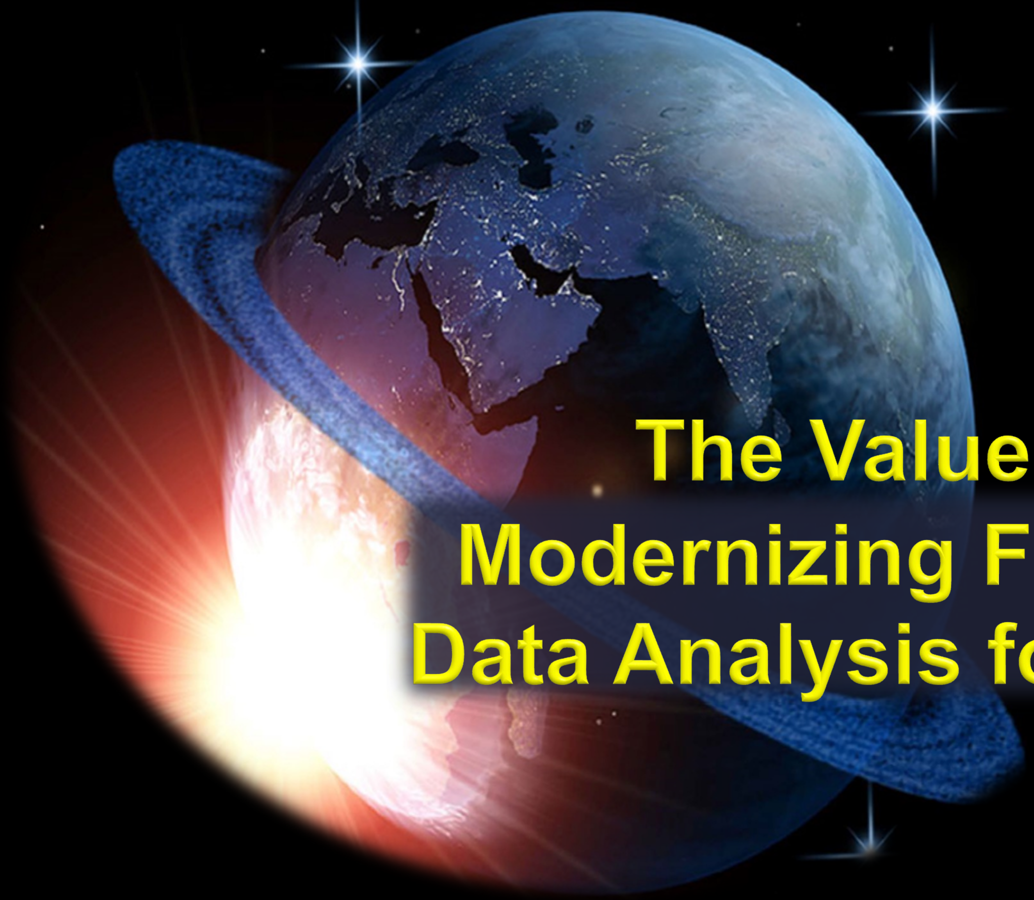


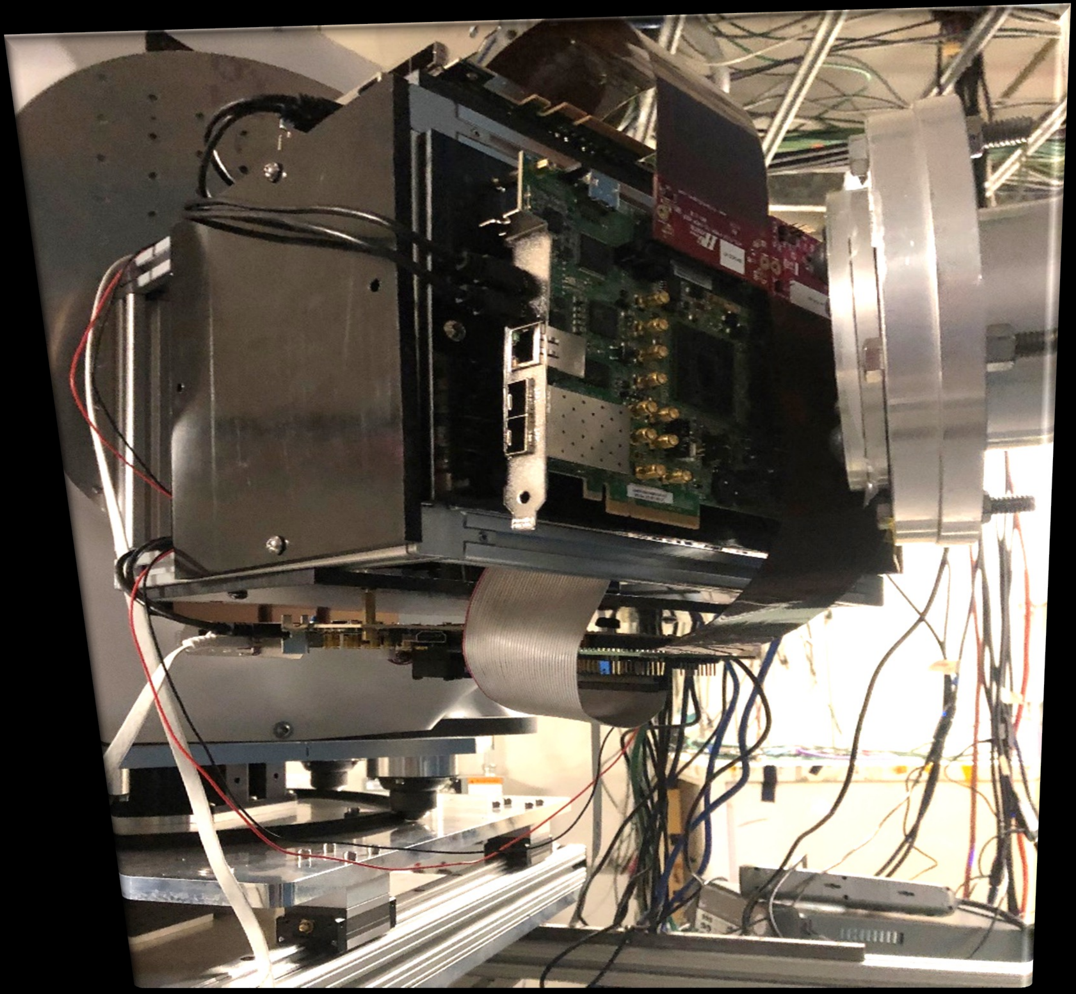
The Value of “Test-As-You-Fly”: Modernizing FPGA Experimentation And Data Analysis for Critical Space Missions "

Prepared by: Melanie Berg Space R^3 LLC
www.SpaceR2.com

SPACE R^3 | 979 E. Whitmire Dr. Melbourne FL 32935 | 443-904-6510

Space R^3 LLC Services: Design

- ASIC and FPGA Design
  - Over 34 years of design experience
  - Space and Terrestrial applications
  - Custom design and verification
  - IP development
  - Design Reviews
- Reliability
  - Test system design
  - At-speed custom
  - Emulation



World leader in design mitigation:

- Development
- Analysis
- Implementation

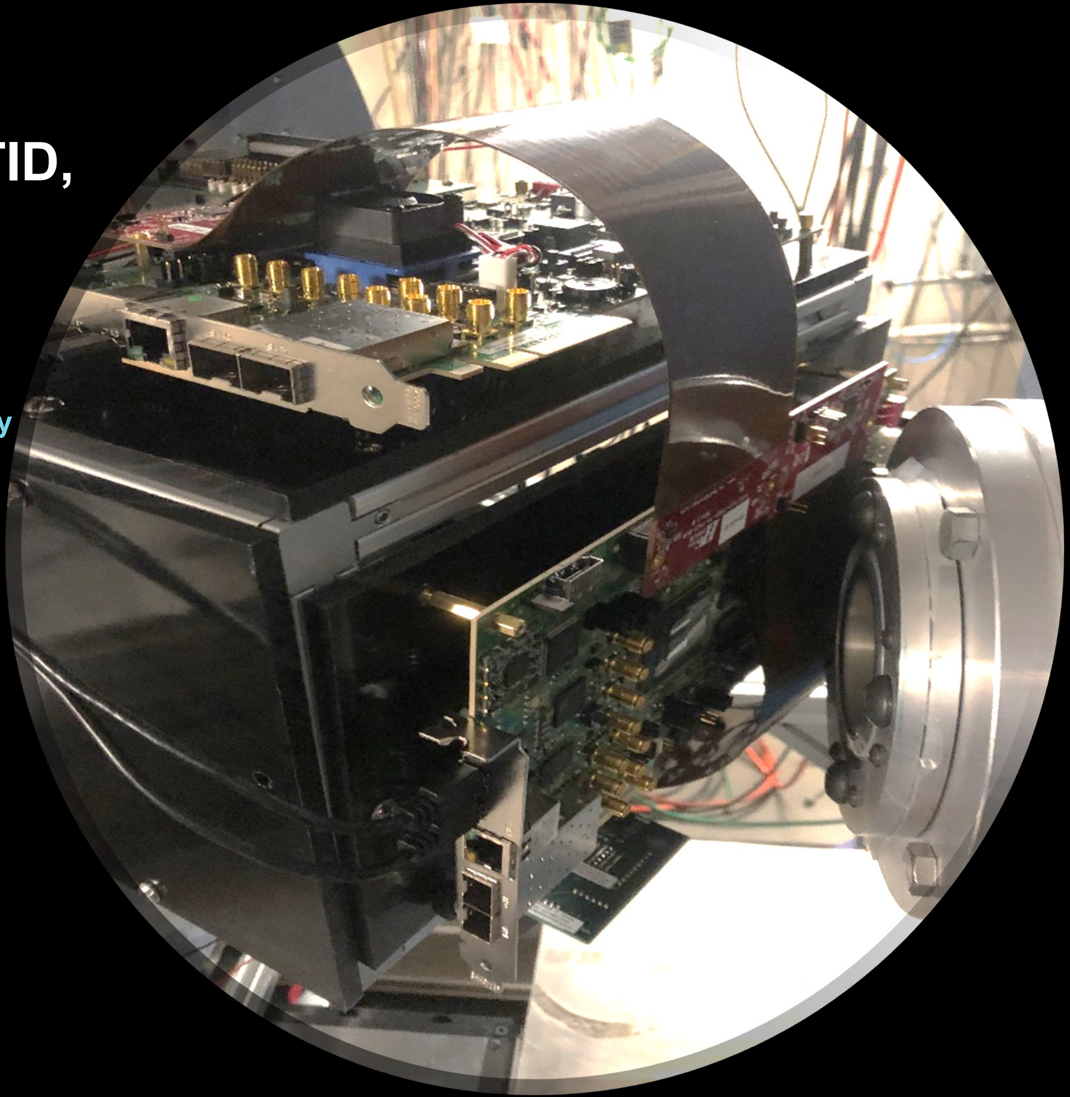
Space R^3 LLC Services: Radiation

- 20 years experience in radiation test and analysis
- Test-system development
- SEE (heavy-ion, proton, and Neutron), TID, Prompt Dose
  - FPGA
  - Custom ASICs
  - Memories
  - ADC
- Data Analysis

SEE: single event effects
TID: Total Ionizing Dose
FPGA: Field Programmable Gate Array
ADC: Analog to Digital Converter

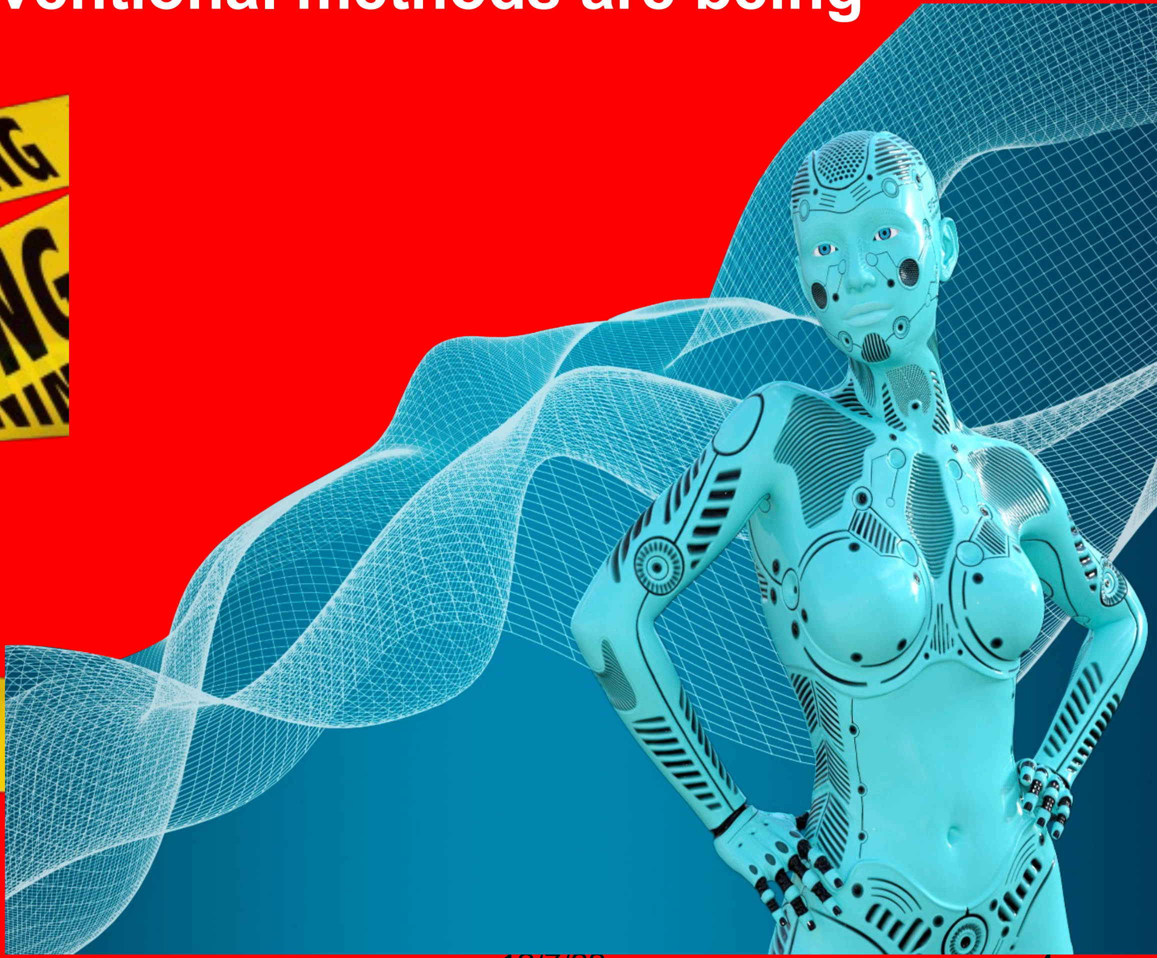
Formerly Space R^2 LLC

Providing services with RadTek Space since 2017



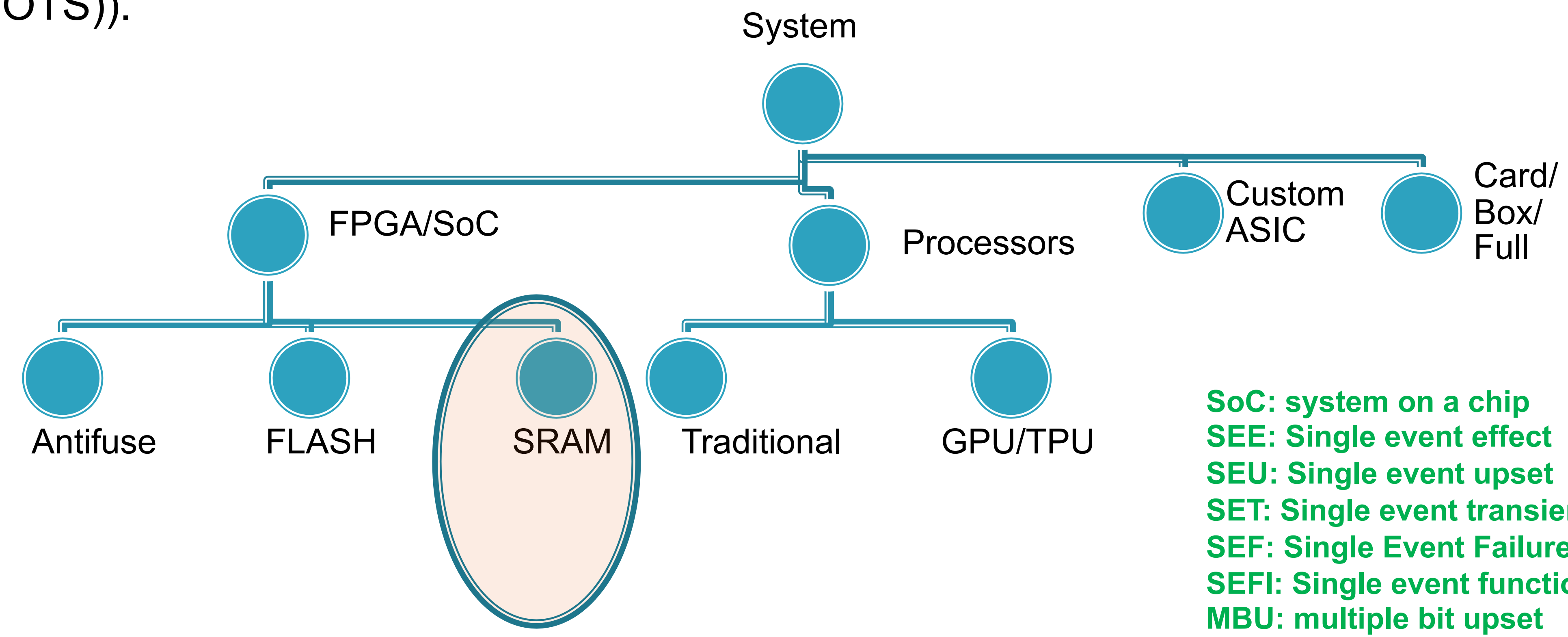
Beware…

- This presentation contains controversial information.
- Considered controversial because conventional methods are being challeged



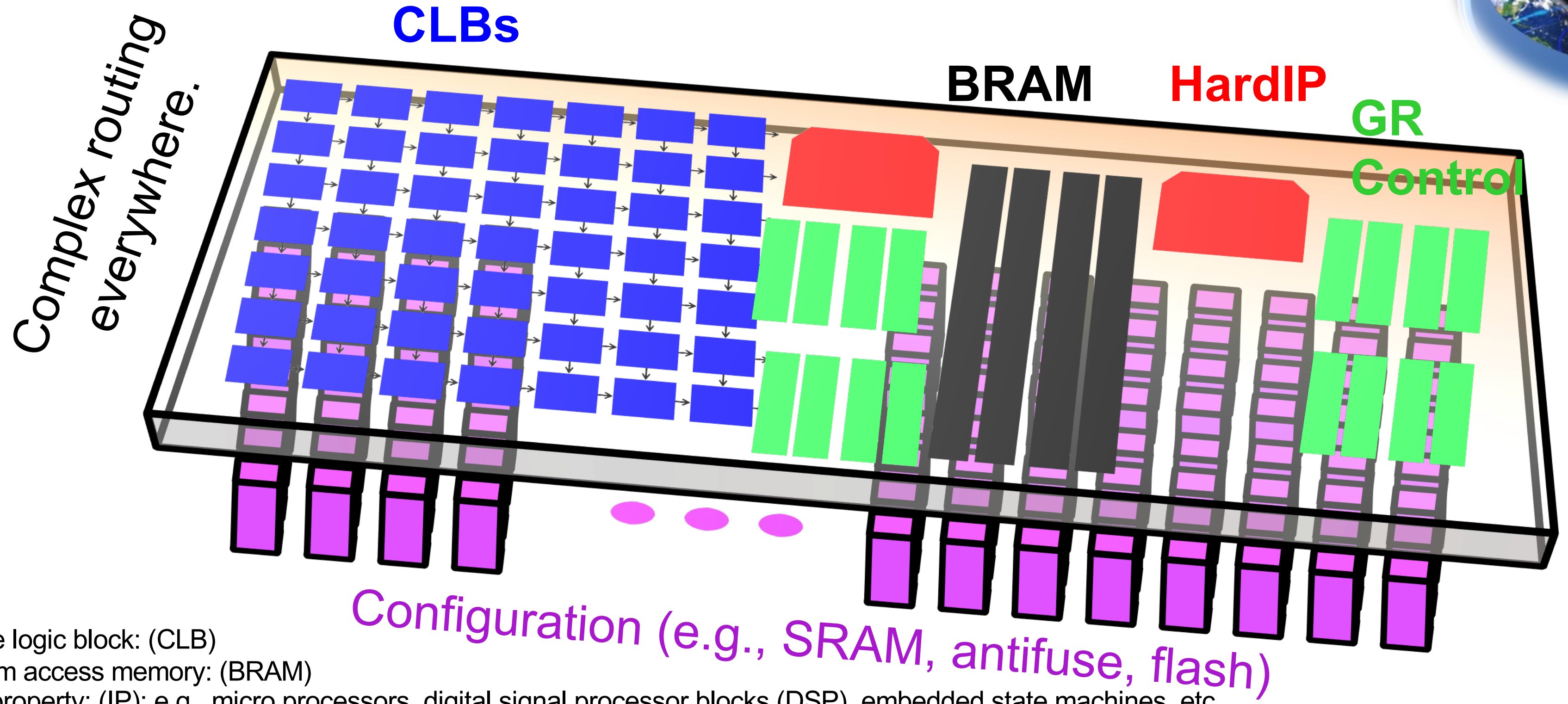
Problem Statement

- Conventional single event test and analysis methods were not developed to handle complex systems.
- When following the conventional approach, in many cases, **system-level error rate calculations** inaccurately predict failure for unhardened SoCs (e.g., commercial-off-the-shelf (COTS)).



SoC: system on a chip
SEE: Single event effect
SEU: Single event upset
SET: Single event transient
SEF: Single Event Failure
SEFI: Single event functional interrupt
MBU: multiple bit upset

FPGA SEU Cross Section Model


CLBs
BRAM
HardIP
GR
Control
Complex routing everywhere.
Configuration (e.g., SRAM, antifuse, flash)

Configurable logic block: (CLB)
Block random access memory: (BRAM)
Intellectual property: (IP); e.g., micro processors, digital signal processor blocks (DSP), embedded state machines, etc.
Global Routes: (GR)
Analog circuits

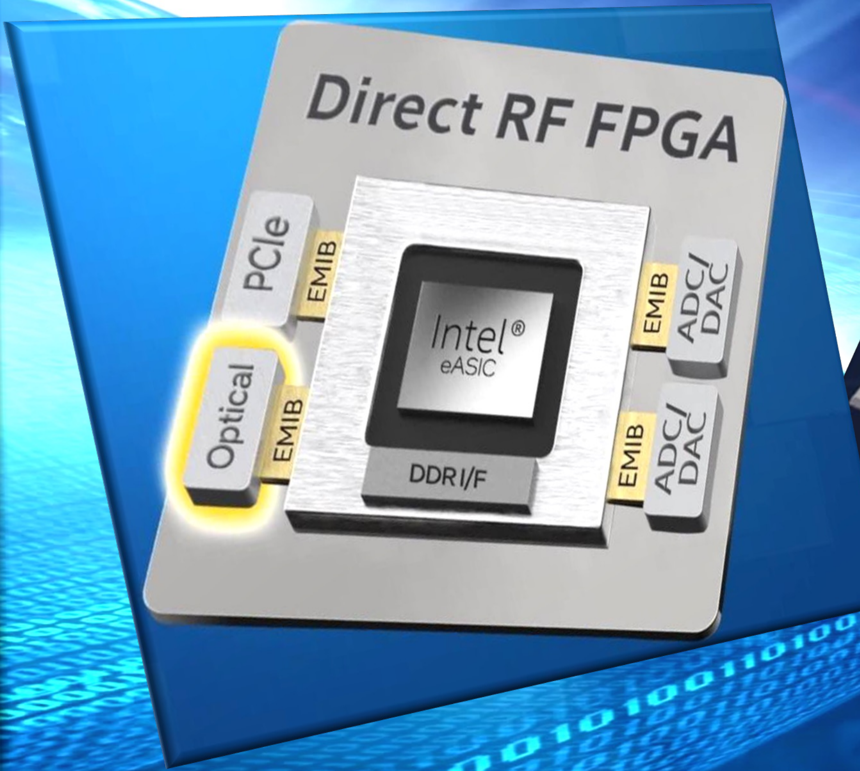
SEU Cross sections for a mapped design (σ_{SEF}) are based on the FPGA's internal elements and the mapped design's topology.

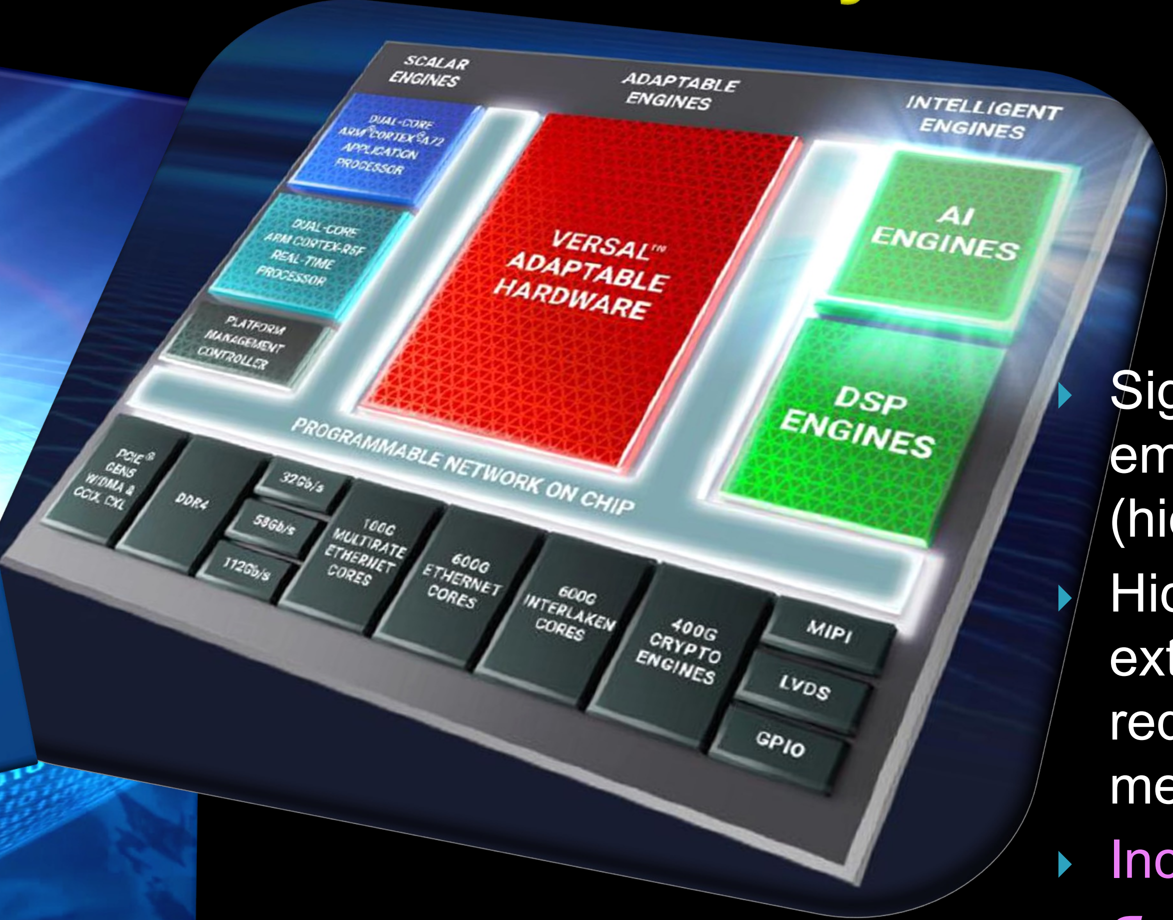
$$\sigma_{SEF} = f\left(\sigma_{configuration}, \sigma_{BRAM}, \sigma_{functionalLogic}, \sigma_{HiddenLogic}\right)$$

For research purposes, there are established testing techniques to study various FPGA elements

Melanie Berg et. al, "FPGA SEU Radiation Test Guidelines:" https://nepp.nasa.gov/files/23779/fpga_radiation_test_guidelines_2012.pdf

New Generation System on a Chip (SoC)… New Challenges for SEE Test and Analysis




SCALAR ENGINES
ADAPTABLE ENGINES
INTELLIGENT ENGINES
DUAL-CORE ARM CORTEX-A72 APPLICATION PROCESSOR
DUAL-CORE ARM CORTEX-R5F REAL-TIME PROCESSOR
PLATFORM MANAGEMENT CONTROLLER
VERSAL™ ADAPTABLE HARDWARE
AI ENGINES
DSP ENGINES
PROGRAMMABLE NETWORK ON CHIP
PCIE® GEN5 W/DMA & CCIX, CXL
DDR4
32Gb/s
58Gb/s
112Gb/s
100G MULTIRATE ETHERNET CORES
600G ETHERNET CORES
600G INTERLAKEN CORES
400G CRYPTO ENGINES
MIPI
LVDS
GPIO

- Significant amount of embedded circuitry (hidden logic)
- Hidden circuits are extremely complex and require complex test methods.
- Increased focus on $\sigma_{HiddenLogic}$

$$\sigma_{SEF} = f\left(\sigma_{configuration}, \sigma_{BRAM}, \sigma_{functionalLogic}, \sigma_{HiddenLogic}\right)$$

Investigating Failure Modes: Radiation Testing and SEE Cross Sections

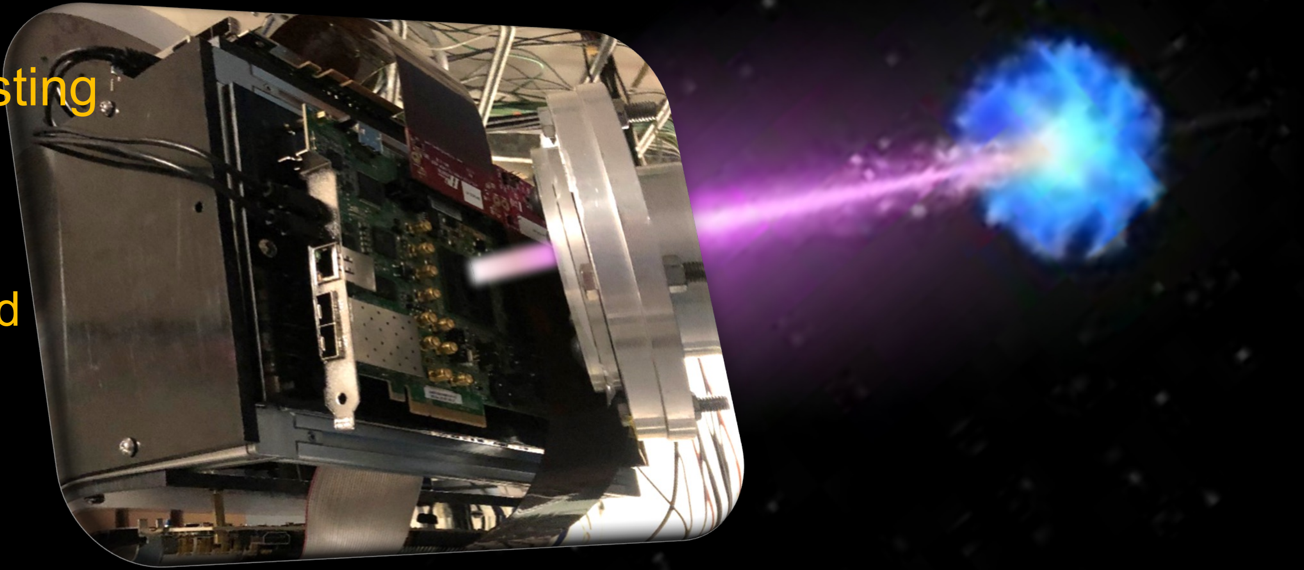
System failures due to SEEs are second order:

- Probability that a transistor will change state, and
- Probability the SEU or SET will cause system malfunction.

Cross sections are metrics derived from beam testing

$$\sigma_{SEU}(LET) = \frac{\#events}{\#ions/_{cm^2}} = \frac{\#events}{Fluence}$$

σ_{seu}s are empirical data that are calculated per selected LET values (particle spectrum).



Space R^3 testing at LBNL 88in Cyclotron

LBNL: Lawrence Berkeley National Laboratory

Terminology:

- Flux: Particles/(sec-cm^2)
- Fluence: Particles/cm^2
- Linear energy transfer (LET)

Systems, SEE Cross-Sections, and Error Rates

$$\sigma_{SEU} = \frac{\#events}{\#ions/_{cm^2}}$$

Number of events that will occur in an area per ion exposure

- Methods for calculating Single Event error rates rely on cross-sections.
  - Traditionally σ_{SEU}(LET) are metrics that describe a sensitive area (SEE susceptibility) of a device.
  - Using the RPP method with σ_{SEU}(LET), x,y,z parameters are determined (z creates the sensitive volume), and error rates are calculated.
  - The concept of sensitive volume works well for transistor or bit-level components error rate calculations.
  - For systems, conventionally, low-level component cross-sections (bit-level) are obtained and are usually extrapolated to characterize system SEE behavior.
- **Problem**: low-level metrics do not consider circuit topology. In this case, systems are poorly characterized under the conventional cross-section metric definition.
- **Approach to Solution:** We can start with classical system-level failure rate theory and its established probability models.

Rethinking System Characterization

Focus on Top-Down analyses: Test-as-you-fly

DFF: Flip-flop

For systems, think about complex arrangements of moving parts.

Extrapolation of low-level parts to characterize a system is impossible without putting in context how the parts move together.

Instead of only charactering low- level mechanisms of error/failure (transistors and DFFs), focus on characterizing system failure with context and information (topology).

i.e., characterize by system failures not by low-level component upsets. Test top-down … or… test-as-you-fly:

$$Single\ Event\ Failure = \sigma_{SEF} = \frac{\#events}{Fluence}$$

Reimagine Cross Sections as Probabilities

$Single\ Event\ Failure = \sigma_{SEF} = \frac{\#events}{Fluence}$ Failure Rate in the fluence domain

For system analyses: Step away from the conventional methods of cross-sections representing sensitive areas and the RPP method.

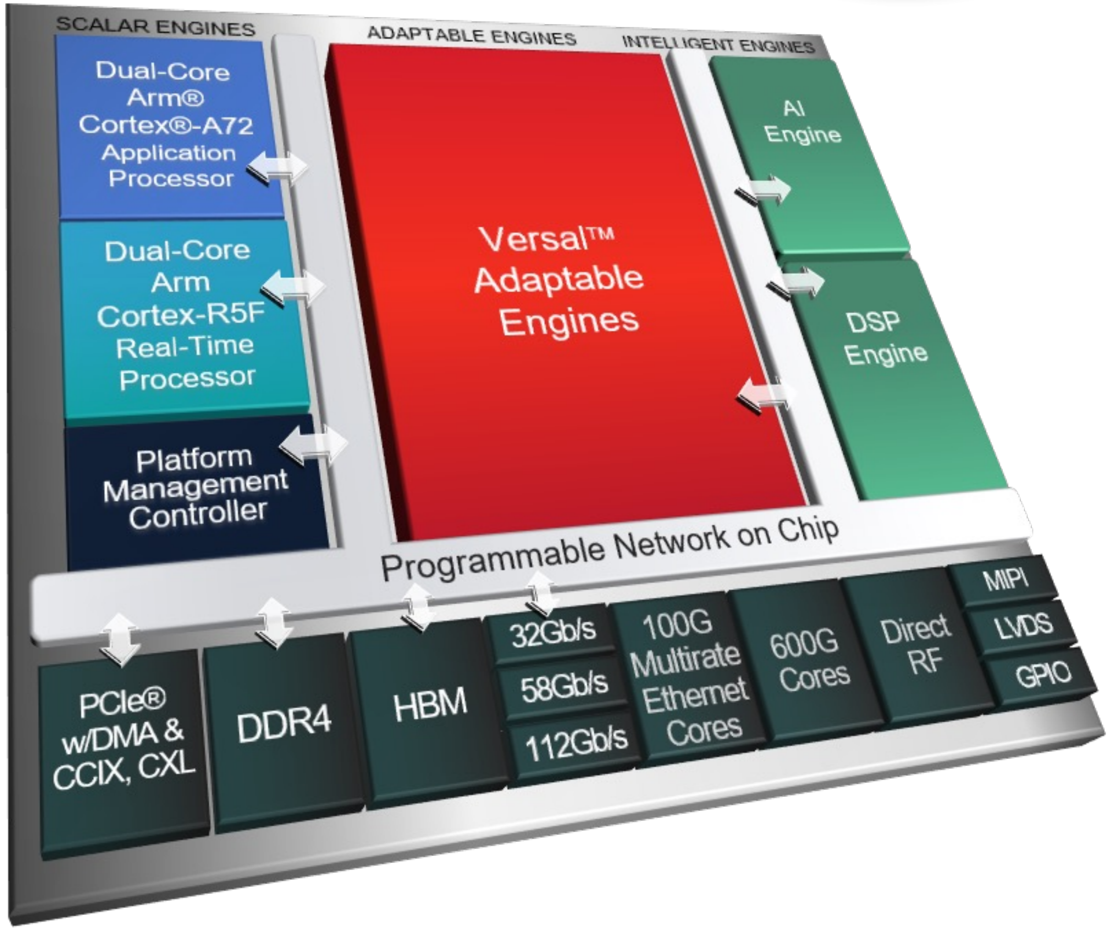
Redefine the cross-section metric to be a probability.

The probability an event will occur when the target is subjected to a given number of particles (per area).

σ_{SEF} is now a rate. However, the rate is in the fluence domain not the time domain.

In Other Words…Modernize to Accommodate Complexity

- Shift register data (the conventional golden metric) is insignificant towards the characterization of an SoC.
- Yet, shift register tests are good as a first look into device sensitivities.
- FPGA configuration memory testing is good to have, however, it will not cover the overwhelming amount of embedded logic.
- The SoC with its network on chip (NoC) requires complex test and analysis.
- Time to adapt from system-level classical reliability theory.
- **Controversial because modernization defies conventional methods.**


SCALAR ENGINES
ADAPTABLE ENGINES
INTELLIGENT ENGINES
Dual-Core Arm® Cortex®-A72 Application Processor
Dual-Core Arm Cortex-R5F Real-Time Processor
Platform Management Controller
Versal™ Adaptable Engines
AI Engine
DSP Engine
Programmable Network on Chip
PCIe® w/DMA & CCIX, CXL
DDR4
HBM
32Gb/s
58Gb/s
112Gb/s
100G Multirate Ethernet Cores
600G Cores
Direct RF
MIPI
LVDS
GPIO

Single Event Effects And The Binomial Distribution

Trial → Event→ Effect (Response)

- Each ion can either cause an event or not:
  - Binomial distribution… over multiple Bernoulli trials … each ion is an independent random trial with two (2) possible outcomes
  - Trial outcomes:
    - event (1) or
    - no event (0)
- For this definition, cross-sections can never be greater than 1.
- Law of large numbers states that these binomial experiments can be characterized by Poisson distributions.
- For systems, there will be times when the exponential distribution is a better model. The exponential distribution is a special case of the Poisson… P(X=0)

Flipping a coin is the most common example of a binomial experiment

- Just like each coin toss, each particle is a Bernoulli trial
- An Event is an upset/failure

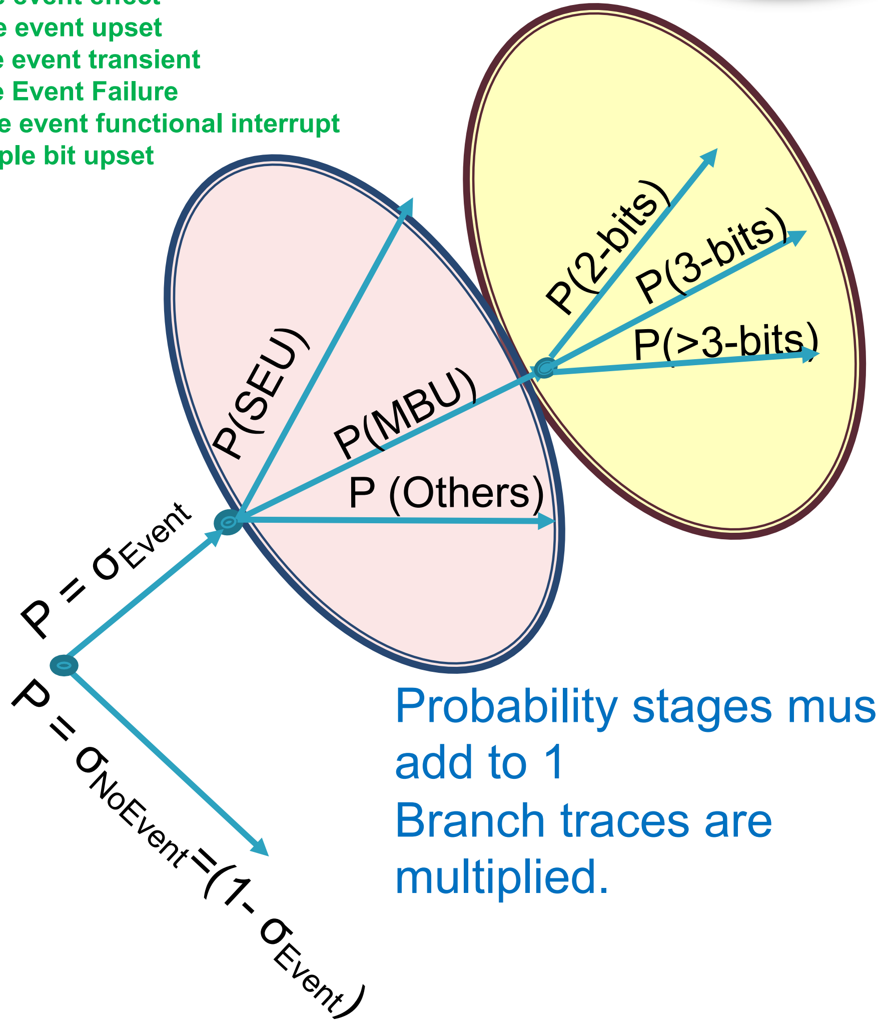
$$\sigma_{SEF} = \frac{\#events}{\#ions/_{cm^2}} < 1$$

Makes sense if we are redefining a cross section as a probability

Separating Event Datasets and Measuring/Characterizing Details with Probability Trees

- **Stage 1: Cross-Sections describe #events per ion.**
  - Probability (P) of an Event model.
  - Cross section (σ) probabilities follow the law of large numbers (each trial is an ion/cm^2).
- **Stage 2: Event-effects are separated:**
  - SEU, MBU, Other
  - Each has a probability of occurrence.
- **Stage 3: MBU event-effects are further broken down (for detailed analysis):**
  - Probabilities are percentages of occurrences
  - Example contains 3 MBU groups; yet user can divide as intended
  - SEFI probabilities can be analyzed in a similar fashion

SoC: system on a chip
SEE: Single event effect
SEU: Single event upset
SET: Single event transient
SEF: Single Event Failure
SEFI: Single event functional interrupt
MBU: multiple bit upset

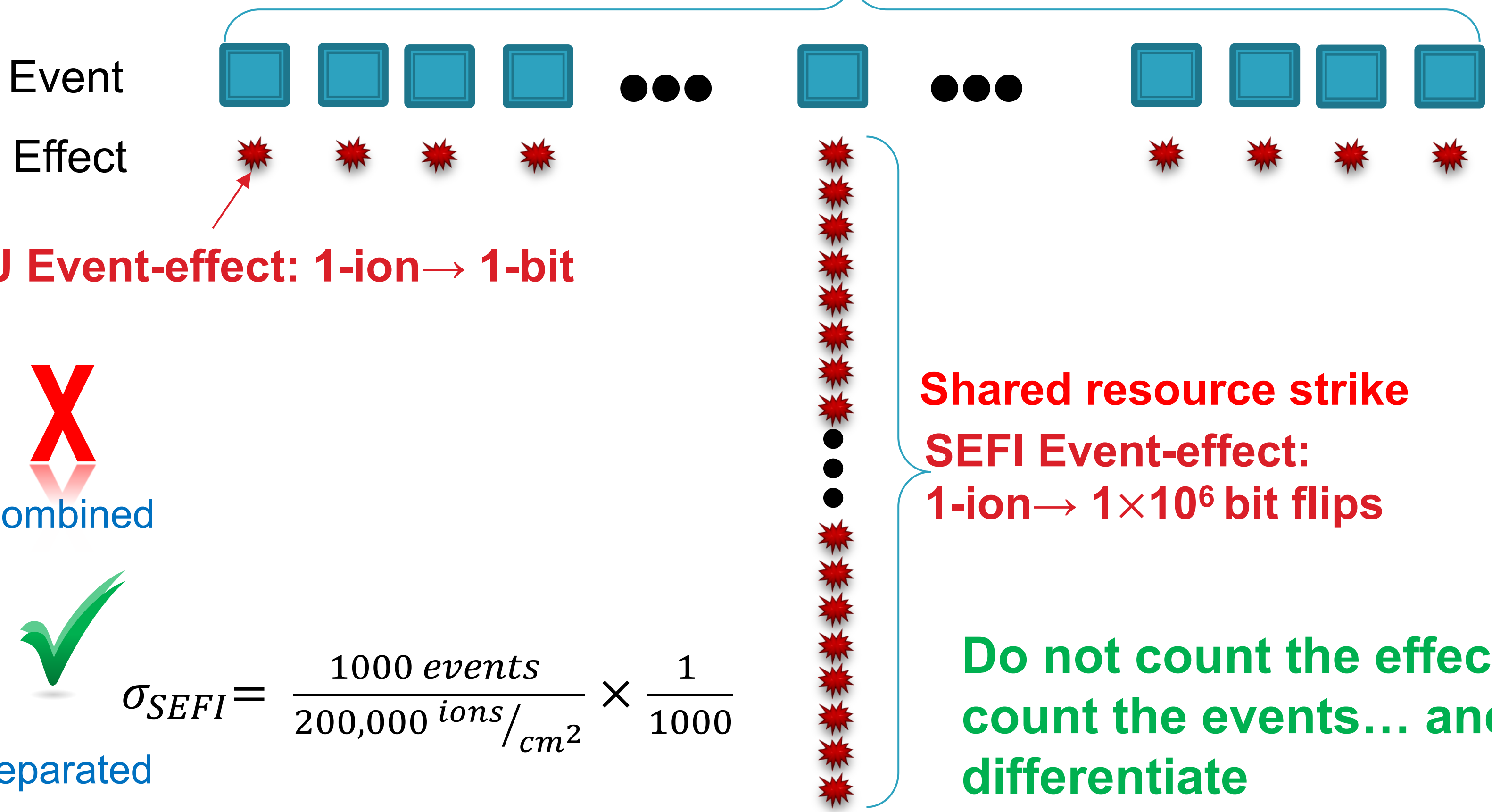


Probability stages must add to 1
Branch traces are multiplied.

Event versus Effects Example: Memory Cross-Sections: SEFIs and SEUs

Trials = fluence: 200,000 ions/cm²

1000 Events out of 200,000 ions



SEU Event-effect: 1-ion→ 1-bit

Shared resource strike

SEFI Event-effect: 1-ion→ 1×10^6 bit flips

Careful:

$$\sigma = \frac{1{,}000{,}999\ events}{200{,}000\ {}^{ions}/_{cm^2}}$$

✗ Combined

or

$$\sigma_{SEU} = \frac{1000\ events}{200{,}000\ {}^{ions}/_{cm^2}} \times \frac{999}{1000}\ ; \quad \sigma_{SEFI} = \frac{1000\ events}{200{,}000\ {}^{ions}/_{cm^2}} \times \frac{1}{1000}$$

✓ Separated

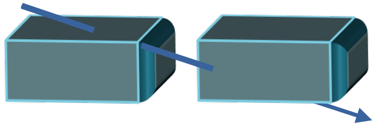
Do not count the effects, count the events… and differentiate

Example: Memory Cross-Sections:

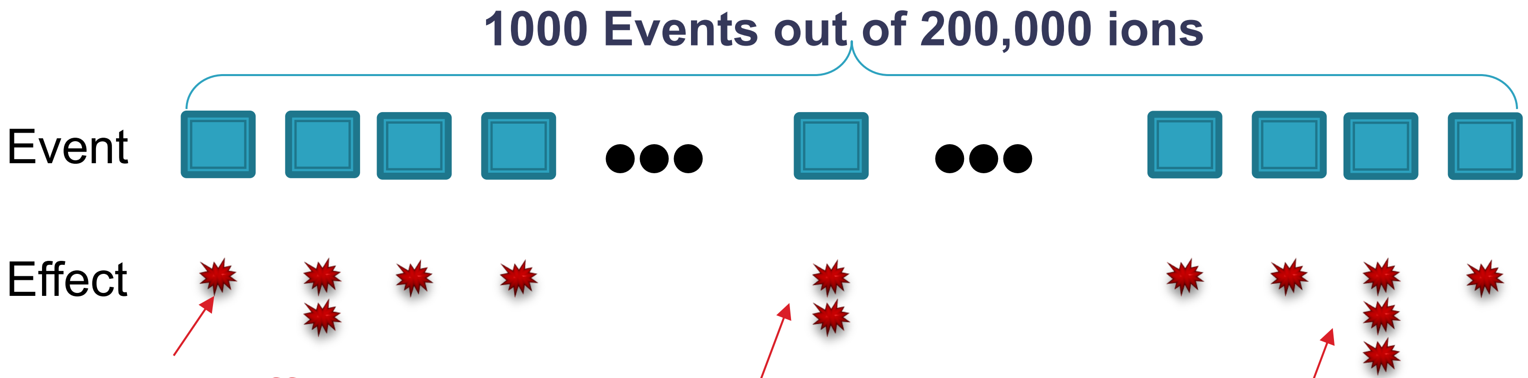
MBUs and SEUs

Example: Trials = fluence: 200,000 ions/cm^2

Ion strikes more than one bit cell



- Alert: example does not include P(bits>3) events.
- Assumes P(bits>3) for simplicity



SEU Event-effect : 1-ion→ 1-bit

MBU Event-effect : 1-ion→ 2-bits

Event-effect : 1-ion→ 3-bits

$$\sigma_{SEU} = \frac{997\ events}{200{,}000\ ^{ions}/_{cm^2}}\ ;\ \sigma_{MBU} = \frac{3\ events}{200{,}000\ ^{ions}/_{cm^2}}$$

Additional Details

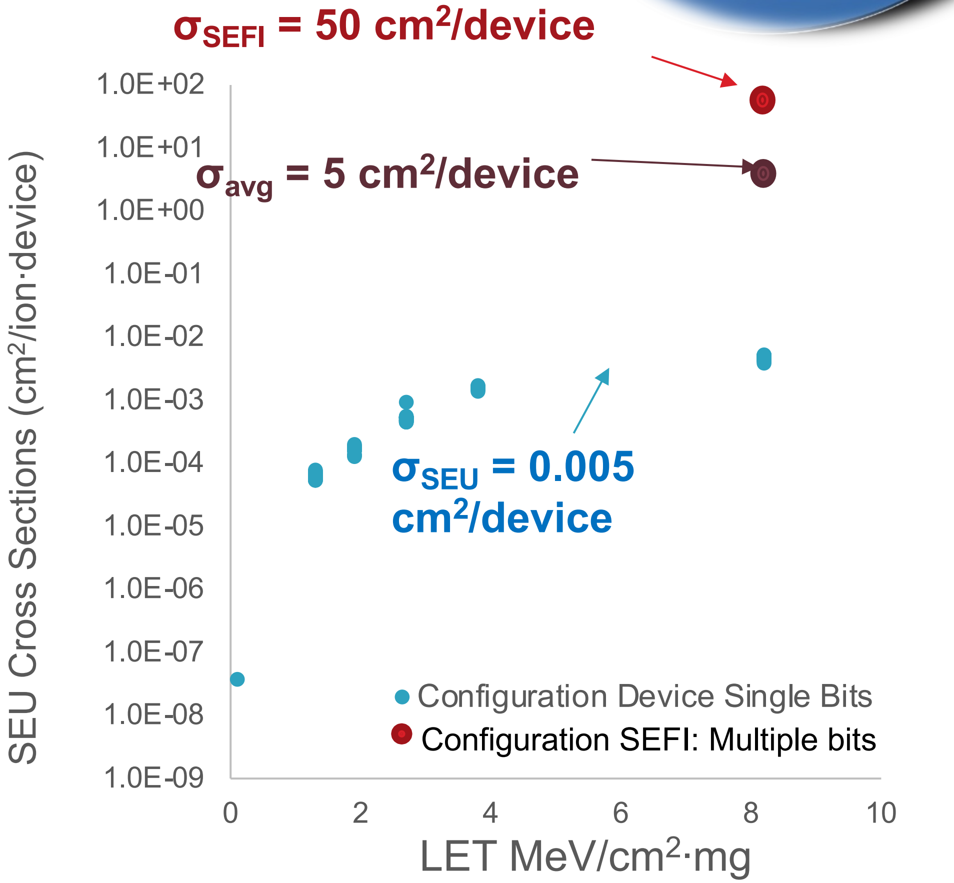
MBU Weighted bit analysis

$$\sigma_{MBU(2-bits)} \approx \frac{3\ events}{200{,}000\ ^{ions}/_{cm^2}} \boxed{\times\frac{2}{3}} \qquad \sigma_{MBU(3-bits)} \approx \frac{3\ events}{200{,}000\ ^{ions}/_{cm^2}} \boxed{\times\frac{1}{3}}$$

Problem: Correctly Separating SEU and SEFIs for Error Rate Analysis

Example: FPGA Configuration memory tests

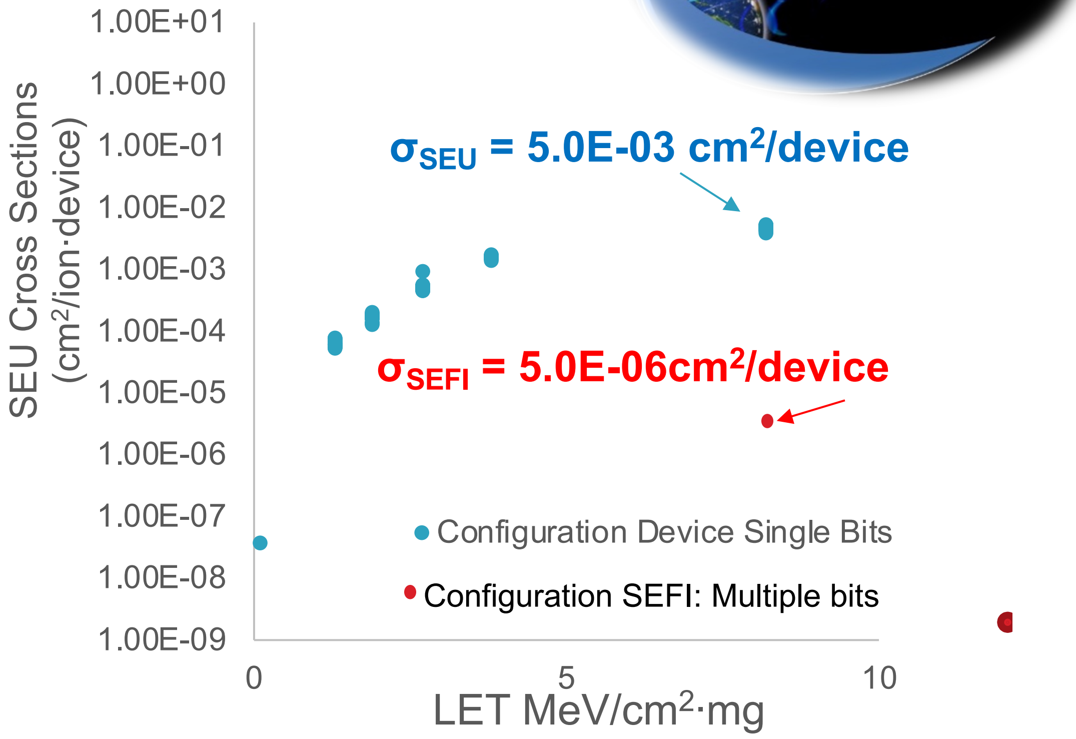
- LET = 8.2 MeV·cm^2/mg: 1 out 200,000 ions/cm^2 is a SEFI.
- **9 tests:** all events are single bit flips (no SEFI):
  - fluence per test = 20,000 (ions/cm^2)
  - #events-effects per test ≈ 100 SEUs
  - Conventional **$\sigma \approx$ 0.005 cm^2/device**
- **1 test:** SEFI causes 1,000,000 configuration bits to flip state:
  - fluence = 20,000 (ions/cm^2)
  - #events-effects ≈ 1,000,999 (SEU+SEFI 1,000,999 bit-flips)
  - Incorrect analysis: for this experiment **σ = 50**
- Conventional average (across all 200,000 ions/cm^2 – single event bit count and SEFI bit count) average **$\sigma \approx$ 5.045** cm^2/device
- SEFI event was not separated from SEUs and caused the average to jump. **Wrong calculation.**



Note: reemphasize how total fluence of all 10 tests = 200,000 (ions/cm^2)

Method: Separating Event Datasets (SEU versus SEFI)

- **First separate the SEUs and the SEFIs**
- For that 1 test: SEFI causes 1,000,000 configuration bits to flip state:
  - fluence = 20,000 ions/cm^2
  - #events-effects ≈ 1,000,999 bit-flips
  - **#events = 1 (only 1 SEFI event occurred)**
  - **Use the total fluence (across all runs) = 200,000 (not 20,000)**
  - SEFI is identified:
    - σ_{SEFI} = **1/200,000 (cm^2/(fluence·device))**
    - SEU and SEFI datasets are separated
  - **Note: Additional tests or higher fluences per test are required for the user to better measure SEFI effects (e.g.,how many bit's are affected during a SEFI)**

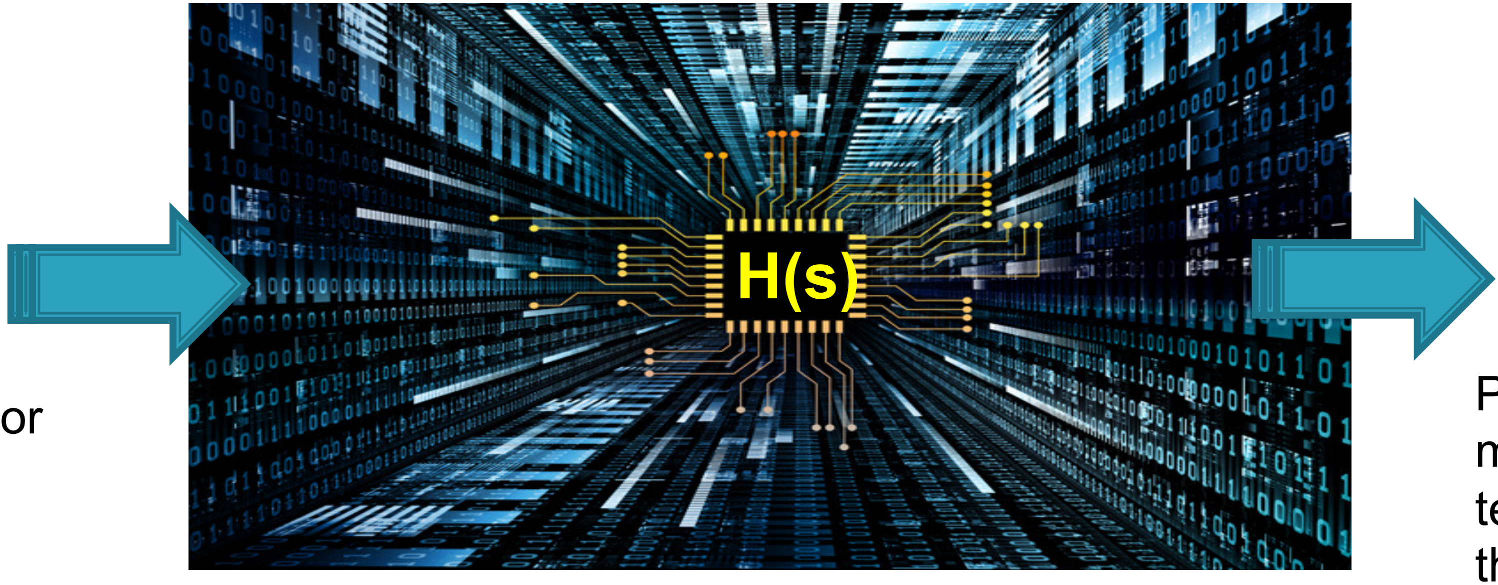


$$\sigma_{upset} = \sigma_{SEU} + \sigma_{SEFI}$$

Now cross sections are separated and weighted accordingly.

New cross sections show that the probability of getting a SEFI is decades lower than a bit-flip.

Modeling System-Level Susceptibilities as They Pertain to Empirical Cross-Sections



Raw Probability
(failure rate)

Probability a transistor is affected (lowest level of upset)

Measured Probability
(failure rate)

Probability the system malfunctions AND your test can observe/report the failure

Be careful… your test system can greatly impact the quality of your data.

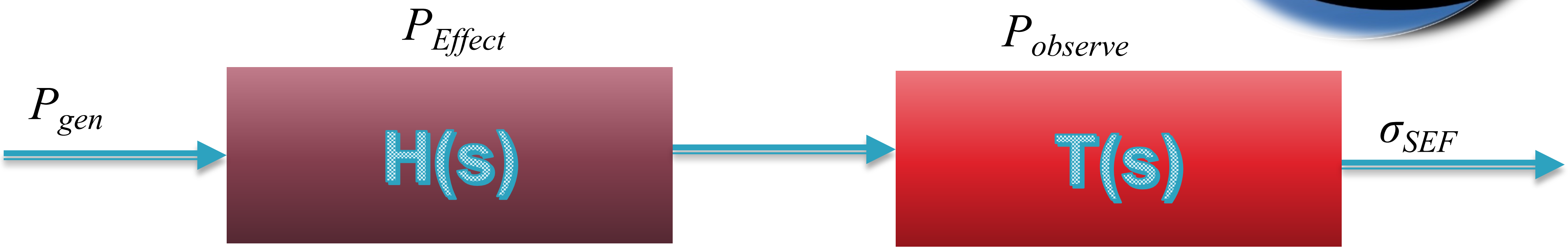
Modeling Empirical Cross-Sections from Beam Experiments: Quality of Test

$$\sigma_{SEF} = Probability\ of\ recording\ upset = \ P_{upset} \ = P_{gen} \times P_{Effect} \times P_{observe}$$

Empirical cross sections are not pure:

- P_{gen} **(Transistor/DFF):**
  - Probability that an ion strike will generate an SET/SEU in a transistor/DFF.
  - This is your lowest level upset: physics, sensitive region, basic mechanisms.
  - Usually what our models target.
- P_{Effect} **(System):**
  - Given P_{gen}, what is the probability that the system will be disturbed?
  - Design topology, operation, frequency:
  - Incorporates design dependent topology and frequency as a transfer function (H(s)).
- $P_{observe}$ **(Test Environment):**
  - Probability that the system disturbance is observed (and nothing else).
  - Goal is to capture and observe every event in its purity with $P_{observe}$ =1
  - Challenges: test system efficiency, dosimetry, and test conductor.
  - Incorporates disturbance from the test system as a transfer function (T(s)).

Cross-Section (σ_{SEF}) As A Set of Transfer Functions of P_{gen}

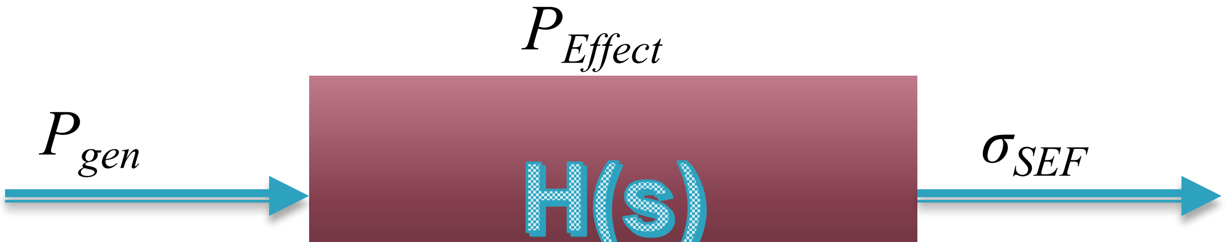


P_{Effect} $P_{observe}$

P_{gen} H(s) T(s) σ_{SEF}

Poor test systems, $P_{observe} \rightarrow 0$:

- *Inability to reliably observe and report failures (lack of observation points, test system monitoring speed, low fluence, limited DUT operation, noisy monitors (probes))*
- *Test system adds noise to data (bad system design, dosimetry, flux control)*



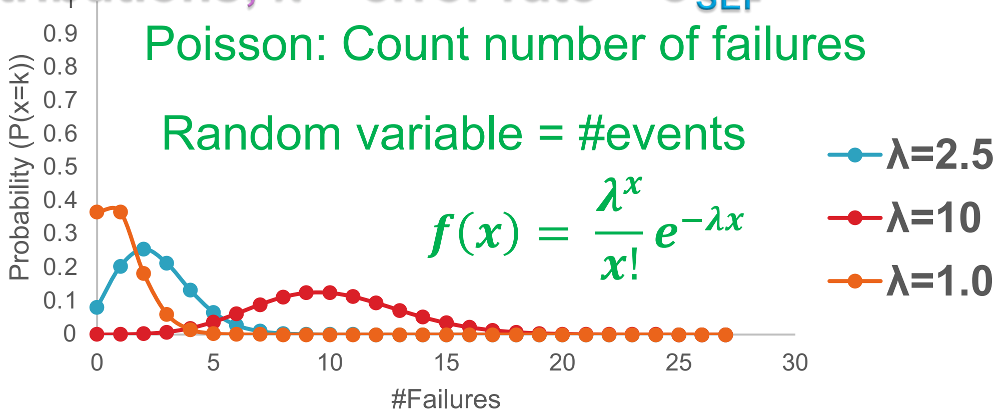
P_{Effect}

P_{gen} H(s) σ_{SEF}

Many assume, $P_{observe} = 1$

Empirical Data Expectations and Probability Distributions

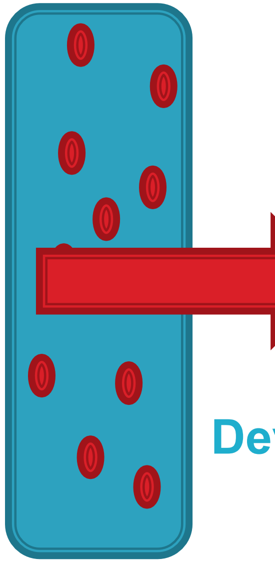
Random trials with independent events can create Poisson and Exponential probability distributions; λ = error rate = σ_{SEF}



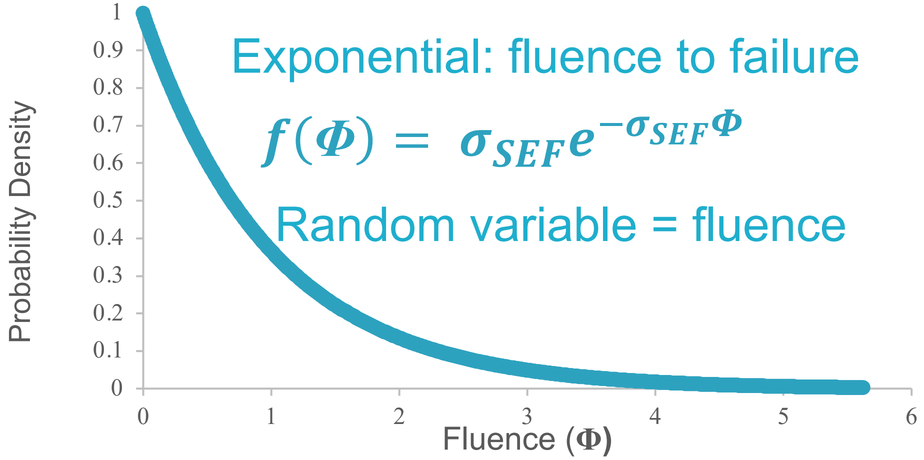
Poisson: Count number of failures

Random variable = #events

$$f(x) = \frac{\lambda^x}{x!} e^{-\lambda x}$$



Device under test



Exponential: fluence to failure

$$f(\Phi) = \sigma_{SEF} e^{-\sigma_{SEF}\Phi}$$

Random variable = fluence

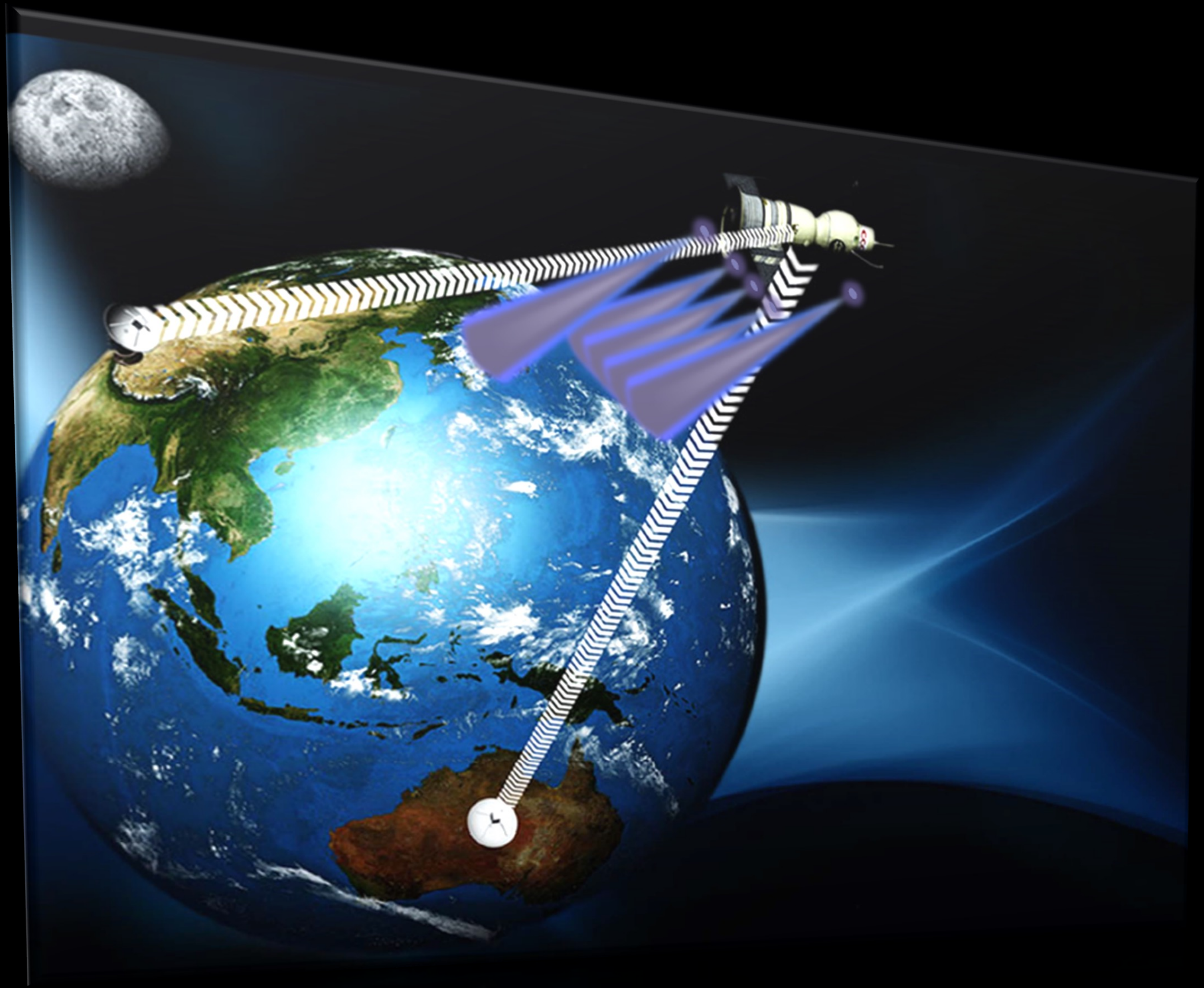
$$\frac{\#failures}{\#trials} = \frac{\#failures}{fluence}$$

- Random trials (random particle strikes)… should create failures that are correlated to a probability distribution
- If the variation of empirical data are too far from the mean, data then:
  - Analyze the integrity of your test system,
  - Check dosimetry
  - Partition your design and add better observation points.

Misperception: exponential test data (cross-sections) will be all over the fluence spectrum.

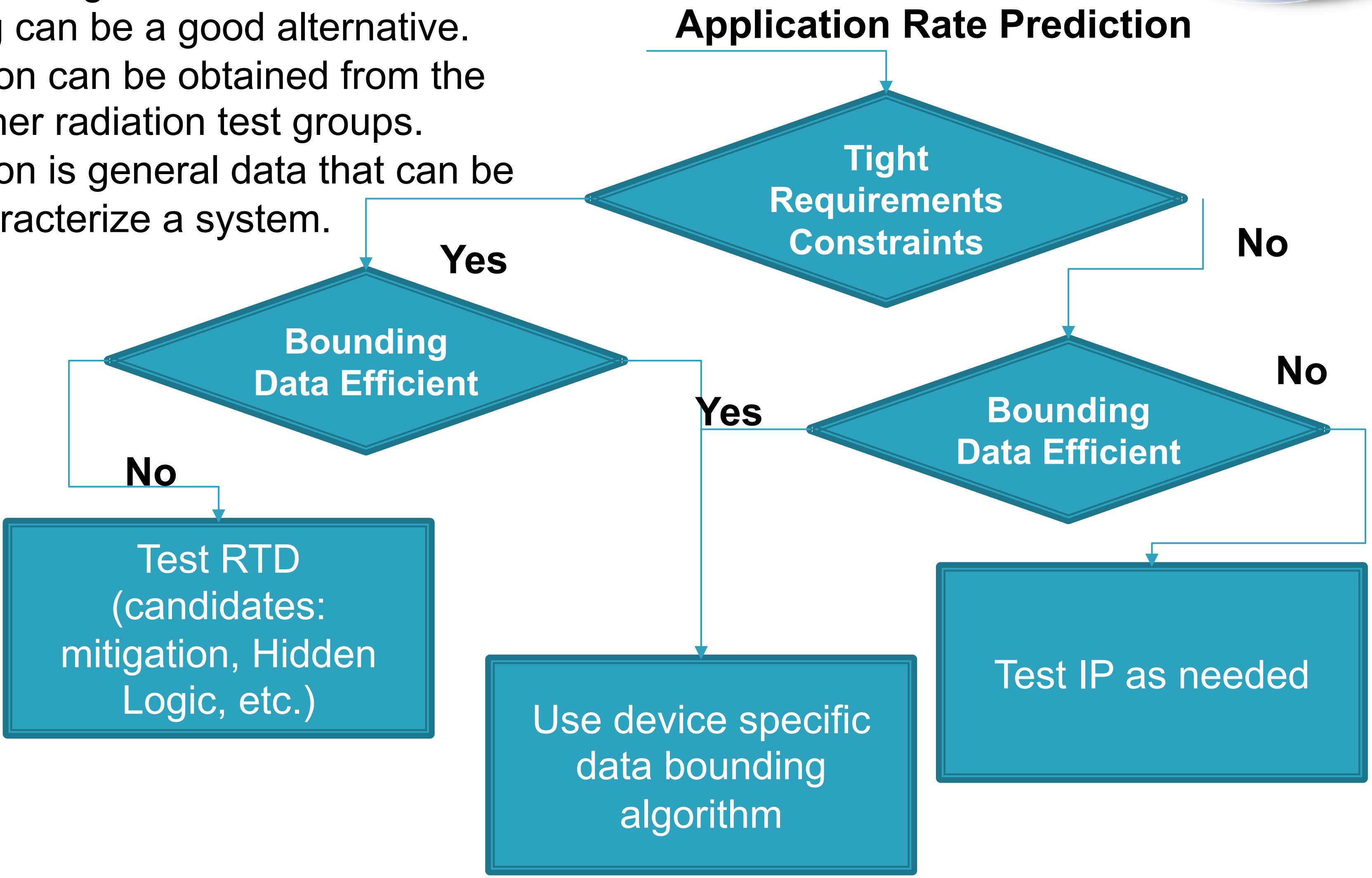
No: for a good test system, exponential experiments will produce “most-likely” data near the mean (although a low number of points will deviate)

Mission Specific Application of Methods



Mission Requirements and Application Rate Prediction

- We can't test every design.
- Error rate bounding can be a good alternative.
- Bounding information can be obtained from the manufacturer or other radiation test groups.
- Bounding information is general data that can be extrapolated to characterize a system.



Application Rate Prediction
Tight Requirements Constraints
Yes
No
Bounding Data Efficient
No
Yes
Bounding Data Efficient
No
Test RTD (candidates: mitigation, Hidden Logic, etc.)
Use device specific data bounding algorithm
Test IP as needed

Systems Created by Serial (Independent/Constant Error Rate) Components

- Simple model…assumes lower-level error rates (λ_n) are independent from one another and can be summed to calculate their system error rate (λ_s).
- A system will have co-dependent parts; however, most systems can be partitioned into independent entities.
- This is the model we use for error rate bounding.

$$System\ Reliability : R_s(t) = e^{-(\lambda_s)t} \qquad System\ Failure: F_s(t) = 1 - e^{-(\lambda_s)t}$$

$$R_s = e^{-\lambda_1 T} \cdot e^{-\lambda_2 T} \cdot e^{-\lambda_3 T} \dots e^{-\lambda_{N-2} T} \cdot e^{-\lambda_{N-1} T} \cdot e^{-\lambda_N T}$$

$$R_s = e^{-(\lambda_1+\lambda_2+\lambda_3+\cdots\lambda_{N-2}+\lambda_{N-1}+\lambda_N)T}$$

$$\lambda_s = \sum_{n=1}^{N} \lambda_n$$

For Serial Reliability, error rates can be summed.

Complex Parallel Systems and Their Steady State

$$\lambda_s = \frac{n\lambda_c e^{\lambda_c t}(1 - e^{\lambda_c t})^{n-1}}{1 - (1 - e^{\lambda_c t})^n}$$

Error rates for a parallel system are not constant over time

Over simplified model assumes each component has the same error rate (λ_c)

$$\lambda_s = n\lambda_c \boxed{\frac{e^{\lambda_c t}(1 - e^{\lambda_c t})^{n-1}}{1 - (1 - e^{\lambda_c t})^n}}$$

Parallel systems derate (reduce error rates)

Reliability of parallel derating decreases over time

Steady state calculation: $\lambda_s = n\lambda_c \lim\limits_{t \to \infty} \frac{e^{\lambda_c t}(1 - e^{\lambda_c t})^{n-1}}{1 - (1 - e^{\lambda_c t})^n} = n\lambda_c$

Most system topologies are a combination of parallel and serial components

The error rate of a parallel topology can be characterized in its steady state (serial form) as the error bound (worst case)

Determining Error Bounds for FPGA Devices

$$\lambda_s = \lim_{t\to\infty} \frac{n\lambda_c e^{\lambda_c t}(1-e^{\lambda_c t})^{n-1}}{1-(1-e^{\lambda_c t})^{n}} = n\lambda_c \lim_{t\to\infty} \frac{e^{\lambda_c t}(1-e^{\lambda_c t})^{n-1}}{1-(1-e^{\lambda_c t})^{n}} = n\lambda_c$$

Generally, error bound calculations are derived from the steady state serial+parallel reliability models.

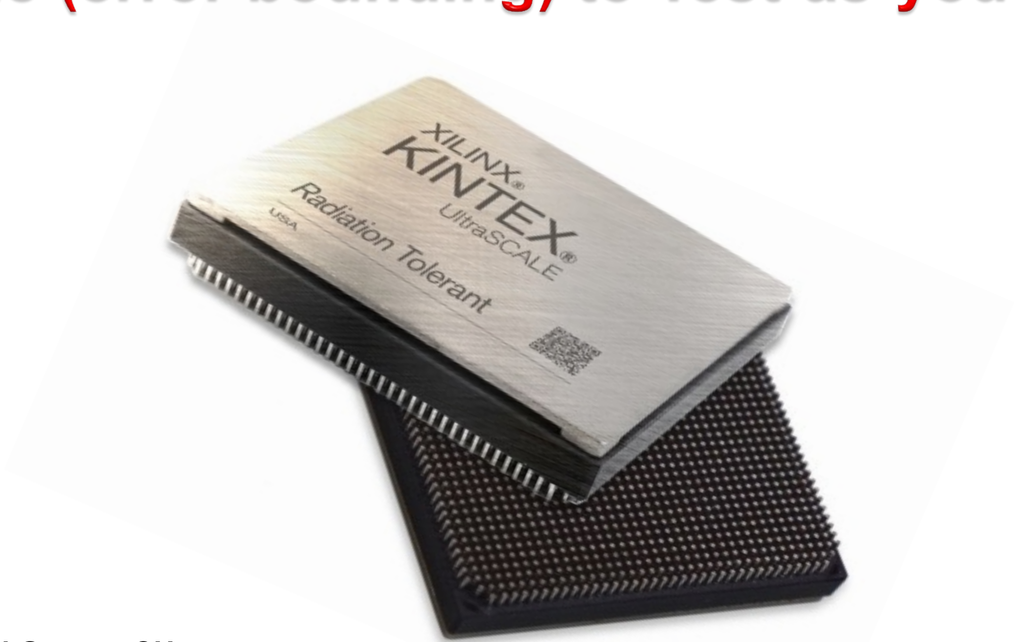
The goal is to:

- Identify a basic element (or elements) that contributes to the dominant mechanism of system failure,
- determine the error rate of said component, and then
- extrapolate to the FPGA design being analyzed
- **Different per FPGA type**

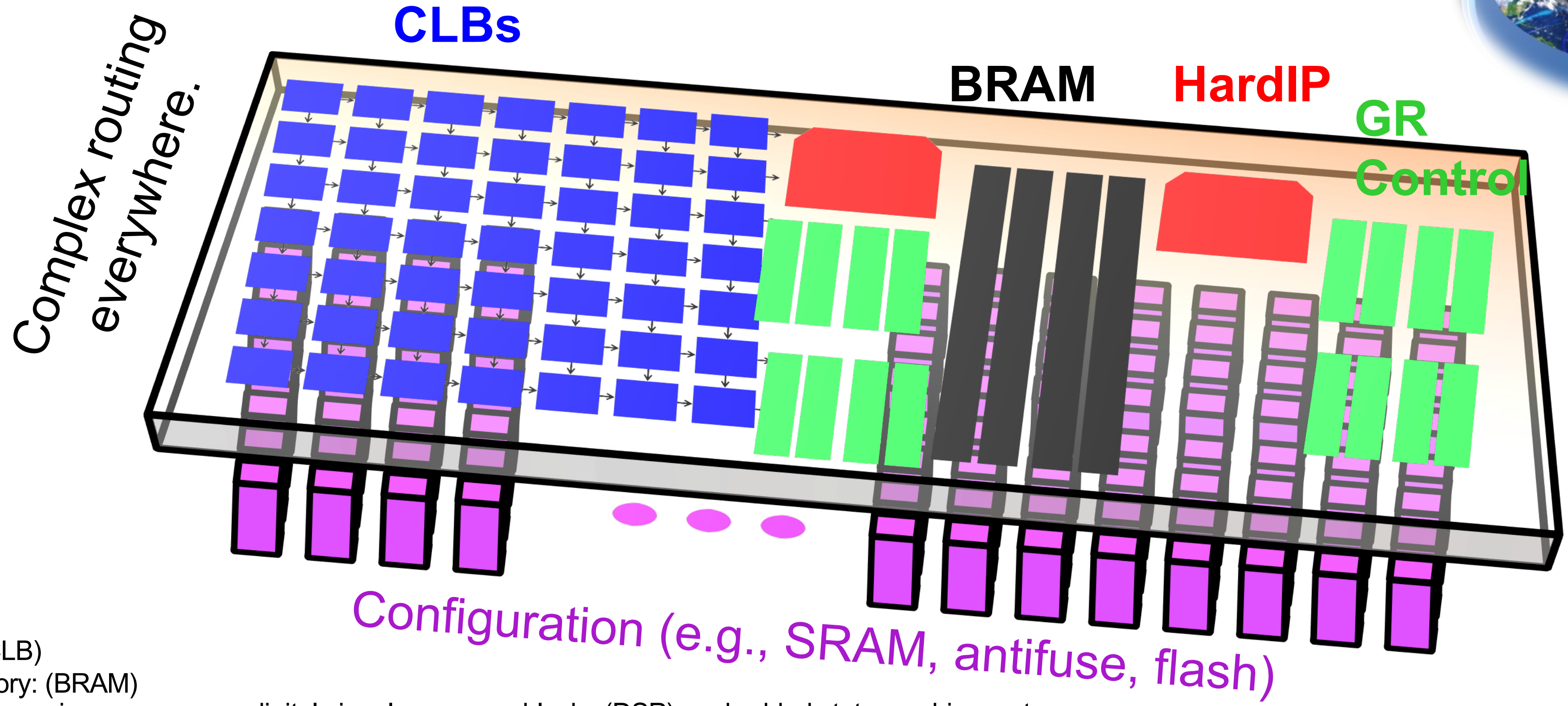
Xilinx error bounding example follows

Xilinx σ_{SEF} Example (SRAM-Based FPGA):

System-Level Calculations Comparing Linear Translation Technique (error bounding) to Test-as-you-fly H(s)



FPGA SEU Cross Section Model


CLBs
Complex routing everywhere.
BRAM
HardIP
GR
Control
Configuration (e.g., SRAM, antifuse, flash)

Configurable logic block: (CLB)
Block random access memory: (BRAM)
Intellectual property: (IP); e.g., micro processors, digital signal processor blocks (DSP), embedded state machines, etc.
Global Routes: (GR)
Analog circuits

SEU Cross sections for a mapped design (σ_{SEF}) are based on the FPGA's internal elements and the mapped design's topology.

$$\sigma_{SEF} = f(\sigma_{configuration}, \sigma_{BRAM}, \sigma_{functionalLogic}, \sigma_{HiddenLogic})$$

Dominant mechanisms of failure will drive σ_{SEF}

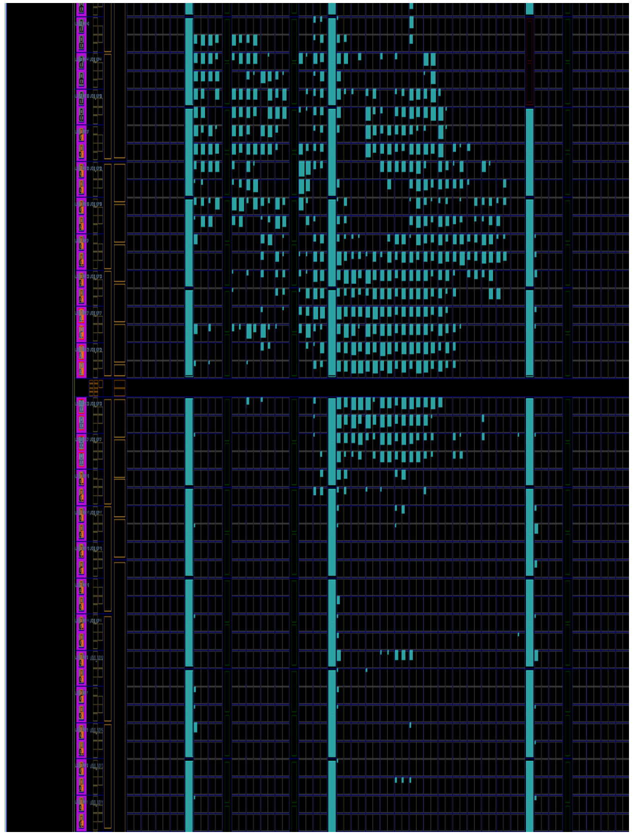
Error Bounding with Xilinx SRAM Based FPGAs

PS: processor; PL: programmable logic; ECC: Error correcting codes

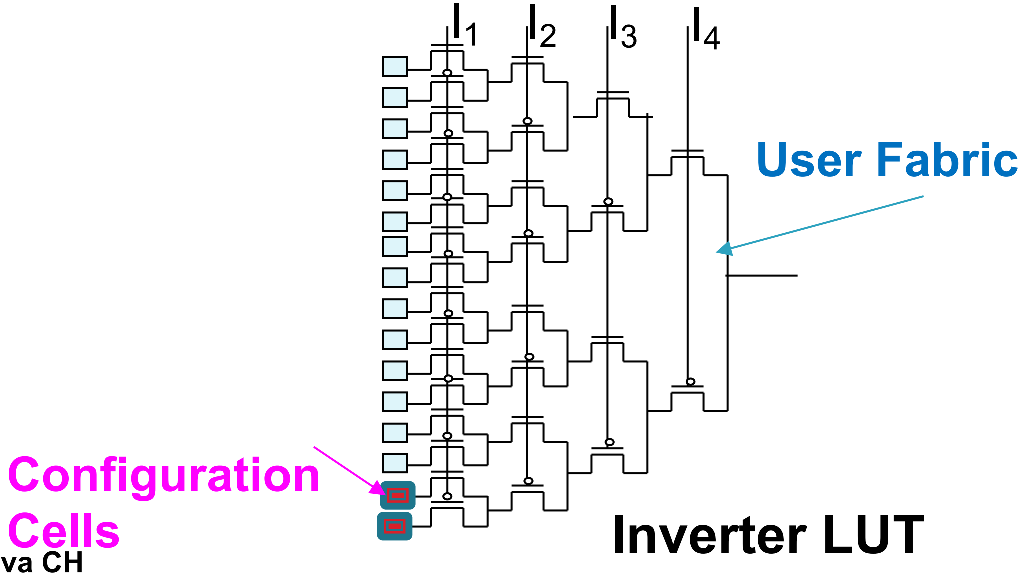
- It has been shown that in older Xilinx FPGA devices (Ultrascale (20 nm) and older (above 20nm)) the SRAM-based configuration and embedded BRAM (with no ECC) are the dominant mechanisms of failure.
- When using a SoC (embedded processor such as the Xilinx Zynq), configuration does not cover the PS, it covers only PL.
- The following is a walk-through example of using the linear extrapolation method with configuration+BRAM bits to bound error rates.

Xilinx Configuration, Mask, and Essential Bits

Design mapping into user fabric logic cells is defined by configuration bit settings.



- **Configuration bits:** Total number of configuration cells… (fixed per each FPGA type)
  - **Masked bits:** calculated by the manufacturer and is not under user control… design and device dependent
  - Unmasked bits
- **Essential bits**: number of configuration cells used by the design mapping (calculated by the manufacturer upon user directive… design and device dependent).



Inverter LUT

Simplification of σ Calculations: Linear Translation

Generally, configuration cross-sections are readily available from generic device investigations.

$$\sigma(LET)_{configuration\_Device} = \frac{\#events}{\#Particles/cm^2}$$

$$\sigma(LET)_{configuration\_bit} = \frac{\#events}{\left(\#\frac{Particles}{cm^2}\right) * (\#unmaskedconfigurationBits)}$$

$$\sigma(LET)_{\mathrm{Essential}} = Essential\_bits \times \sigma(LET)_{configuration\_bit}$$

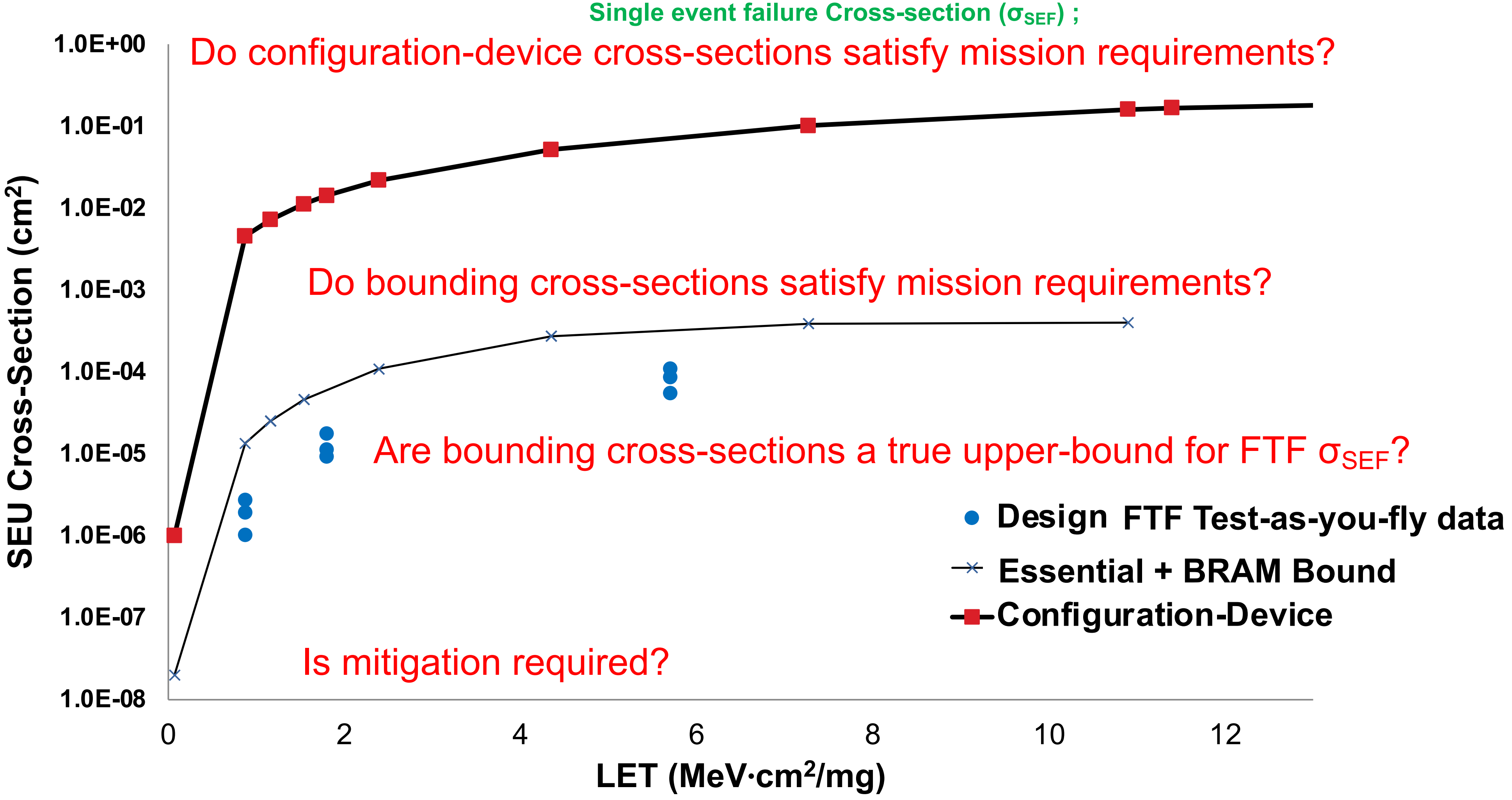
$$\sigma(LET)_{\mathrm{BRAM}} = BRAM\_bits \times \sigma(LET)_{\mathrm{BRAM}\_bit}$$

$$\sigma(LET)_{BOUND} = \sigma(LET)_{\mathrm{Essential}} + \sigma(LET)_{\mathrm{BRAM}\_bit}$$

$$\sigma(LET)_{SEF} = \mathbf{1}/\mathrm{FTF} = \mathbf{1}/((FailureTime - BeamStartTime) * \mathrm{AverageFlux})$$

$\sigma(LET)_{SEF}$ is measured per test per LET (i.e., either 1 event or 0 events per test). There are several FTF tests that are required to be performed per LET

If Upper-bounds Satisfy Mission Reliability/Survivability Requirements, Then No Mitigation is Required.



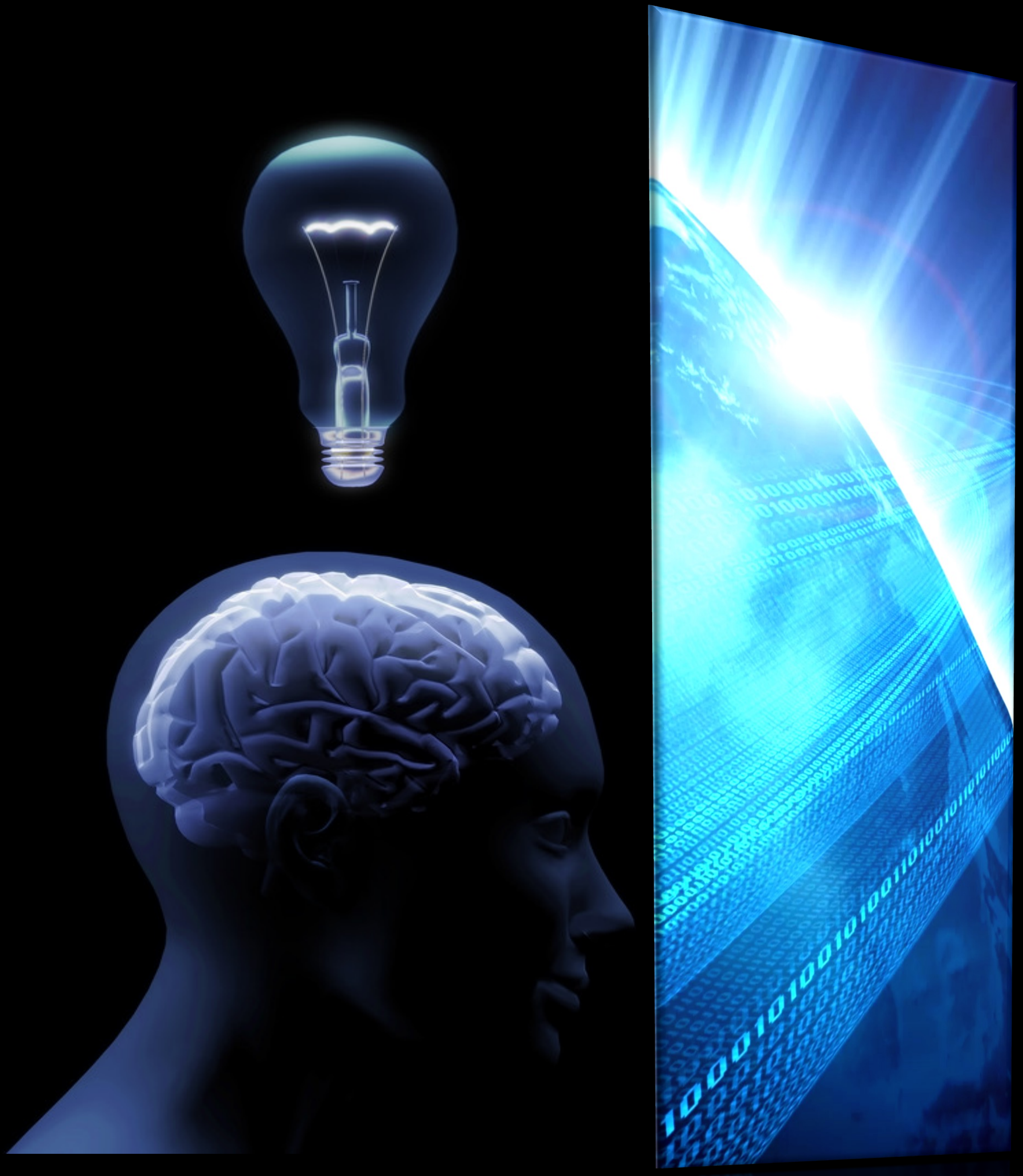
Single event failure Cross-section (σ_{SEF}) ;

Do configuration-device cross-sections satisfy mission requirements?

Do bounding cross-sections satisfy mission requirements?

Are bounding cross-sections a true upper-bound for FTF σ_{SEF}?

Is mitigation required?



Users should include design-specific σ_{BRAM} contribution into the bounding.

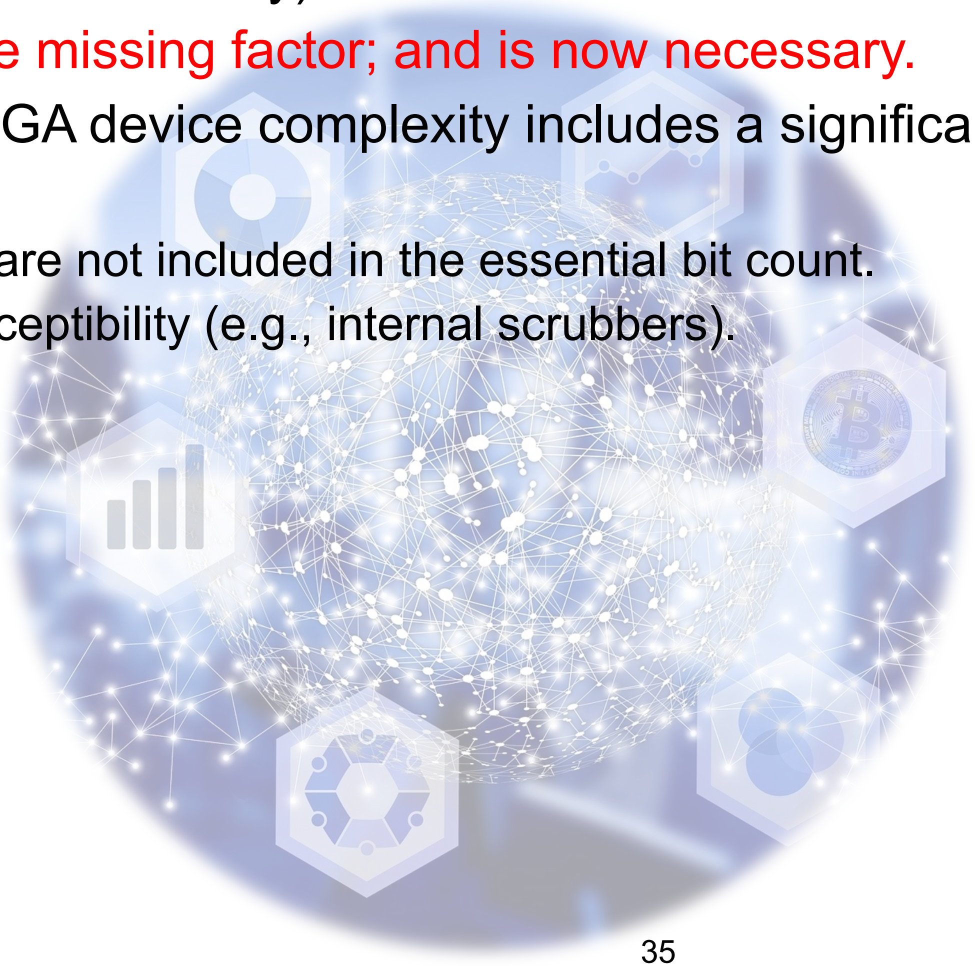
BRAM: bit and SEFI error rates.

Usually the SEFI rate is insignificant for bounding rates, but can be crucial for SEF or FTF (system characterization).

if BRAM includes ECC, and this mitigation significantly lowers BRAM-bit error rates, then the user should include BRAM-SEFI error rate contributions to bounding.

Prove It Before You Use It

- Using $\sigma(LET)_{BOUND}$ as upper-bounds is not a new concept.
- However, $\sigma(LET)_{BOUND}$ should only be used if it is known/proven to be an upper-bound (or close enough depending on criticality).
- The proof of bounding has been the missing factor; and is now necessary.
- Why is a proof necessary now? FPGA device complexity includes a significant amount of hidden logic.
  - Hidden logic have components that are not included in the essential bit count.
  - It has shown (in flight) to impact susceptibility (e.g., internal scrubbers).



It's So Simple: Why Not Always Use Error Bounding?

COTs: Commercial off the shelf

- Error bounding can be a gross overestimate. If requirements are met, this is not a problem. However, if requirements are not met, refinement is necessary. Usage of COTs has made this our reality.
- Error bounding cannot measure the efficacy of your mitigation:
  - Mitigation adds circuitry
  - The intent to measure mitigation efficacy is lost because bounding will increase with the increase of circuitry; and hence the calculated error rate will increase with mitigation.
- Detailed information of operational failures might be necessary…requirements example follows:
  - Requirement states no ground intervention for specified # of days.
  - Most SEE system faults can be autonomously cleared;
  - However, some system faults cannot and will need ground intervention.
  - As is, the design does not meet requirements using error bounding calculations.
  - Too late to add mitigation!
  - What is the probability of these events occurring as opposed to other events? (i.e., any bit-flip versus a specific event).
  - If it is a low probability, then no need to add mitigation.
  - In many cases we cannot report and determine the probability for these events using error bounding.
  - Test as you fly is necessary!

If Bounding Methods Are Not Sufficient, Test-As-You-Fly Might Be Necessary

If Test-as-you-fly is necessary:

- Strive to test FPGA IP that embody real applications: representative tactical design (RTD).
- Strategy is complex, however, in some cases it is necessary.
- Use real-time control and response test systems.
- Mimic DUT peripherals such that the DUT assumes tactical operation during test (beam –exposure).
- **Space R3 has a test platform that can mimic a space craft/instrument. DUT is tested as if it communicating and operating as it would in space.**

We can now test the full system in its operational state

$$\sigma_{SEF} = f\left(\sigma_{configuration}, \sigma_{BRAM}, \sigma_{functionalLogic}, \sigma_{HiddenLogic}\right)$$

An illustration of the Orion spacecraft traveling through space. (Image credit: NASA)

Fluence-to-Failure Testing and Test-as-you-fly Background Information

- **Failure (for each experiment) must be defined prior to testing**.
  - Unexpected behavior will occur… what should be done?
  - User decides when to stop each experiment.
- **Fluence-to-failure is a misunderstood technique:**
  - Fluence-to-failure describes the technique for calculating cross-sections.
    - $\sigma_{SEF} = {}^{1}/_{fluence-to-failure}$
  - Fluence-to-failure does not describe when you stop the beam
    - Poor test systems rely on human observation and manual test termination.
      - Fluence-to-failure calculations will be wrong.
      - User can miss important events…$P_{\text{observation}}$ is low
  - A good automated test system will have the ability to timestamp event occurrences.
  - Post processing will differentiate event-effects and calculate the fluence-to-failure needed for σ_{SEF}.
  - Multiple tests (per LET) are required to hone in on the mean probability of failure.

The Importance of Fluence-to-Failure Testing

- **Fluence-to-failure is a top-down test method that incorporates the full design:**
  - System topology with clocks and resets
  - Embedded IP and hidden logic
  - **Mixed signal effects (ground plane instability): hot topic**
  - Derating (includes mitigation efficacy)
  - Co-dependencies between all the above per tactical application
- **Provides detailed sensitivities/fault probabilities of a system… examples:**
  - Scrubber performance/sensitivity (internal versus external)
  - Efficacy of mitigation
  - Protocol bit upsets versus communication hang-ups
  - The above cannot be measured with error-bounds.

SEE Cross Section Data Metrics

$$fluence = \frac{\#ions}{cm^2}$$

LET: Linear Energy Transfer

Countable events-effects $\sigma(LET) = \frac{\#Events}{fluence}$ Cross Section: Number of events per fluence (Poisson Distribution)

System events-effects $\sigma_{SEF}(LET) = \frac{1}{fluence}$ Cross Section: How much fluence until system malfunction (Exponential Distribution)

- SEF: data represent fluence to failure (FTF) and events are 1 count per experiment.
- Statistics:
  - Multiple tests are required.
  - Important to differentiate (two very different questions):
    - What is the probability that each empirical SEF sample will fall close to the mean?
      - Depends on the quality of the test system (plus other considerations)
      - Typical rule of thumb for a system with good visibility and control is 5-10 tests (assumption is that a large percentage of experimental data points will fall close to the mean)
    - How many tests are required to reliably calculate a mean?
      - Don’t be fooled. For a poor test system, increasing the number of experiments will not help.

Fluence-to-Failure Experiments and The Exponential Model

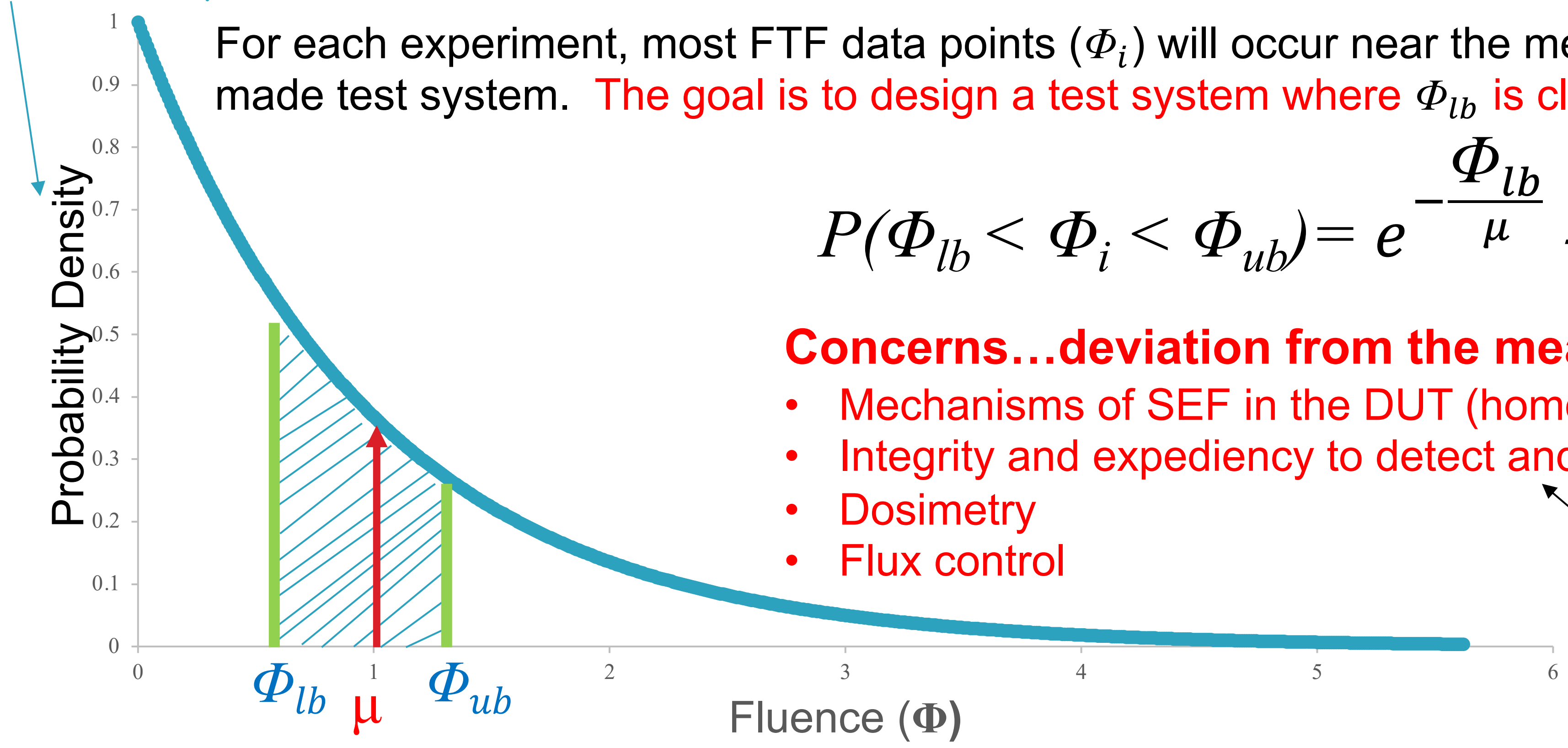
Classical Reliability : transformation from the time domain to the fluence domain.

	Exponential Distribution Variables
Fluence-to-failure (FTF)	Φ_i Random Variable: per experiment-*i* for a selected LET
SEF Cross-section (rate w.r.t. fluence)	$\sigma_{SEF_i} = \frac{1}{\Phi_i}$
Sample mean (MFTF)	$\mu = \frac{1}{n}\sum_{i=i}^{T}\Phi_i$ Average of fluence-to-failure test results. ***n* = number of events** ***T* = number of experiments**
Mean SEF	$\sigma_{SEF\mu} = \frac{1}{\mu}$ Classical Reliability: Constant per LET
Standard deviation	$\mu = MFTF$ Use of exponential population standard deviation definition
Standard error of the mean (SEM)	$\frac{\mu}{\sqrt{n}}$ Generally used for error bars
Exponential PDF Probability distribution function	$\sigma_{SEF_\mu} e^{-\sigma_{SEF_\mu}\Phi}$ or $\frac{1}{\mu}e^{-\frac{1}{\mu}\Phi}$

FTF PDF Expected Empirical Data: How Can 5-10 Tests Be Sufficient?

$$\boldsymbol{Experiment\ fluence\ to\ failure = \Phi_i = \frac{1}{\sigma_{SEF_i}}} \qquad \boldsymbol{MFTF = \mu}$$

$$f(\Phi) = \sigma_{SEF_\mu} e^{-\sigma_{SEF_\mu}\Phi}$$



For each experiment, most FTF data points (Φ_i) will occur near the mean, for a well-made test system. The goal is to design a test system where Φ_{lb} is close to Φ_{ub}

$$P(\Phi_{lb} < \Phi_i < \Phi_{ub}) = e^{-\frac{\Phi_{lb}}{\mu}} - e^{-\frac{\Phi_{ub}}{\mu}}$$

Concerns…deviation from the mean depends on:

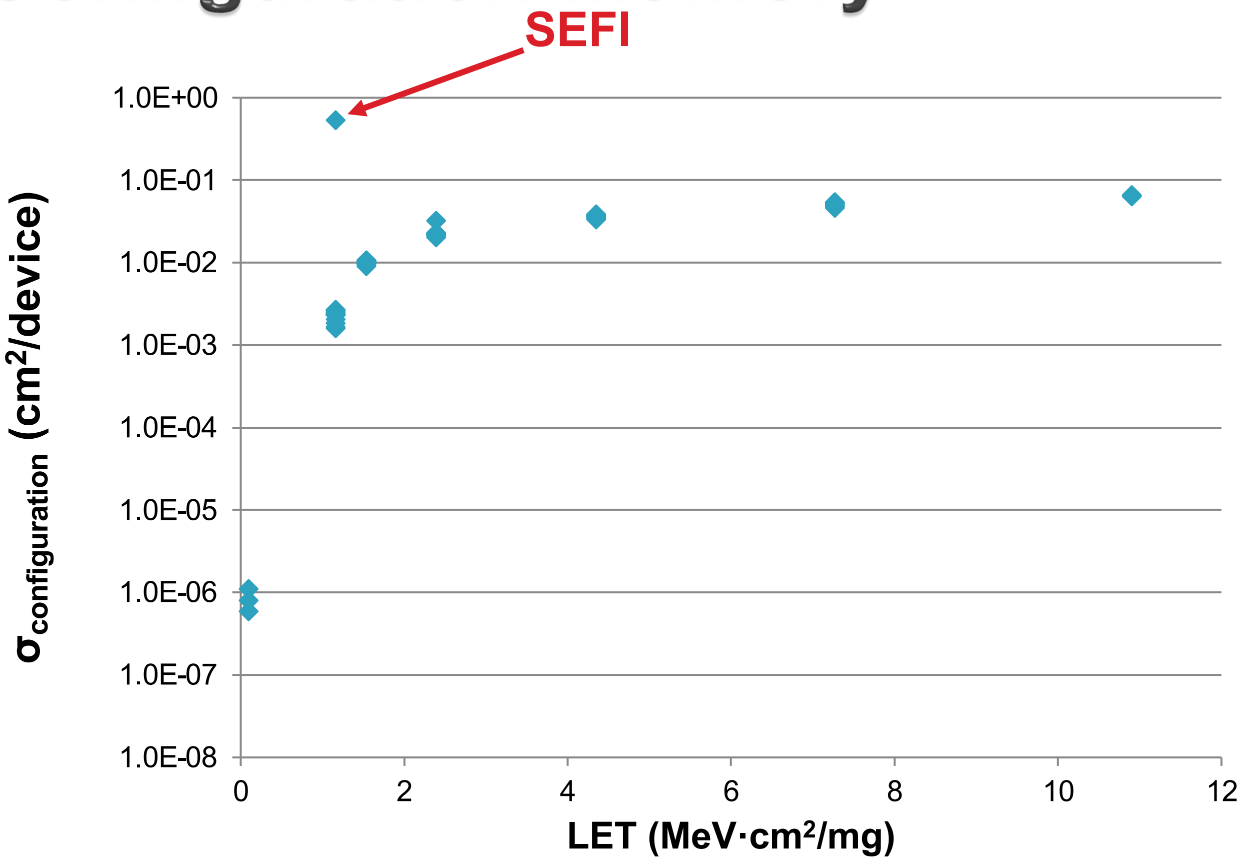
- Mechanisms of SEF in the DUT (homogenous, multi-modal)
- Integrity and expediency to detect and report SEF
- Dosimetry
- Flux control

The reality is: increasing the number of tests will not bring your empirical mean closer to the actual mean if concerns are not controlled.

Testing Homogenous Cells versus Complex Systems

- **Homogenous Cells:**
  - Copies of a simple structure (inverters, buffers, memory cells)
  - Testing homogeneous cells is easy!
  - Each test has many targets (that are the same components) and hence increases statistics.
- **Complex systems:**
  - Many variables, moving parts, and state space exploration paths
  - Difficult to test and requires strategic planning.
  - Planning includes taking advantage of dominant mechanisms of failure.
  - Remember, error rate tests are not simply looking for if a failure can occur.
  - Alternatively, the tests are evaluating probabilities of failure with respect to fluence exposure.

Homogenous Cross Sections: FPGA Configuration Memory



LET	Number of Experiments (82 total tests)
0.1	3
1.16	21
1.54	16
2.39	15
4.35	12
7.27	12
10.9	3

- SEFIs can occur, however, they have a low event probability during testing (depending on fluence)
- If the experiments go to a high enough fluence, it is highly likely that a SEFI will occur… yet its $\sigma_{configuration}$ will be low

All 82 tests are represented in the graph. The results are so close that it is difficult to decipher between each experiment per LET.

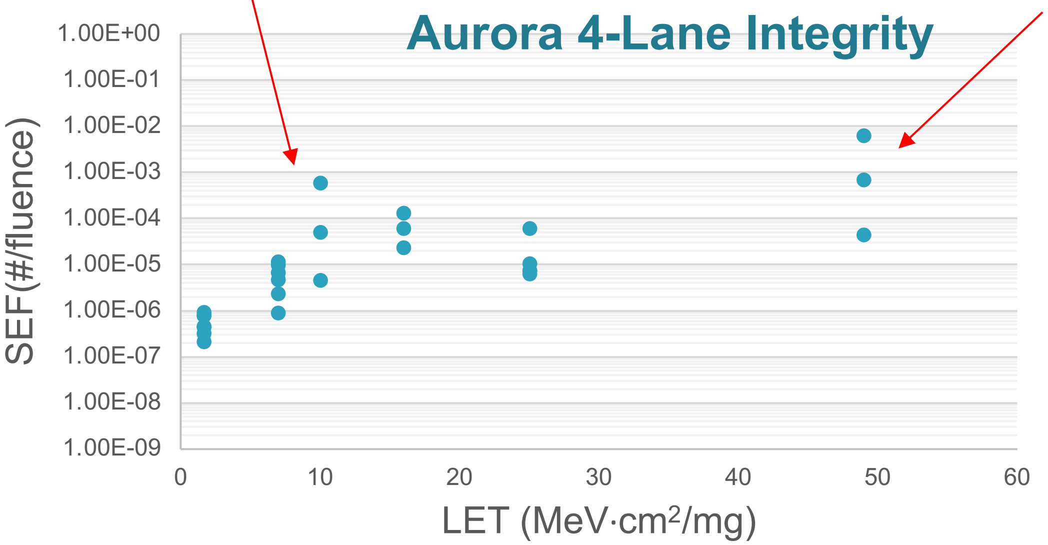
Test-as-you-fly: Aurora Data-SEUs and Lane Integrity

In-air testing at LBNL 16 MeV/u… bad dosimetry day!

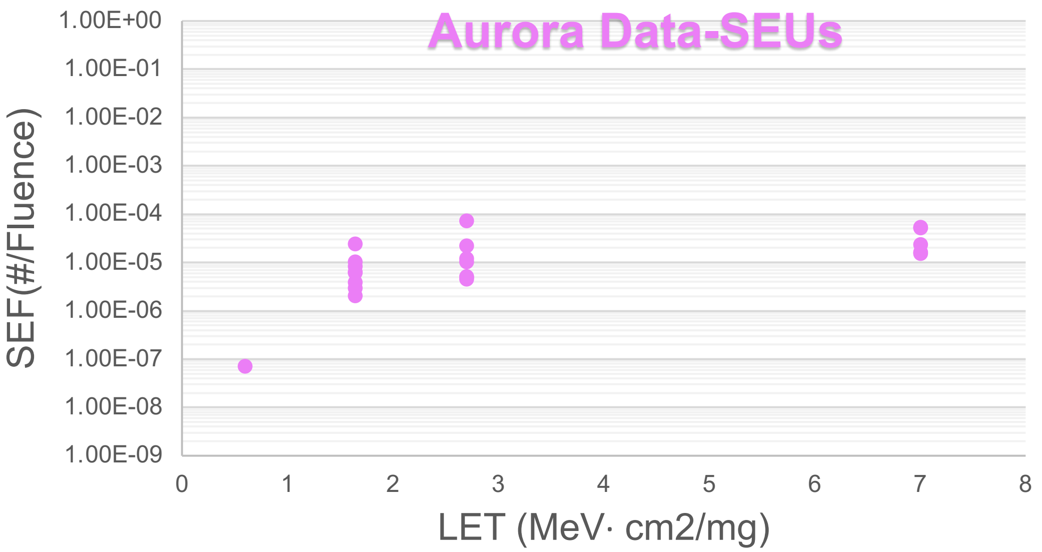
Most data points are contained within a decade per LET

Bad dosimetry at V=10 MeV·cm²/mg

Inadequate Penetration



0.6 MeV·cm²/mg < LET_{TH} < 1.64 MeV·cm²/mg

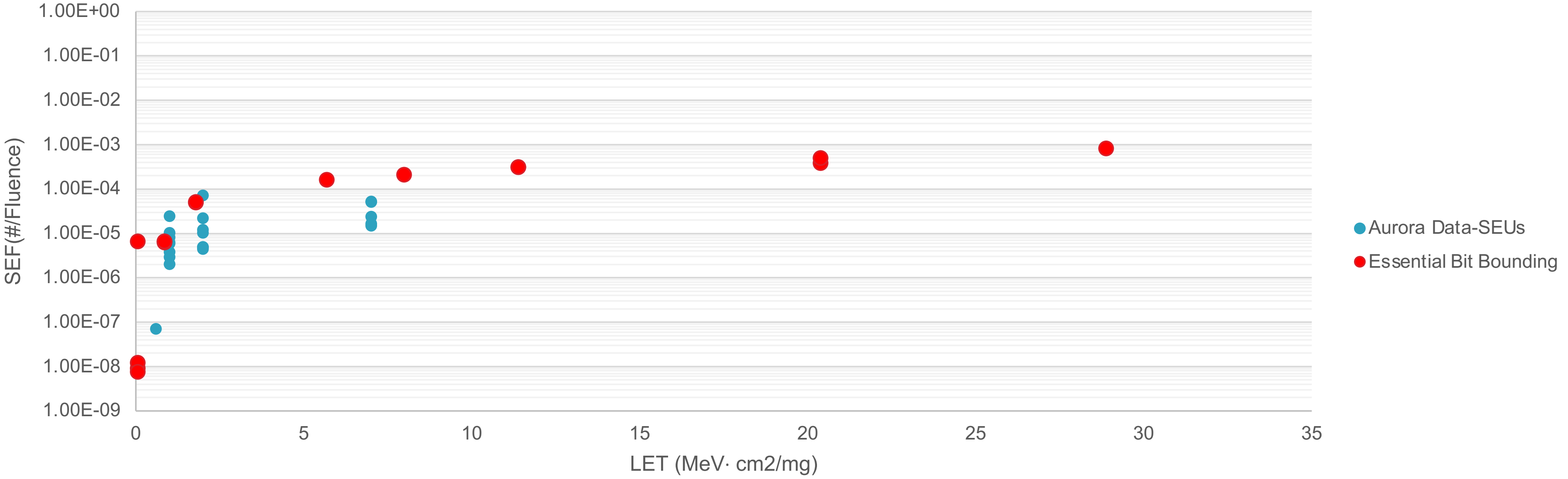


LET_{TH} <.0.6 MeV·cm²/mg

Test-as-you-fly can differentiate between lane health and bit-flips. Bit flips have slightly higher cross sections than lane down-links (as they should).

Aurora Data-SEUs Not fully Covered by Essential Bit Bounds at Low LET

Example illustrating the impact of hidden mechanisms to SEF Cross Sections

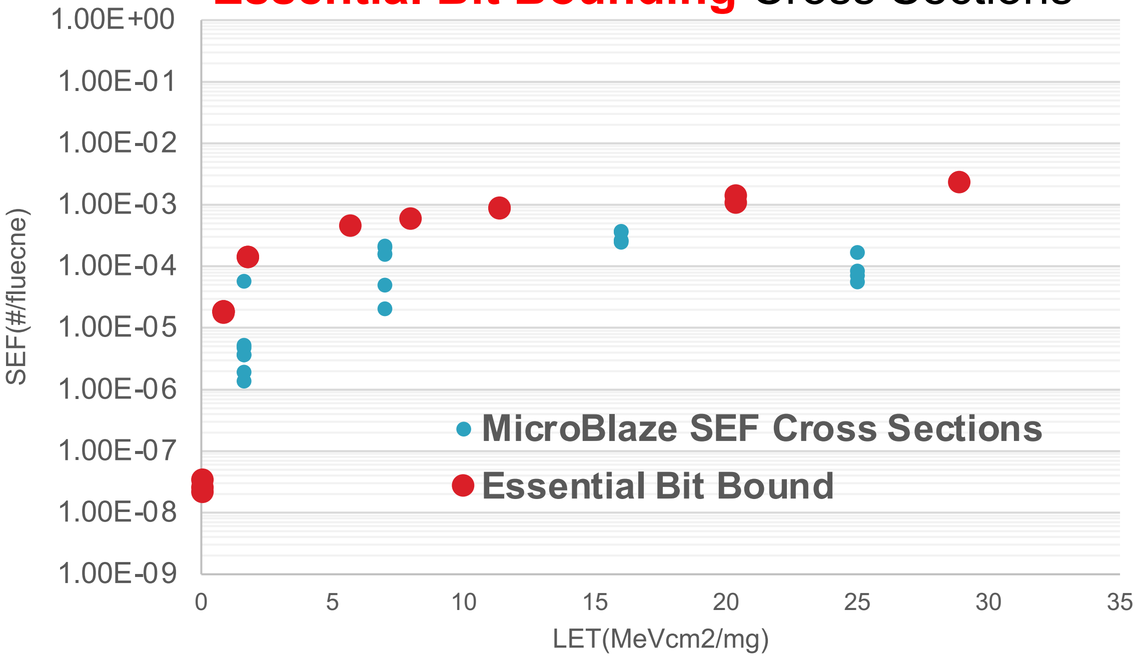


- **Assumption is that essential bit cross sections will bound SEF cross-sections. This is not always the case.**
- **SEF for Aurora data-bits can be higher than essential bit cross sections**
- **This is most likely due to buffering and hidden/embedded structures (not accounted for in the essential bit calculator)**

MicroBlaze SEF Cross Sections

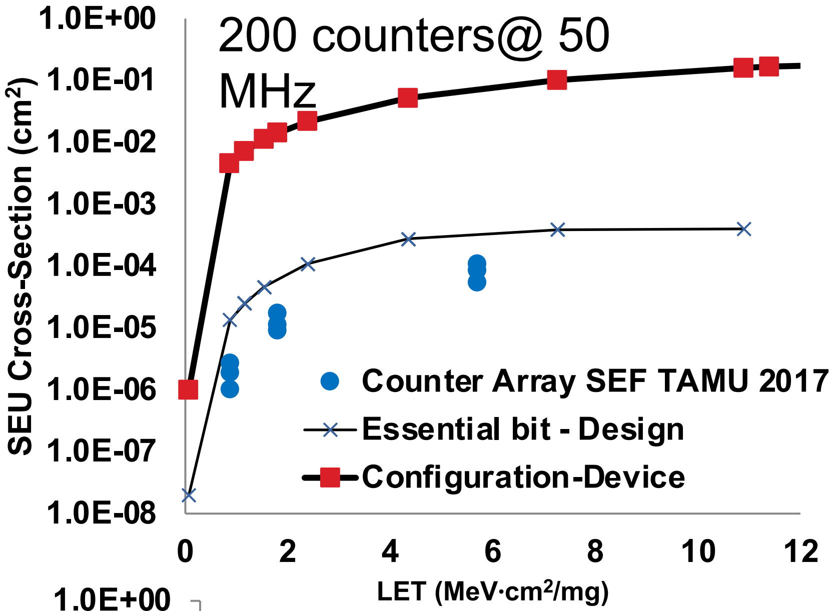
Depending on the target environment and MicroBlaze usage, SEFs can in the order of days to hundreds of days. For worst-week 100 mils, SEFs can be in the order of hours.

MicroBlaze SEF Cross Sections and
Essential Bit Bounding Cross Sections

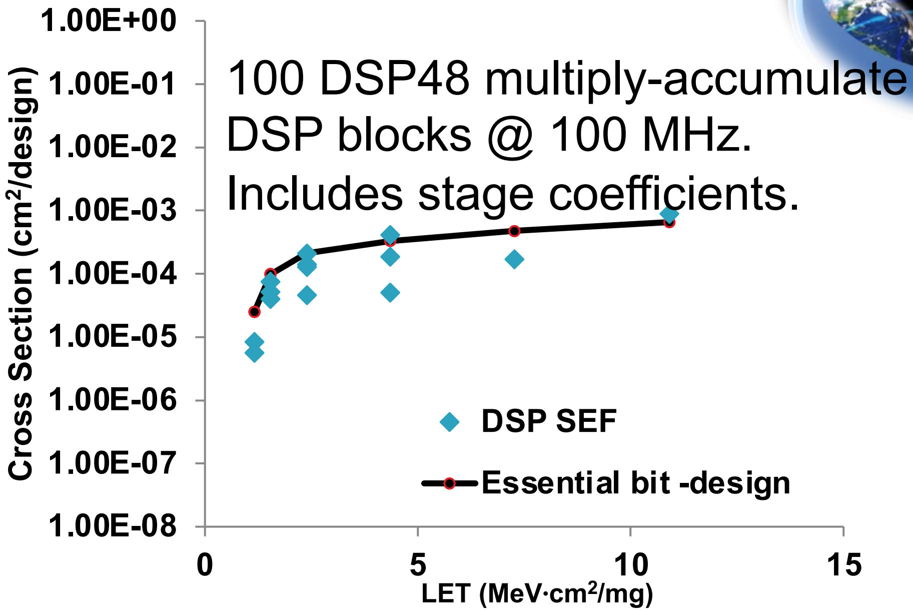


Use of Essential bit Bounding can provide a reasonable error rate estimate.

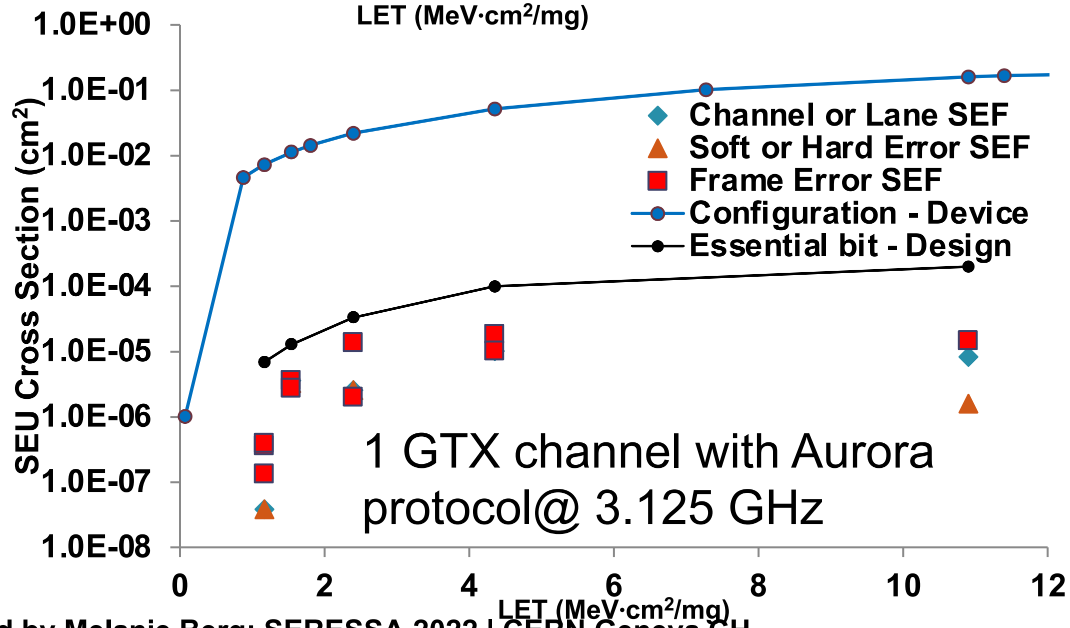
Kintex UltraScale SEU Cross-Sections



200 counters@ 50 MHz



100 DSP48 multiply-accumulate DSP blocks @ 100 MHz. Includes stage coefficients.



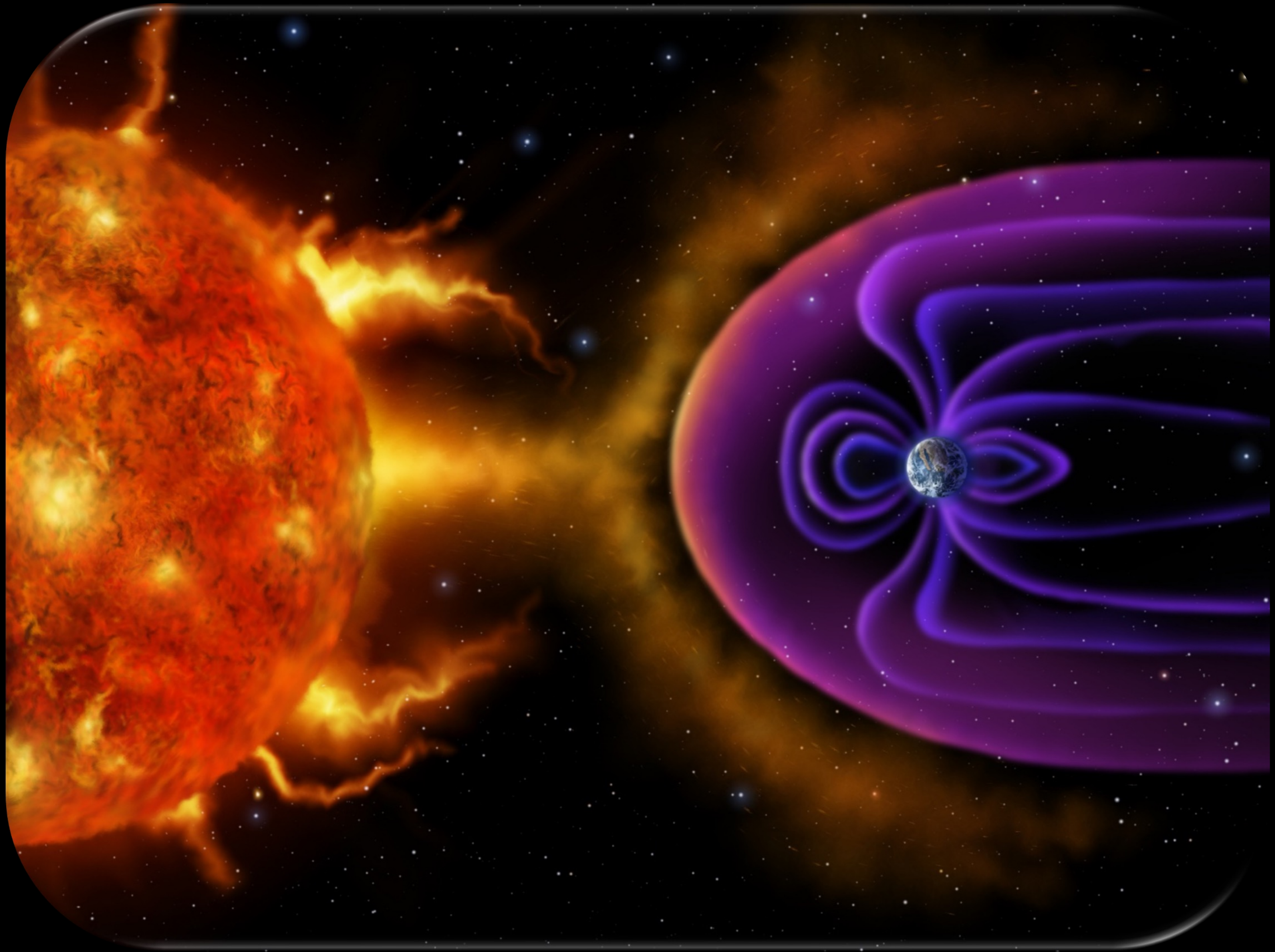
1 GTX channel with Aurora protocol@ 3.125 GHz

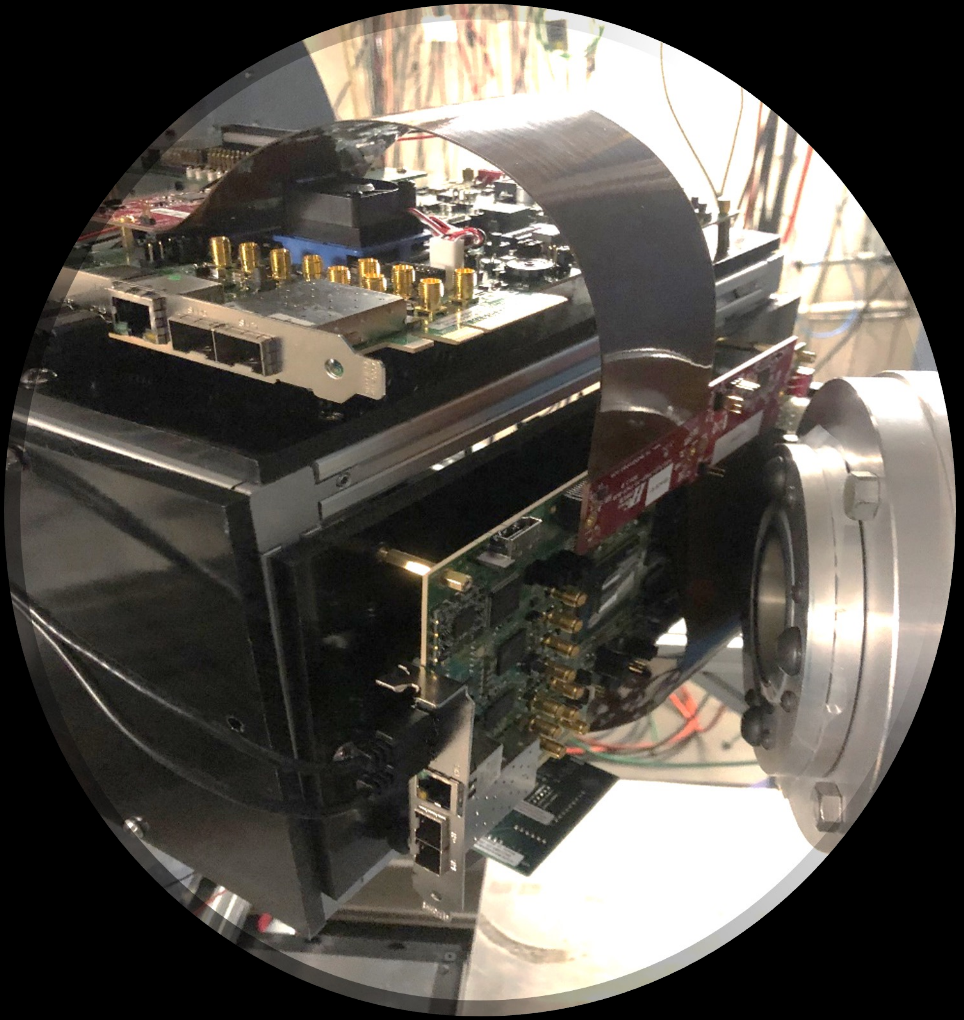
$\sigma_{ssential\_bit} > \sigma_{SEF}$

Implies $\sigma_{Essential\_bit}$ can be used to predict survivability (non-mitigated design).

\

Empirical Data and Analysis





CREME96 Environment Flux Bins

$$\boldsymbol{Differential\ Flux} = \frac{\#ions}{cm^2 \cdot day \cdot (MeV \cdot \frac{cm^2}{mg}) \cdot \Delta LET}$$

$$\boldsymbol{LET} = (MeV \cdot \frac{cm^2}{mg})$$

Low LET Bins

LET	Differential Flux
0.00162752	5105.903934
0.00164651	7832.196377
0.00166571	15419.65236
0.00168514	14570.46082
0.0017048	13542.46358
0.00172468	13653.64409
0.0017448	14689.21758
0.00176515	16039.42323
0.00178574	17336.28967
0.00180657	18630.93558
0.00182764	20305.34293
0.00184896	21751.11611
0.00187052	23301.22349
0.00189234	25957.92746
0.00191441	28483.81801
0.00193674	30429.20872
0.00195933	32809.42086

J. Barak, **R.A.** Reed, **and** K.A. LaBel,"On the Figure of Merit Model for SEU Rate Calculations",
IEEE **TRANSACTIONS** ON NUCLEAR SCIENCE, **VOL. 46.** NO. 6, DECEMBER 1999

$$R_h = \int_0^\infty f(L)\sigma(L)dL$$

Transformation to numerical integration

$$R_h = \lim_{\Delta L \to 0} \sum_{L=0}^{NB} f(L) * \sigma(L)\Delta L$$

- We have the binned differential flux
- We need $\sigma(L)$.
- We need to test.

Mid-Range LET Bins

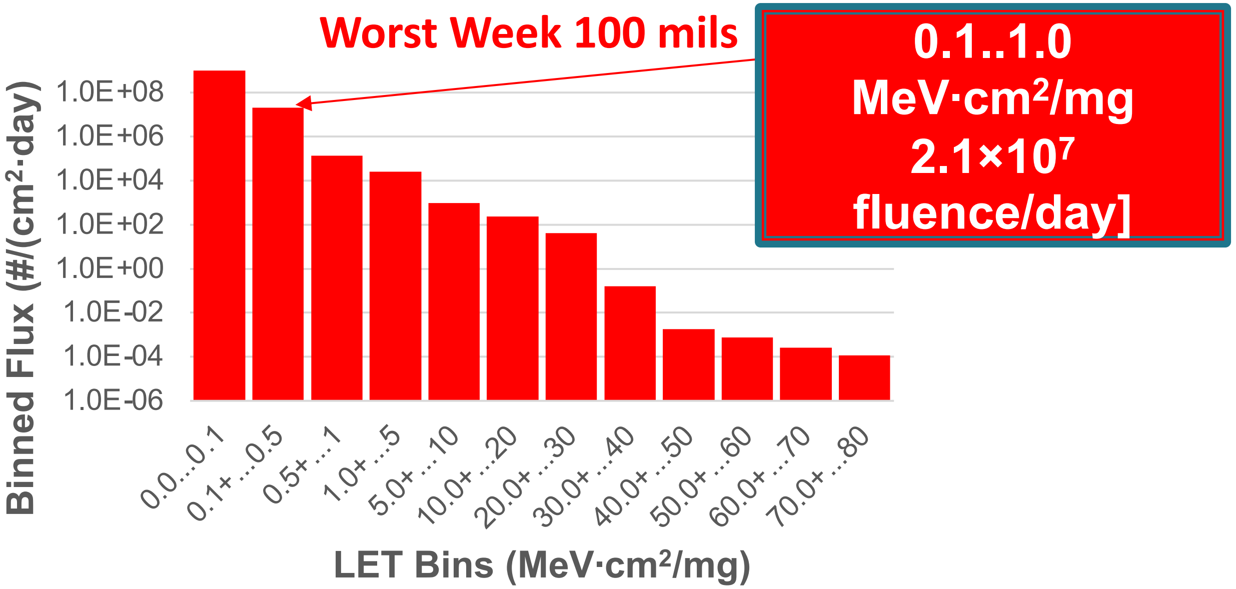
LET	Differential Flux
9.41207	9.275450831
9.52185	9.45859438
9.63292	8.383953917
9.74528	7.485131836
9.85895	7.232144294
9.97394	6.942081614
10.0903	6.871039145
10.208	7.00345475
10.327	6.898374333
10.4475	6.584985778
10.5694	6.591478144
10.6926	6.708589693
10.8174	6.451545109
10.9435	6.074335239
11.0712	6.266070618
11.2003	6.255571675
11.331	5.971099159
11.4631	6.072492965
11.5968	6.289463961
11.7321	6.630031705
11.8689	6.483416192
12.0074	5.488057435
12.1474	4.750240095

Units have been converted from raw CRÈME output.

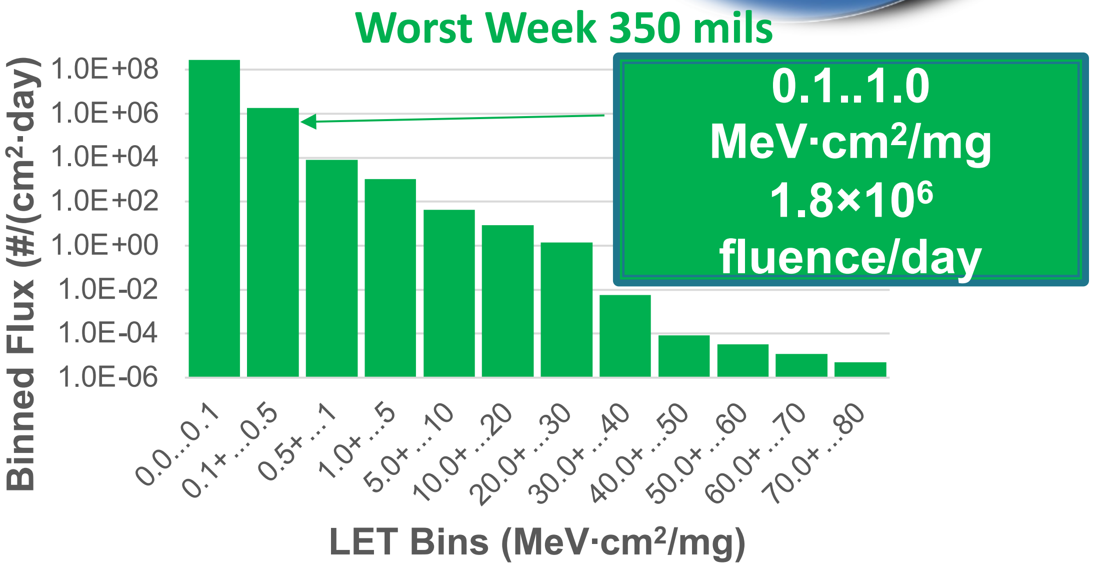
Flux is binned

Why Are Low LET Data Important?

Comparing Particle Flux Across Environment Conditions


Worst Week 100 mils

0.1..1.0 MeV·cm^2/mg 2.1×10^7 fluence/day]

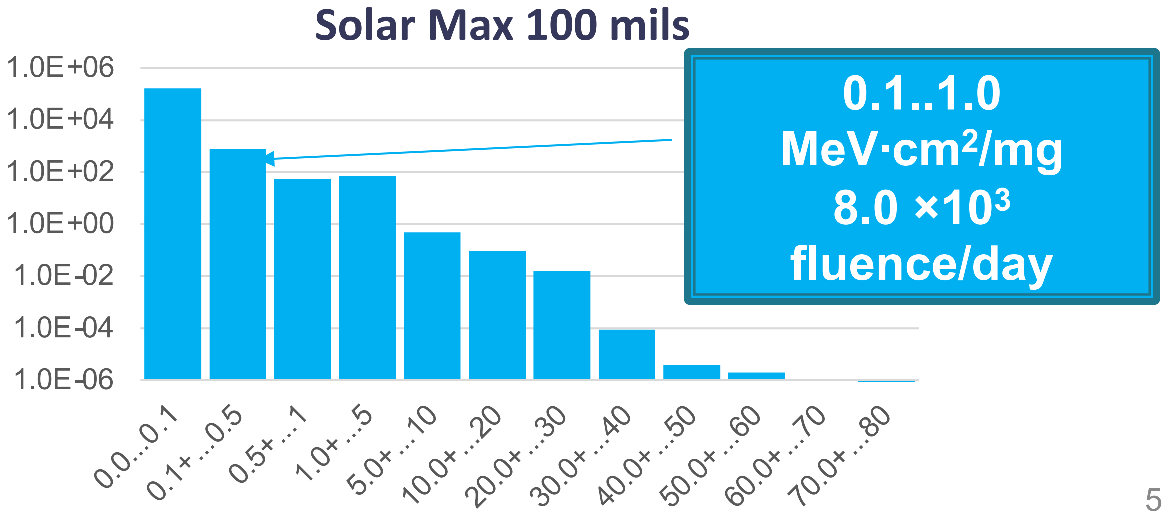

Worst Week 350 mils

0.1..1.0 MeV·cm^2/mg 1.8×10^6 fluence/day

Usually, upsets→0 with LET< 0.1 MeV·cm^2/mg.

As LET_{TH} decreases, error rate increases.

Observe how low flux is at high LET values.


Solar Max 100 mils

0.1..1.0 MeV·cm^2/mg 8.0×10^3 fluence/day

Test to high enough fluences for your target environment. The conventional 1e7 ions/cm^2 is no longer sufficient for COTs devices.

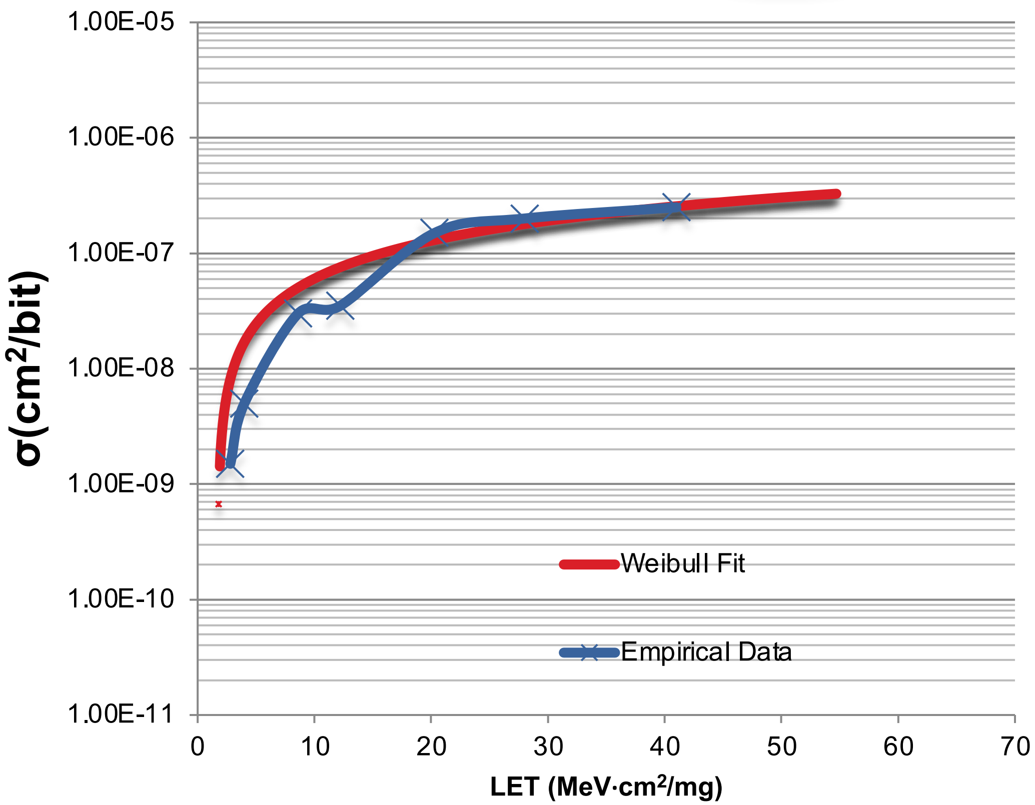
Deriving $\sigma(L)$:

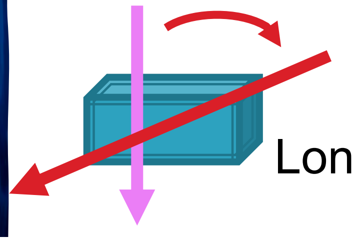
SEU Cross Sections Across LET Spectrum

- Obtain discrete σ_{SEU} data from accelerated testing.
- Fit the data to create a continuous representation. Most tools require Weibull fitting parameters:

$$\sigma(LET) = \sigma_{SAT}(1 - e^{\left[\frac{LET-LET_0}{W}\right]^s})$$

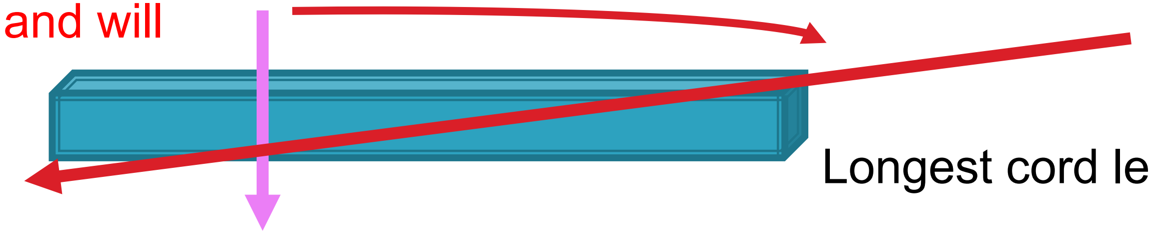
  - σ_{SAT} = limiting or plateau cross-section (saturated cross-section);
  - LET_0 = onset parameter, such that σ(LET) = 0 for LET < LET_0;
  - W = width parameter;
  - s = a dimensionless exponent.
- RPP requires x, y, z to determine LET effective cord lengths.
- We do not have σ continuous angular data (cover the volume). LET cord lengths are used by the tools to derive σ angular contributions.
- **Questionable how good the derivations are.**





Allowable angles affect cord length and will affect derived σ

Longest cord length

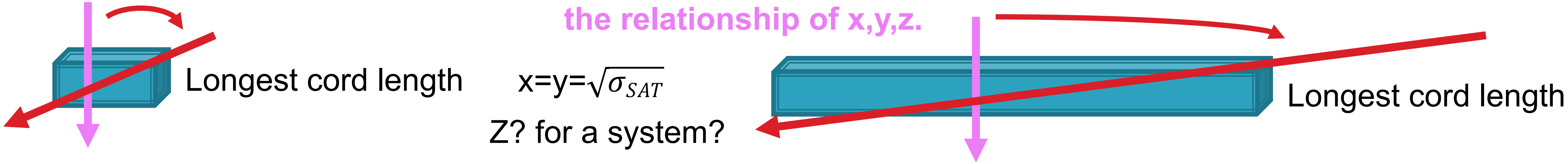


Longest cord length

The Impact of System-Level Cross Sections To RPP Derived Angular Effects

- RPP requires x, y, z to determine LET effective cord lengths to derive angular effects:
  - Z is left for the user to select:
    - The relationship of Z to X and Y can greatly impact the derived cross section error rate contributions.
    - It is important to note that Z is usually an unknown entity.
  - x=y=$\sqrt{\sigma_{SAT}}$ is not in any way a proper model for a system. System error rates are not area dependent… system faults are **functionally dependent,** and faults are **not all caused by homogenous components**.
- Because of all the above, system-level cross-section data are not producing accurate error rates with our current tools.
- This is because the tools were not developed to calculate at the system-level.

As COS(θ) increases, σ increase (so we assume). COS can approach infinity depending on the relationship of x,y,z.



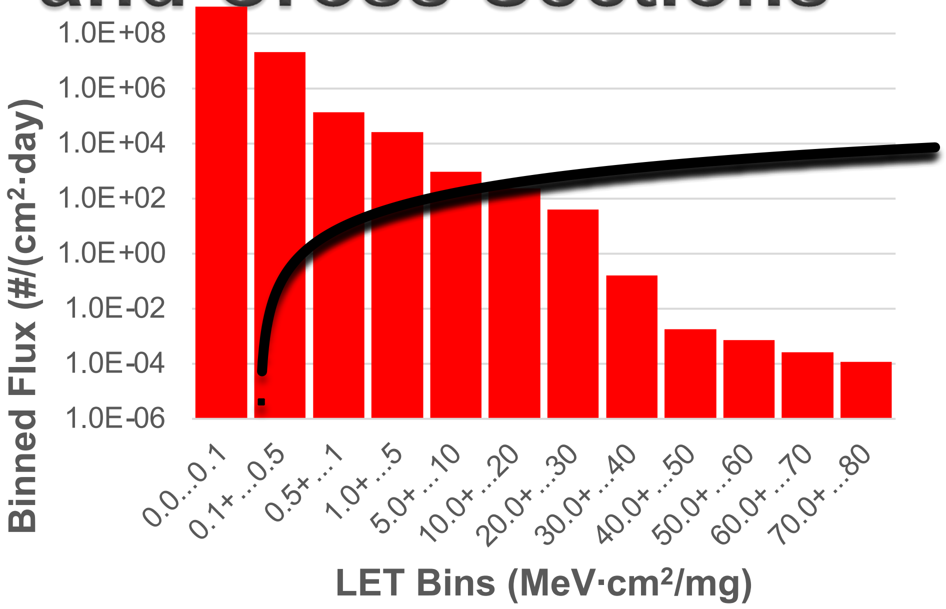
Longest cord length

x=y=$\sqrt{\sigma_{SAT}}$

Z? for a system?

Longest cord length

Error Rate Calculations: Integration of Flux and Cross Sections



$$R_h = \int_0^\infty f(L)\sigma(L)dL$$

Transformation to numerical integration

$$R_h = \lim_{\Delta L \to 0} \sum_{L=0}^{NB} f(L) * \sigma(L)\Delta L$$

$$\sigma_{SEF} = \frac{\#events}{\#ions/_{cm^2}} < 1$$

Questions to be addressed as we move forward?

- Do we need a Weibull fit?
  - Can we use our empirical data with rectangular or trapezoidal numerical integration?
  - Each step will have some over estimation. Will this cover potential angular effects?
- Are angular effects at the system level real or are they insignificant?

The advancements of technology enables enhances capabilities with new complexities.

Conventional SEE testing techniques, models, and data analysis do not apply to today’s complex architectures.

It is essential for SEE test and analysis of radiation components (such as FPGAs) to modernize.

Melanie.D.Berg@spacer2.com

Thank You…

Questions