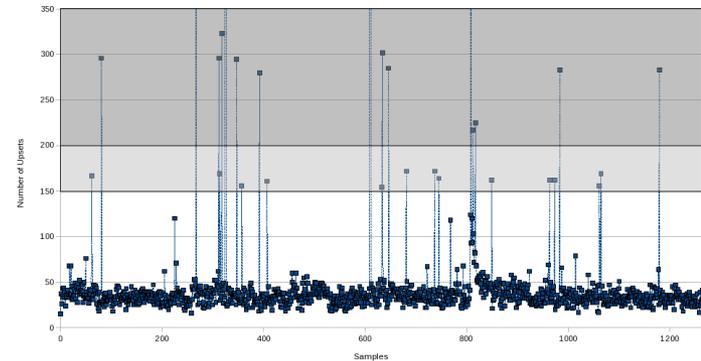
SRAM-based FPGA Case Study: Analyzing the Virtex-4 Data Set

- The Virtex-4 data sets had SEUs, SEFIs and some unrecognizable effect we called “micro-SEFIs”
 - The SEFIs could be separated easily
 - We usually tested at 25-1,000 SEUs/record and the records with SEFIs had 10,000-1,000,000 SEUs
- We would pre-process the data using a threshold:
 - Any record with 1,000 or fewer SEUs were SEUs and micro-SEFIs
 - Any record with 1,001 or larger SEUs were SEFIs

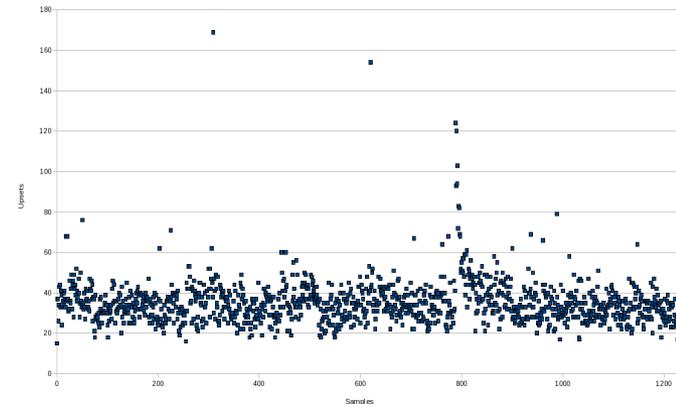


SRAM-based FPGA Case Study: Jackknifing the Virtex-4 Data Set

- Once the SEFIs were removed, then we tried to separate SEUs and micro-SEFIs
 - Micro-SEFIs looked like very large MBUs – anywhere between 50-200 SEUs would be tightly clustered physically
 - We categorized them separately, because they looked more like transients in the control logic and not SEUs
 - Micro-SEFIs were hard to separate from the SEU data and could skew the SEU data set quite a bit – had to separate them in a two-step process
- Pre-processing the Micro-SEFI separation using jackknifing:
 - Looking at windows of 10-15 records could determine if there was a record out of step with the other records based on record sizes
- The rest of the micro-SEFIs were removed using processing for “unusually large MBUs”



Before jackknifing



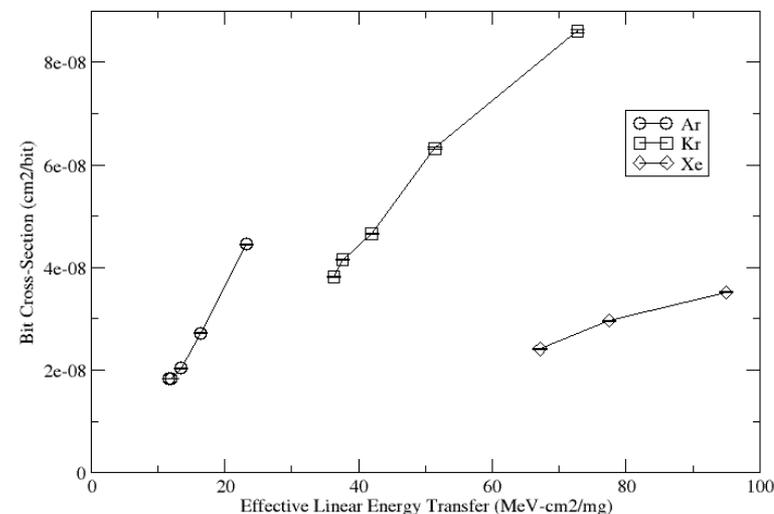
After jackknifing



SRAM-based FPGA Case Study: Angle Data

- Once you get the data plotted it should become clearer whether the angle data fits with the normal incidence data
- For SRAM FPGAs, the angle data does not fit the inverse cosine law
 - Cross sections increase at a rate greater than the cosine of the angle
 - The MBUs increase greatly
- If the angle data is not in alignment with the normal incidence data, do not combine the data sets

Heavy Ion Angular Bit Cross-Sections



H. Quinn, K. Morgan, P. Graham, J. Krone, and M. Caffrey, "Static Proton and Heavy Ion Testing of the Xilinx Virtex-5 Device," in *The Proceedings of Data Workshop for Nuclear and Space Radiation Effects Conference*, 2007, pp. 177-184.

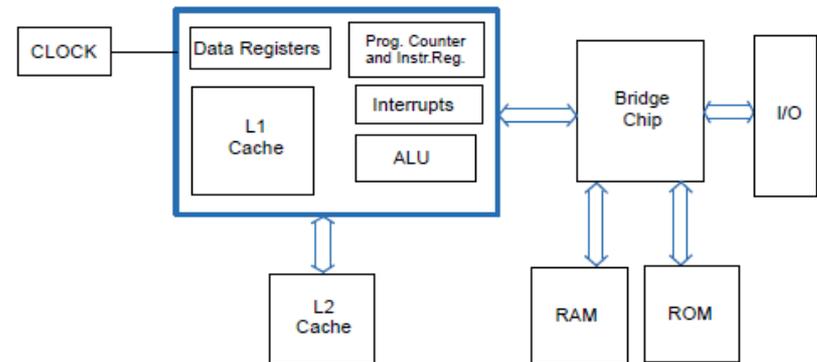


Case Study: Microprocessors



Microprocessor Case Study

- The microprocessors or microprocessor-like components of every generation can be some of the most complex components to test:
 - Multiple interfaces to input/output data,
 - Multiple levels of memory,
 - Multiple processing cores,
 - Multiple processing modes,
 - Software interferences,
 - Operating system interferences,
 - Limited understand of the internal organization, and
 - Limited observability of the internal state.



F. Irom, "Guideline for Ground Radiation Testing of Microprocessors in the Space Radiation Environment," 2008.



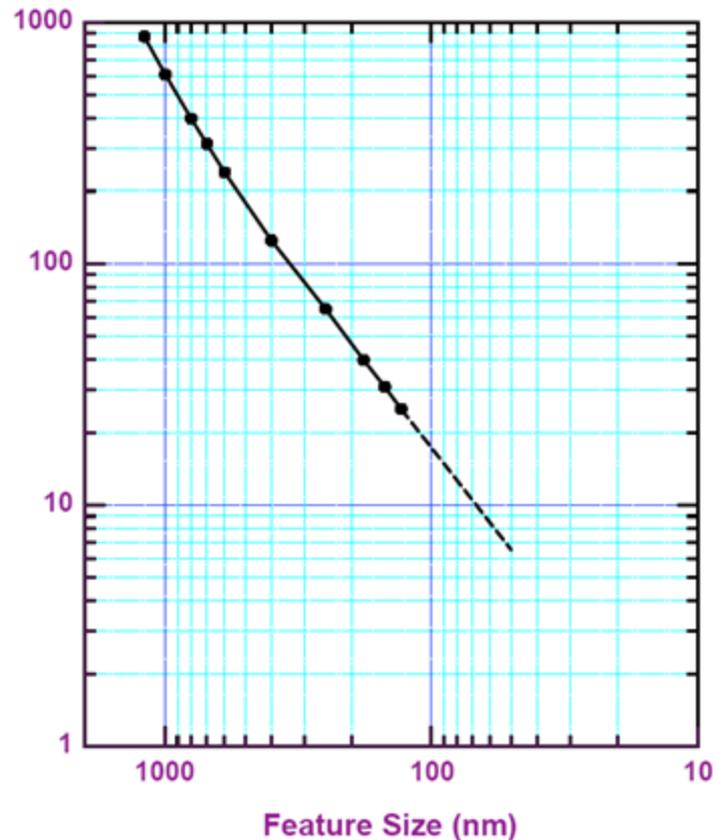
Microprocessor Case Study: Test Design

- The Jet Propulsion Laboratory (JPL) Microprocessor Test Guideline recommends testing:
 - Registers
 - Cache
 - Flight software
- The JPL SOC Test Guideline recommends testing:
 - Peripherals
 - Fault-tolerance circuitry
 - Radiation-hardened circuitry
- Microprocessors can be difficult because it is hard to piece together an understanding of radiation sensitivity from static cross-sections:
 - How does the software use the registers?
 - How does the software use the cache?
 - Which interfaces does the system use?



Microprocessor Case Study: Clock Speed

Critical Pulse Width for Unattenuated Propagation

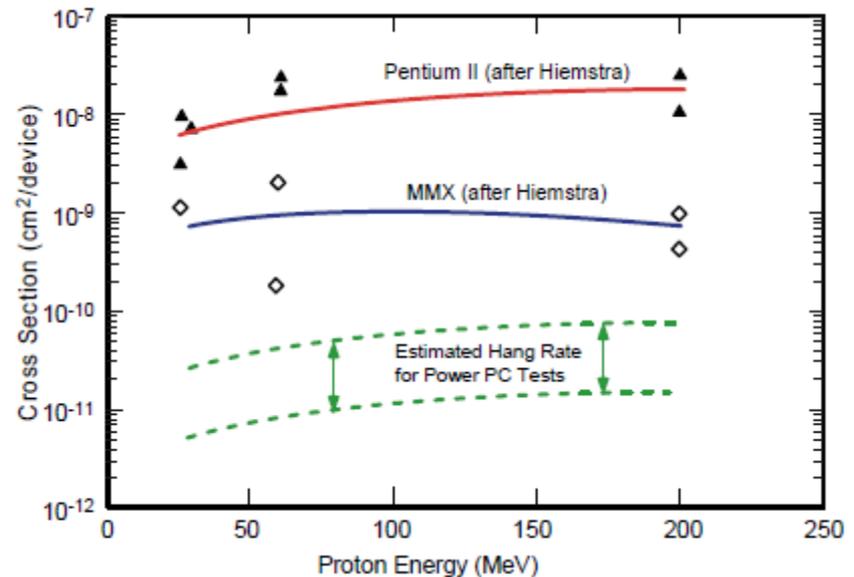


Mavis, "Single-Event Transient Phenomena: Challenges and Solutions." MRQW, 2002.



Microprocessor Case Study: Operating System Interference

- Unless you want to test using assembly code when testing, it is often hard to avoid having an operating system loaded during the test
- The classic example of the problems with testing with operating system test was completed by Heimstra and Baril on the Pentium processor using Windows NT
- “The hang rate was so high that it was not possible to determine the error rate for registers in those tests.” [JPL Microprocessor Test Guideline]
- Test the operating system that you want to fly

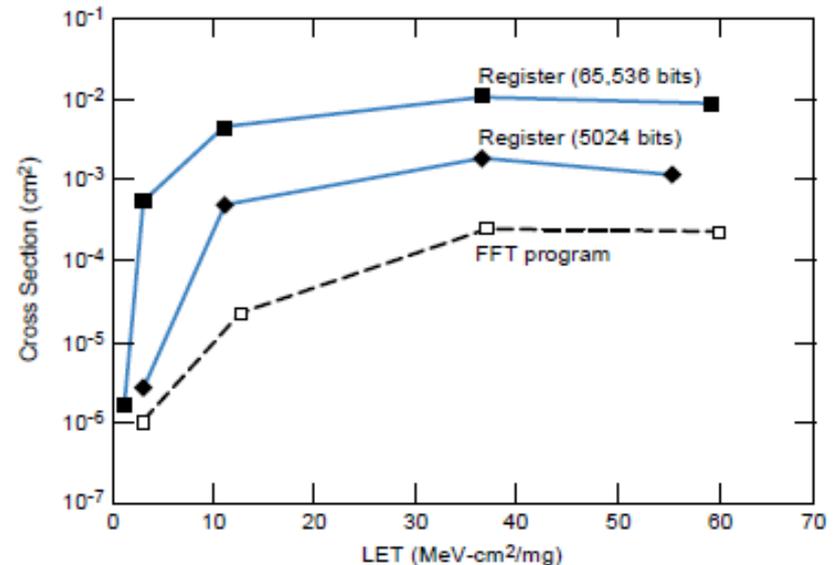


F. Irom, "Guideline for Ground Radiation Testing of Microprocessors in the Space Radiation Environment," 2008.



Microprocessor Case Study: Software Test

- The JPL Microprocessor Test Guideline will help provide an understanding of the basic understanding of the caches and the registers
 - The figure to the right shows the cross-section for the PowerPC 603E registers, cache and signal processing algorithm
 - Is the decreased sensitivity for the software caused by how the cache is used or the registers?
- Ideally, there would be a way to translate the cache and register cross-sections to a software cross-section
 - We do not have an encompassing understanding of how software uses caches and registers, except as an average case
 - I think we can get there with modeling, but we are not there yet
- Test the software that you want to fly

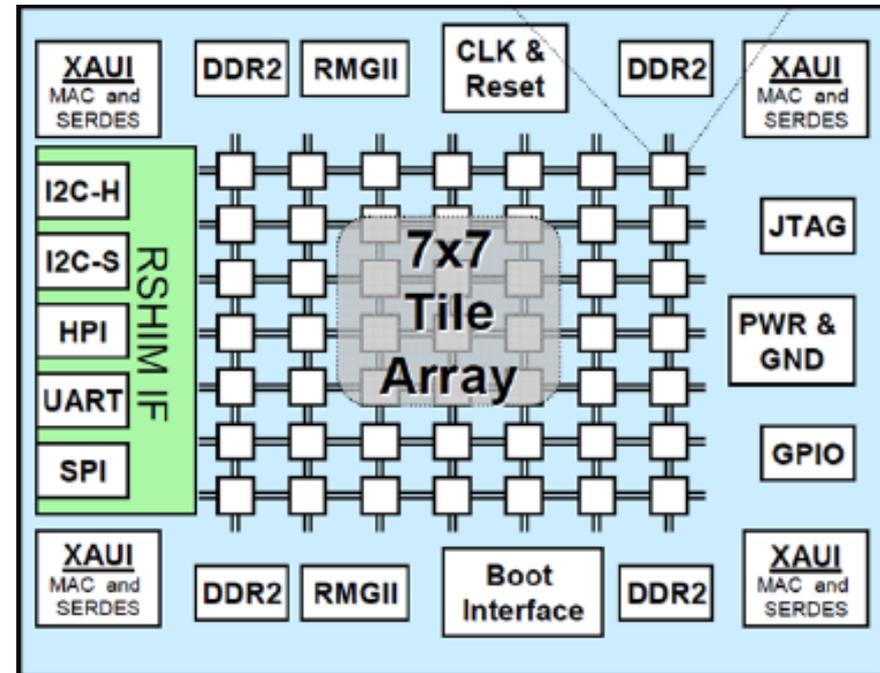


F. Irom, "Guideline for Ground Radiation Testing of Microprocessors in the Space Radiation Environment," 2008.



Microprocessor Case Study: Multi-core Microprocessors

- One method for increasing the performance of microprocessors is to increase the number of cores
- Two approaches to testing multi-core components:
 - Optimistic: independently functioning cores scales the amount of data you can collect
 - Pessimistic: How will we get all of that data onto and off of the chip? Will there be a shared failure mode?
- There is validity to both approaches

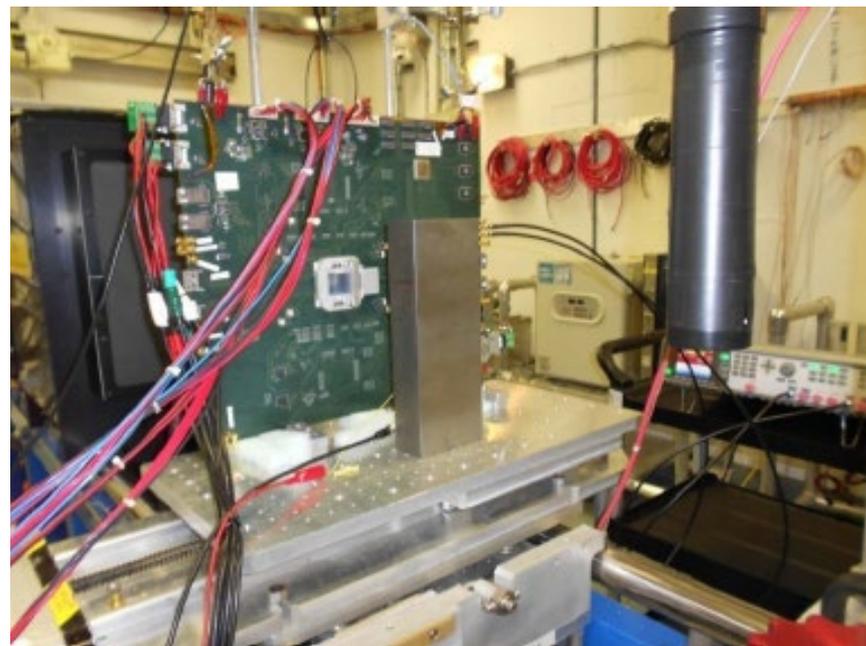


Steven M. Guertin, Brian Wie, Michael K. Plante, Antwoing Berkley, Lonnie S. Walling, and Manuel Cabanas-Holmen. SEE Test Results for Maestro Microprocessor. RADECS 2012.



Microprocessor Case Study: Test Fixture Design

- You can do FPGA-based boards, but in general not using the microprocessor as the slave
 - Most of these test fixtures depend on in-house design
- Functionally testing a microprocessor is a mixed bag:
 - Easy to get inputs in and outputs out using the standard interfaces
 - Harder to get at the internal state – must use boundary scan
 - The standard interfaces will also be in the beam



S. M. Guertin, B. Wie, M. K. Plante, A. Berkley, L. S. Walling, and M. Cabanas-Holmen, "SEE Test Results for Maestro Microprocessor," in RADECS, 2012.



Microprocessor Case Study: Boundary Scan

- Most components have a boundary scan port – the Joint Test Action Group (JTAG) standard being the most common
- We have found that many manufacturers provide extremely powerful JTAG implementations so that values in registers and caches can be read
- Using a JTAG connection between the test component and a computer will allow you to monitor the functionality of the hardware



Conclusions

- Radiation effects research will continue to evolve in the upcoming years
- In this short course we have discussed many factors in designing tests for both TID and SEE tests
 - Biasing conditions,
 - Temperature conditions,
 - Angle conditions,
 - Internal fault tolerance mechanisms,
 - Functionality conditions.
- We have also presented ideas on test fixture design, including monitoring both current consumption and functionality
- We have covered some basics of analyzing TID and SEE test data

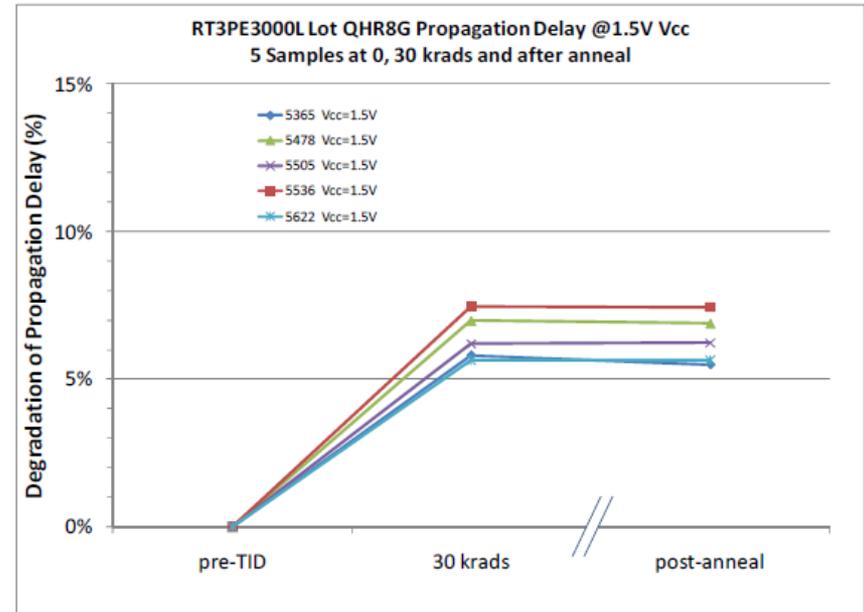


Back Up Slides



Complications with Flash-based FPGAs

- The components have the usual flash problems with TID:
 - Write/program capacity will be affected by the accumulation of dose around 20-30 Krads
 - Read/execution of the user circuit will be affected by the accumulation of dose between 20-70 Krads
- The interesting aspect is that increasing dose accumulation also affects the timing of the circuit
 - You can decrease the effect of the dose accumulation with reprogramming the FPGA...until you lose that



<http://www.actel.com/documents/11T-RT3PE3000L-LG896-QHR8G.pdf>



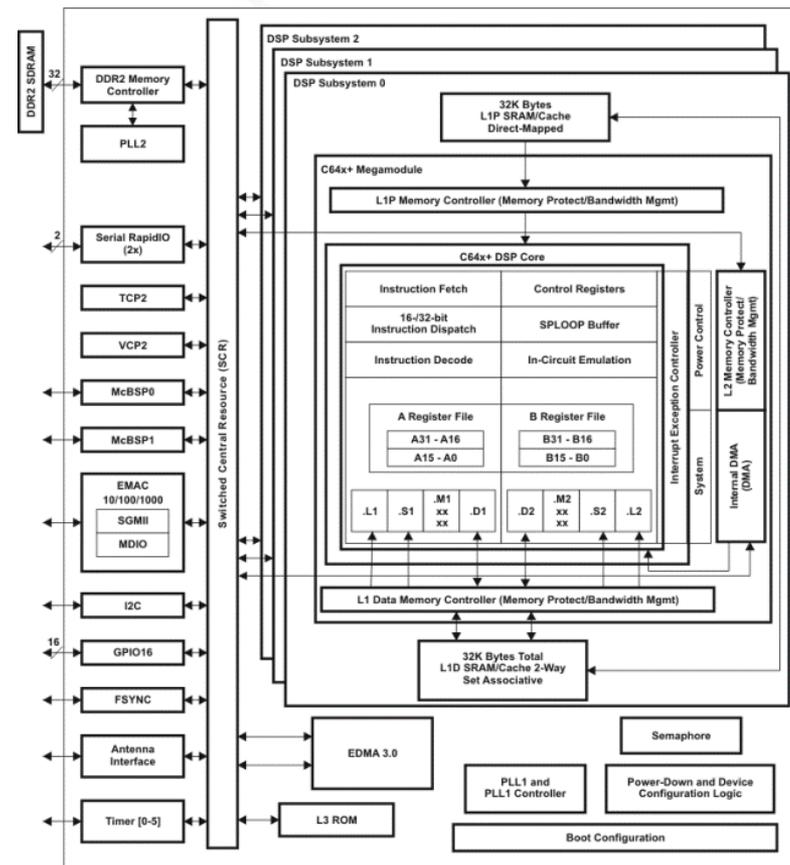
Flash-based FPGA Test Design

- There has been a lot of testing on flash-based FPGAs, but the fabric has not been dissected like SRAM-based FPGAs
- The testing has focused on:
 - SETs
 - TID-based delay
- Quote Melanie and Sana papers



Digital Signal Processors

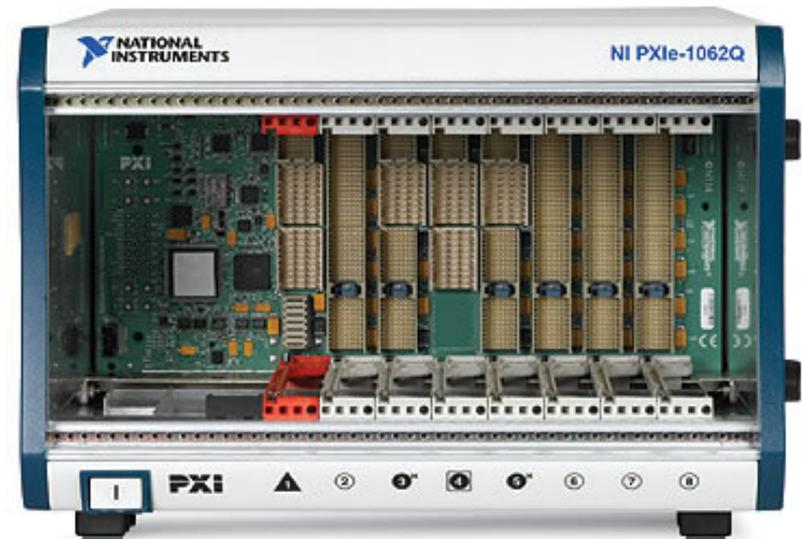
- Have a lot of similarities with microprocessors but are optimized for signal processing
- There can be more interfaces in a DSP than in a microprocessor
 - Which ones do you test?
 - There can be a lot of memory in interfaces for buffering data





National Instrument Chassis

- National Instruments Chassis are a reasonable place to start building test fixtures
- A number of different accessories can be added to the slots:
 - Power supplies
 - Oscilloscopes
 - Function generators
 - Data acquisition
- If you do not have board design capabilities readily available, then these systems can be use

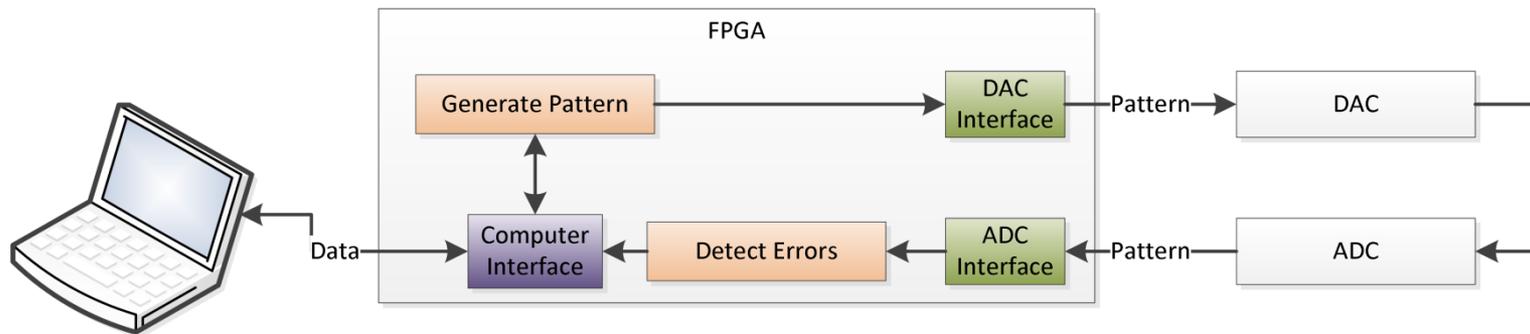


<http://sine.ni.com/nips/cds/view/p/lang/en/nid/202664>



Example of FPGA-based Test Fixtures: ADC/DAC Components

- FPGAs can be used on the digital side
- Functional or waveform generators can be useful on the analog input side.
- Detecting transients on the analog side:
 - Generally can use high-speed oscilloscopes
 - You can also put an ADC down to convert back into digital





General Thoughts on Test Design: Things That Are Nice to Have

- Automated logging
 - For when the beam is off or on
 - For logging functional failures
- Watchdogs
 - If you don't know how the SEFIs affect the system, it might be necessary to use a watchdog to reset the hardware if/when it crashes in the beam



Fit to Curves:

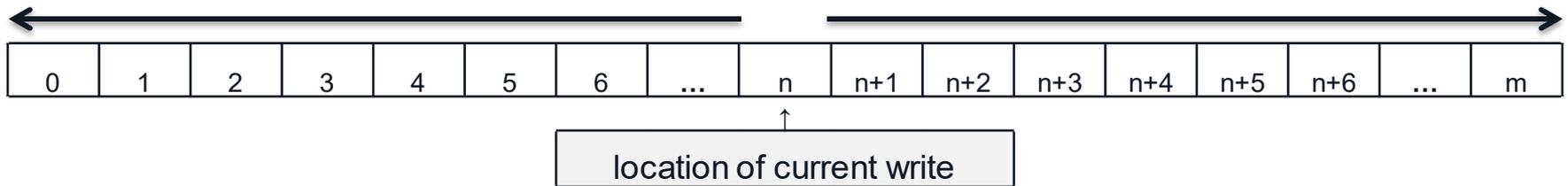
- There are a number of different ways to represent test data:
 - Weibull
 - Bendel
 - Figure of Merit
- CREME96 takes the first two, as well as Q_{crit} and a table of data points
- Curves can be fit by hand, although the least means square value for the data points will need to be minimized by hand
- Matlab will fit to a Weibull curve
- Once the data is fit to a curve, the curve and the data points can be plotted
 - By convention plots are in log-normal but that hides a lot of sins
 - Plotting is a useful method of determining visually whether you have any outliers that need to be examined



Dynamic Writes

SEUs in these locations will be read

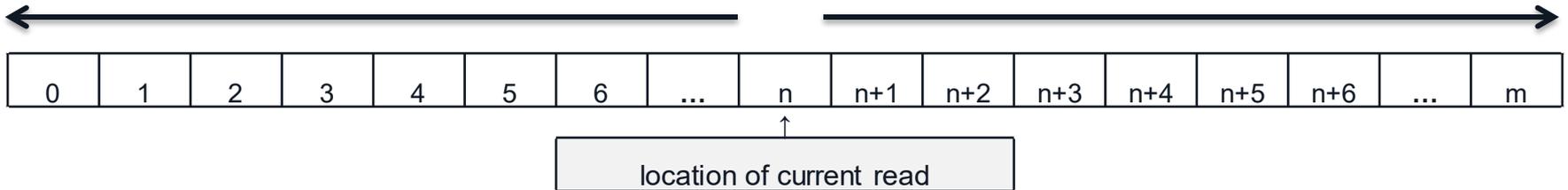
SEUs in these location will be overwritten



Dynamic Reads

SEUs in these location will be overwritten

SEUs in these locations will be read







Designing Tests:

Designing Tests for Complex Processing Elements

- Some components have complex internal organizations, which can lead to complex tests
- There can be limited understanding and visibility of significant portions of the design
 - Peripheral circuitry of memories
 - Control logic in processing elements
- We will study case studies on SRAM and SRAM-based FPGAs