LA-UR-

Approved for public release;
distribution is unlimited.

Title:	
Author(s):	
Intended for:	

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

Form 836 (7/06)

FMAC
FPGA Mission Assurance Center

Outside of Normal Operating Conditions: Using Commercial Hardware in Space Computing Platforms for Ubiquitous Sensing

Heather Quinn

Los Alamos National Laboratory

Operated by Los Alamos National Security, LLC for NNSA

LA-UR-08-05799

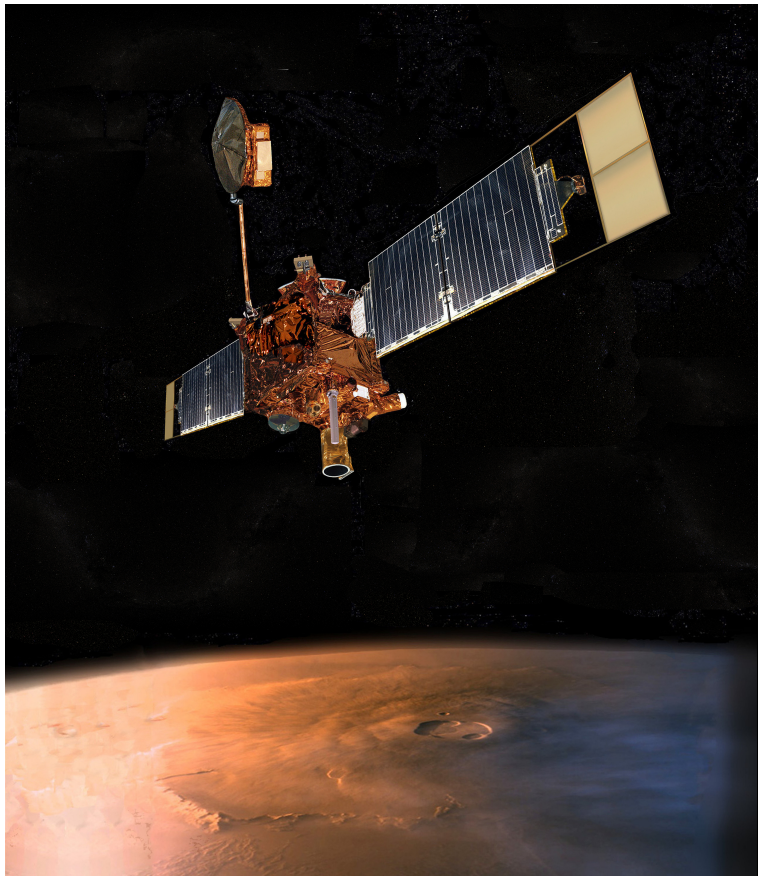
NNSA

Overview

- National Security, Ubiquitous Sensing, and Commercial Technology in Space
- Basic Radiation Effects in CMOS
- Failure Modes in SRAM FPGAs from Radiation-Induced Faults
- Case Study of One Reliability Concern: Logical Constants
- Mitigation and Repair Methods for SRAM FPGAs
- Conclusions

Sensing Applications for National Security

- **In recent years, much of LANL's mission has focused on persistence surveillance of targets and interests to provide an overall reduction in threats to the US**
- **Data is collected from a number of platforms: distributed sensor networks (DSNs), airplanes, unmanned aerial vehicles (UAVs), and satellites**
- **This data plays an important role in national security and policy decisions**
- **The collected data is from a number of sensor types: imagery, seismic, radiation, temperature, radio frequency**
- **Many of these sensors grew out of science programs**
  - Satellite-based detectors that could sense neutrons in the ground have been used to determine whether there is water on Mars and whether there is nuclear proliferation



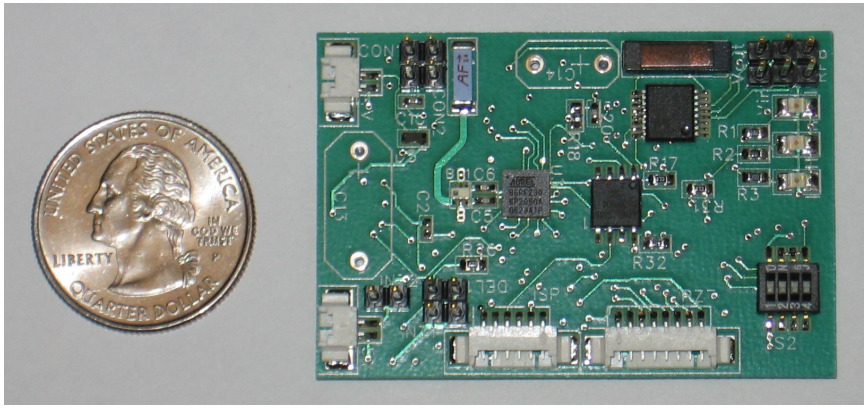
http://mars.jpl.nasa.gov/mgs/gallery/images/mgs-mons.jpg

Transitioning from local to ubiquitous surveillance

- **The lab is striving for a global reduction of threats**
- **The lab's mission is to grow our sensing capabilities so that we could provide constant, global – ubiquitous – surveillance**
  - Increasing the view of our sensing capabilities provides more information, giving us global coverage
  - Increasing the sensitivity of our sensing capabilities provides more accurate information
  - Increasing the number and types of surveilling platforms to provide options for collecting data
- **For example, cameras can differ in the field of view, resolution, sample rate, and can be placed on a number of different platforms to increase global coverage.**
- **The better, the wider, the more proliferate our sensing capabilities are, the less likely we are to miss important events around the world**

Examples of Ubiquitous Sensing

- **DSNs:**
  - Smart paint that can monitor the integrity of physical infrastructure, such as buildings or bridges
  - Intelligent rocks that can monitor the movement of radioactive materials on highways
- **UAVs:**
  - Wide area persistence imagery that can track movement through cities
- **Satellites:**
  - Neutron detectors that can globally monitor the adherence to the Nuclear Test Ban Treaty.
  - Imagery that can globally monitor whether nuclear plants are being built that could be later disguised



http://int.lanl.gov/news/index.php/fuseaction/home.story/story_id/11142

Tactical vs. Strategic Applications

- **In the preceding cases, ubiquitous sensors collect data that can help analysts determine whether a bridge could collapse, whether a country is proliferating nuclear weapons, or whether a warfighter is about to enter a dangerous territory.**
  - How quickly can someone get information so that the situation can be arrested?
- **Tactical applications have to be able to provide information to the end-user in real time**
  - Short term solutions for near term problems
  - Data needs to be collected and disseminated to the end-user in the field as needed
  - The sensor resolution and any computationally aided analysis are often tuned (down) to meet the time demands of the application
- **Strategic applications could take days, weeks or months to reach the end-user**
  - Long term solutions for a variety of problems and long term threats
  - Without real-time constraints, more complicated computational analysis is possible
  - The sensor resolution is usually much higher than tactical sensors and the sensor platforms are often “exquisite solutions”

Challenges of ubiquitous sensing

- **Designing wide-area, extremely sensitive sensors is challenging**
  - Done with one, expensive and expansive sensor or tons of less expensive, less capable sensors?
  - How to blend different sensor types and capabilities?
- **Wide area, constant surveillance stresses computation and communication systems**
  - Do you need to trade off computation for communication?
- **The amount of data collected from these efforts presents many challenges**
  - How do you automate data collection and extract “actionable” information?
    - How can you detect anomalies in information?
  - How can you fuse data sets to provide more information?
    - Can data from another type of sensor confirm the extracted information?

Capabilities needed for ubiquitous sensing

- **Sensors:**
  - Extremely sensitive
  - Wide range
- **Communication systems:**
  - Wide downlinks
  - Efficient communication
  - Methods for prioritizing data retrieval
- **Computational systems:**
  - Robust, autonomous control of deployed systems
  - Methods for saving power, including selectively powering down deployed systems
  - In-situ processing of data to return information instead of data

Increasing On-Board Computational Processing for Space-based Remote Sensing

- **Use commercial-based technologies for high performance portions of the space systems**
  - Leverage billions of dollars of world-wide commercial investment in semiconductor technology
  - Employ well-tested technologies with large user bases rather than unique space solutions
  - Exploit inherent radiation tolerance (e.g., total ionizing dose) of these devices
- **Use system-level, module-level, and application-level engineering to provide the robustness needed for the system (don't "over-engineer" systems)**
  - Employ an excellent understanding of both mission and technologies
  - Employ existing and new mitigation techniques to add robustness: e.g., redundancy, repair, and reconfiguration
- **Use more conventional radiation-hardened technologies in high-risk portions of the system or where performance and cost are not drivers**
  - Spacecraft interfaces
  - Critical non-volatile memory

SRAM-based FPGAs in Space

- **Many organizations have started using commercial SRAM-based FPGAs in space-based computing platforms**
  - Well-suited to DSP-oriented satellites
  - Custom hardware speedups without the cost of manufacturing an ASIC
  - Reconfigurability can extend the useful lifetime of the system by allowing the system to reconfigure to meet changing mission or science needs
- **Much of the why organizations have been putting FPGAs in space is due to research done at BYU and LANL**
  - Radiation tests to characterize how FPGAs will work on orbit
  - Fault injection tools for emulating radiation-induced faults in designs
  - Automated tools for applying redundancy-based mitigation to FPGA designs
  - Automated tools for mitigating architectural-level problems on FPGAs (logical constants)
  - Tools for assessing design reliability
- **To date we have had 11 students intern at LANL, support students through CHREC, and have hired three alumni**
  - We would like to continue this relationship!

The Space Radiation Environment

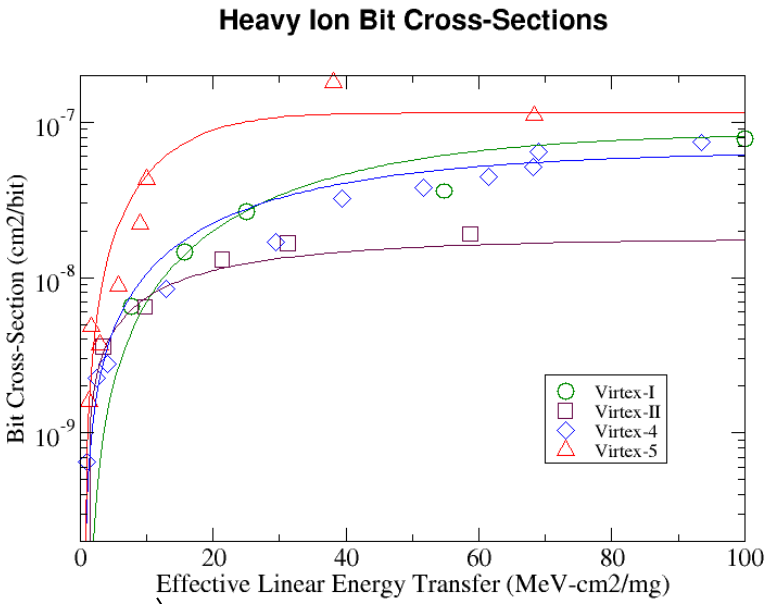
- **The space radiation environment has electrons, protons, and heavy ions**
  - Each orbit has it's own spectrum, each ion it's own energy spectrum
  - The flux is affected by the 11-year solar cycle and solar events
  - The radiation environment is very dynamic
- **The radiation environment affects the electronics by degrading transistors or creating transient errors**
- **For SRAM FPGAs both the circuit and the circuit's state are affected by the radiation environment**

Basic Radiation Effects on Semiconductor Devices

- **<u>Total ionizing dose (TID)</u>: protons and electrons degrade the voltage and switching characteristics of the transistors**
- **<u>Single-event effects (SEEs)</u>:**
  - Single-event latchup (SEL): radiation causes the parasitic transistors to turn on and destroy the device through excessive current draw
  - Single-event upsets (SEUs): radiation causes an SRAM bit to change values (0→1 or 1→0)
  - Single-event functional interrupts (SEFIs): radiation causes an SEU that makes the device unusable until it is reset
  - Single-event transients (SETs): radiation causes a transient current state in logic, which could affect the circuit's state if latched into a flip-flop
- **Predominant concern with Xilinx Virtex devices are SEUs and SEFIs**

Xilinx Virtex SEE Data

5-7 orders of magnitude difference in SEU bit cross-sections



Proton Cross-sections

Device	Energy (MeV)	σ_{bit} (cm^2/bit)	σ_{SEFI} (cm^2/device)
XCV1000	63.3	1.32×10^{-14}	$\sim 7.1 \times 10^{-13}$ (config SEFI)
XC2V1000	63.3	2.1×10^{-14}	9.46×10^{-13}
XC4VLX25	63.3	1.08×10^{-14}	6.43×10^{-12}
XC5VLX50	65.0	7.56×10^{-14}	Unknown

Not a consistent decrease across devices

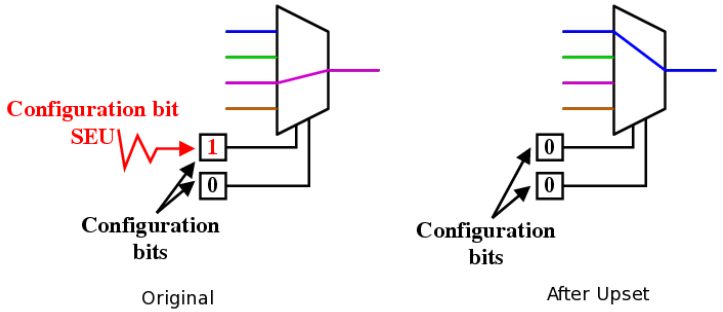
SEFIs ~size of 100-1000 bits upset

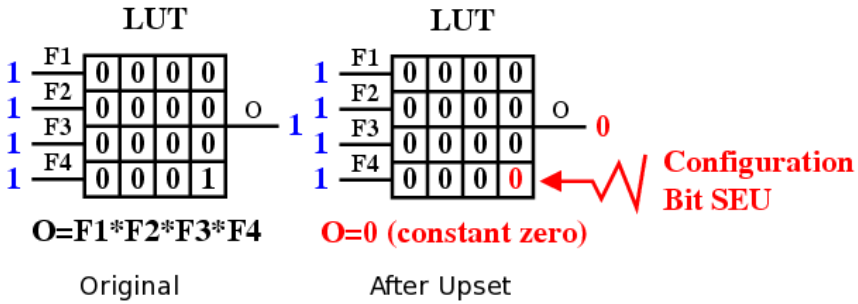
Failure Modes from SEUs and SEFIs

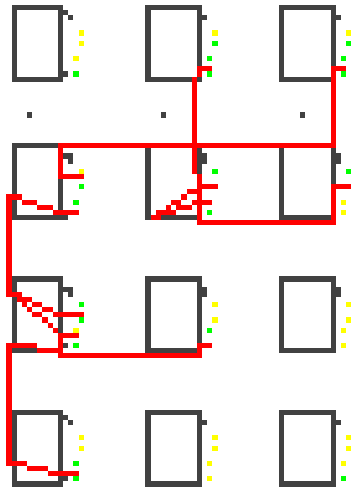
- **Three main classes:**
  - Failures that affect circuit functionality
  - Failures that affect circuit state
  - Failures that affect device functionality

Failures that Affect Circuit Functionality

- **Routing Vulnerabilities**
  - Mux select lines change values
  - Pips and buffers open or short
- **Logic Vulnerabilities**
  - LUT value changes
  - LUT control bit changes
- **Tie-off Vulnerabilities**
  - Implicit logic constants: half-latches
  - Explicit logic constants: constant LUTs and VCC posts







Failures that Affect Circuit State

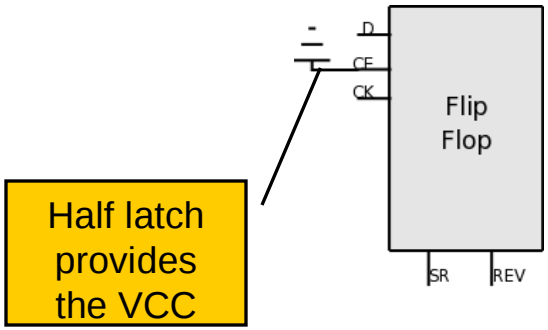
- **Maintaining state is difficult**
  - SEUs in circuit functionality can affect state
  - SEUs in user memory (flip-flops, BlockRAM) can affect state
- **Routing of global signals is particularly vulnerable**
  - Clock and reset trees provide a large target for SEUs
  - Most common method for handling logical constants elevates them to global signals

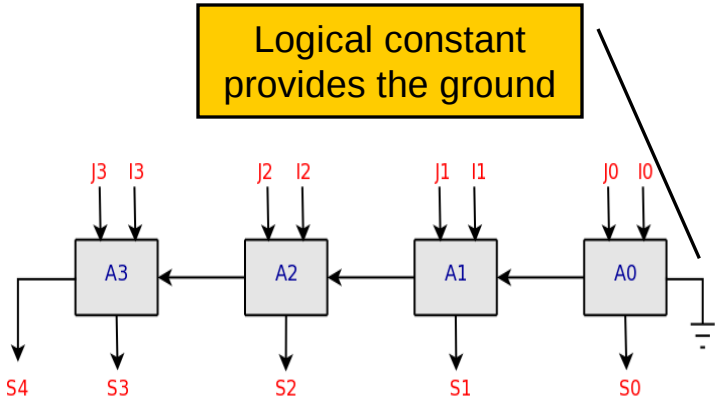
Failures that Affect Device Functionality

- **SEFIs that affect all devices:**
  - JTAG TAP Controller: Many failure modes
  - SelectMAP: unable to read from or write to SelectMAP interface
  - Power-on-reset: configuration is cleared and DONE pin is driven low
- **Virtex-4 specific SEFIs:**
  - FAR and Readback SEFIs that mimic SelectMAP SEFIs
  - Global Signal SEFI: umbrella SEFI that covers SEFIs in Global Set/Reset, Global Write Enable and Global Drive High signals
  - Scrub SEFI: SEU in the control logic while performing on-line reconfiguration, causes a high current state

Case Study: Logical Constants

- **Logical constants are needed to generate constant zero and one logic values used internally by FPGA designs**
  - Artifact of mapping VHDL designs to the specific FPGA architecture
  - Not under design-control, unless the designer is going to extraordinary measures to avoid them in VHDL/Verilog
  - Easy to mitigate at either the EDIF or XDL level
- **"Implicit" logical constants**
  - Inputs to I/O, logic, RAM, clocking, and other resources
  - Implemented in half latches (weak keepers)
- **"Explicit" logical constants:**
  - Tie-offs to the zeroth bit of the carry chain for adders and unused multiplier/DSP inputs
  - Implemented as constant LUTs in the Virtex-I and Virtex-II, implemented as architectural posts in the Virtex-4



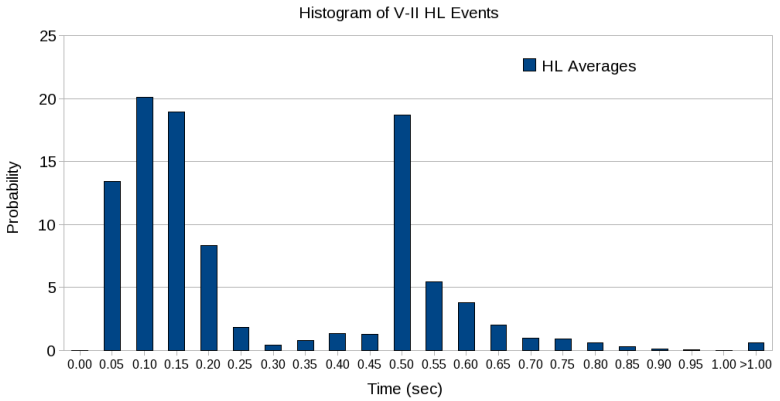


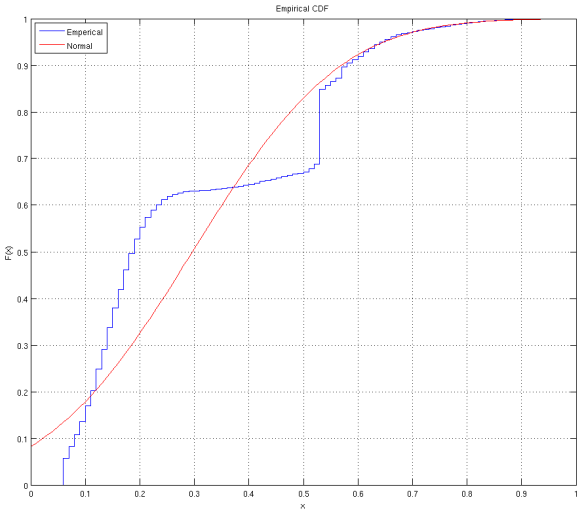
Reliability Concerns with Half Latches

- **Half latches**
  - Are not directly observable though readback and not scrubbable through on-line reconfiguration
  - When upset, the tie-off does not operate properly until reset through off-line reconfiguration or by leaking off
  - In the Virtex-I, half latches did not leak off
  - In the newer devices, the weak keeper circuit will leak off
- **Half latch data:**
  - Virtex-I: Graham, P. et al., “SEU mitigation for half-latches in Xilinx Virtex FPGAs,” *IEEE Transactions on Nuclear Science*, Vol. 50, No. 6, December 2003, pp. 2139 – 2146.
  - Virtex-II, Virtex-II Pro, Virtex-4: follows….

Half Latch Data:
Virtex-II

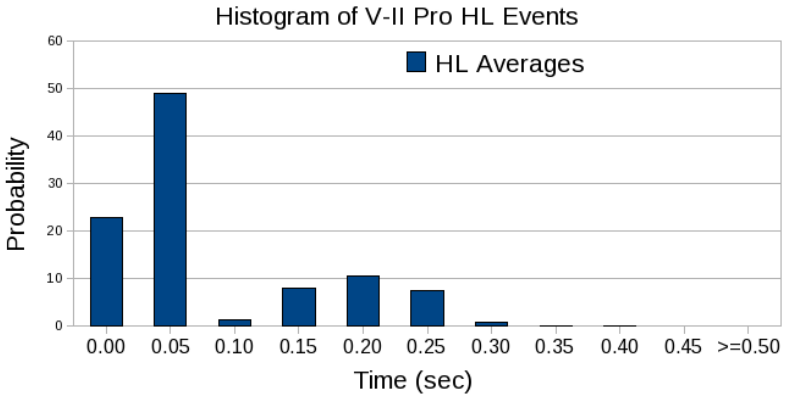
- **Two modes in the data**
  - Peak at 0.13 secs has standard normal distribution shape
  - Peak at 0.53 secs is an impulse function, as seen in the CDF
  - Could be a contamination of data, two types of half latch designs, or constant LUTs
- **The average time that a HL holds is 0.30 secs for the entire data set with a standard deviation of 0.21 secs such that 68% of all half latches leak off within 0.09-0.51 secs**
- **On average 99% of all half latches leak off within 1 second**

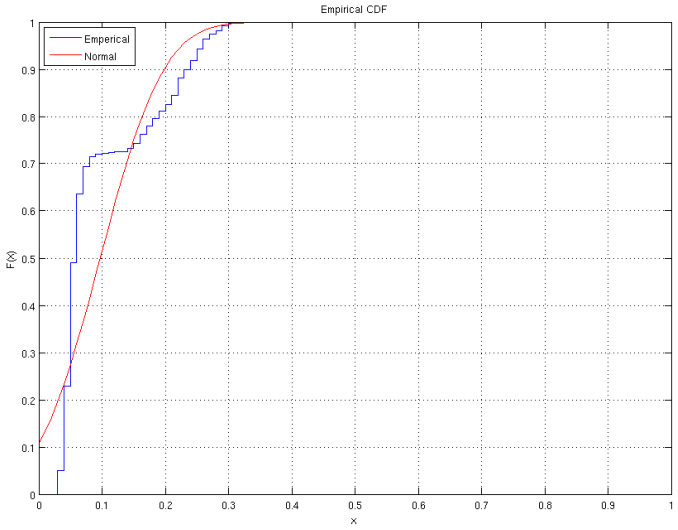




Half Latch Data:
Virtex-II Pro

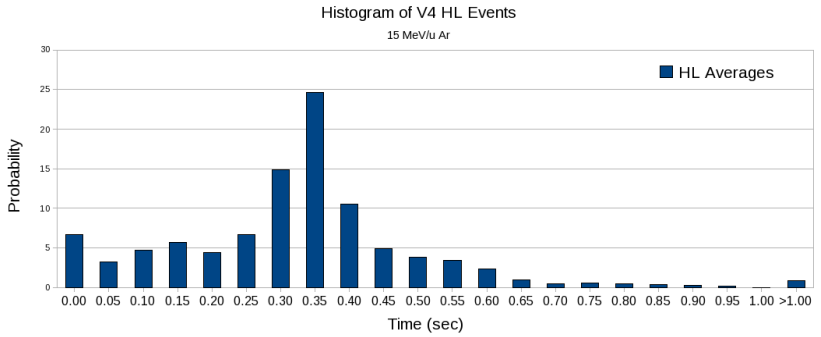
- **Two modes in the data**
  - Peak at 0.05 secs has standard normal distribution shape
  - Peak at 0.22 secs has a standard normal distribution shape
  - Could be a contamination of data, two types of half latch designs, or constant LUTs
- **The average time that a HL holds is 0.10 secs for the entire data set with a standard deviation of 0.08 secs such that 68% of all half latches leak off within 0.02-0.18 secs**
- **On average 100% of all half latches leak off within 0.45 secs**

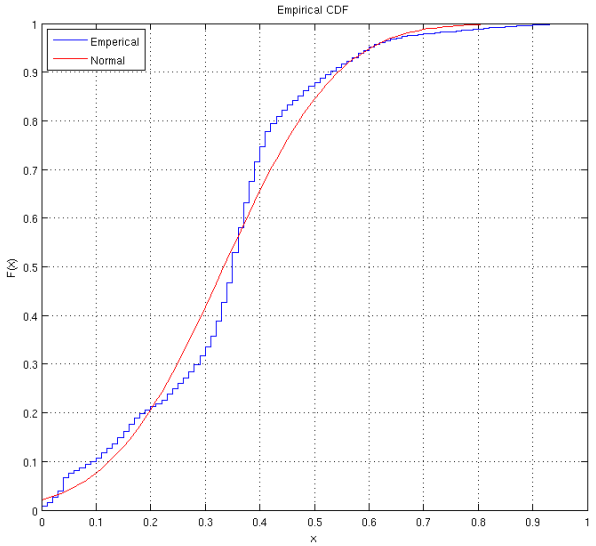




Half Latch Data:
Virtex-4

- **Single mode in the data**
  - Peak at 0.35 secs has standard normal distribution shape with very large tails
- **The average time that a HL holds is 0.33 secs for the entire data set with a standard deviation of 0.16 secs such that 68% of all half latches leak off within 0.17-0.49 secs**
- **On average 99% of all half latches leak off within 1 sec**



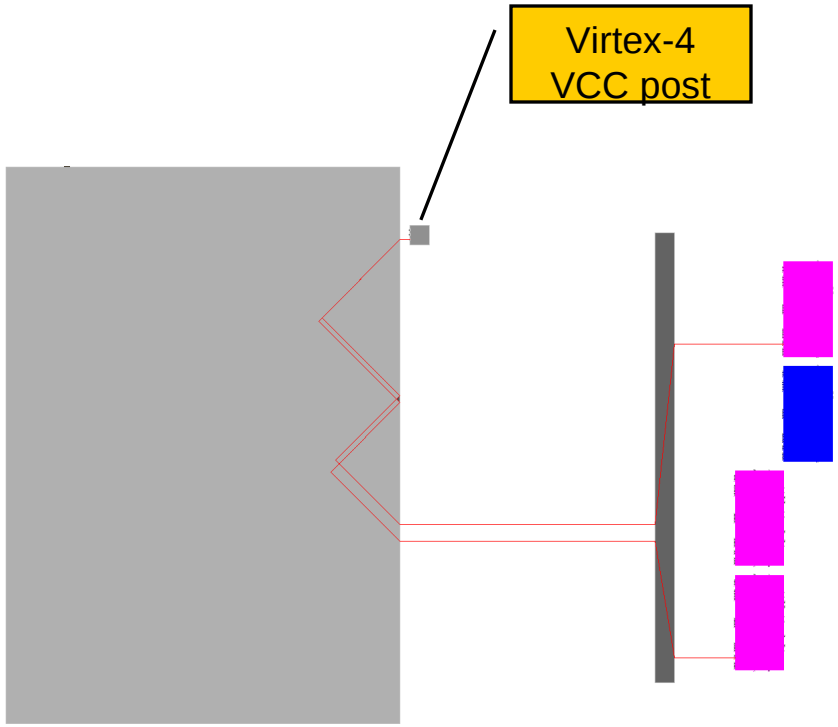


Reliability Concerns with Constant LUTs

- **Constant LUTs**
  - Are directly observable through readback and can be scrubbed through on-line reconfiguration
  - Tie off directly affects the data stream by injecting bad data into adders and multipliers
  - Design flow tools load balance the use of constant LUTs and can cause single points of failures in TMR-protected designs

Reliability Concerns with Architectural Posts

- **Architectural Posts**
  - In the Virtex-II, these posts are an abstraction of a half latch and have the reliability concerns of a half latch
  - In the Virtex-4, not certain what the post is and is still under investigation
    - Uncertain whether configuration bits are used in conjunction with the post
    - Single-bit problems with Virtex-4 do not seem to be post-related, but possibly single-bit domain crossing errors

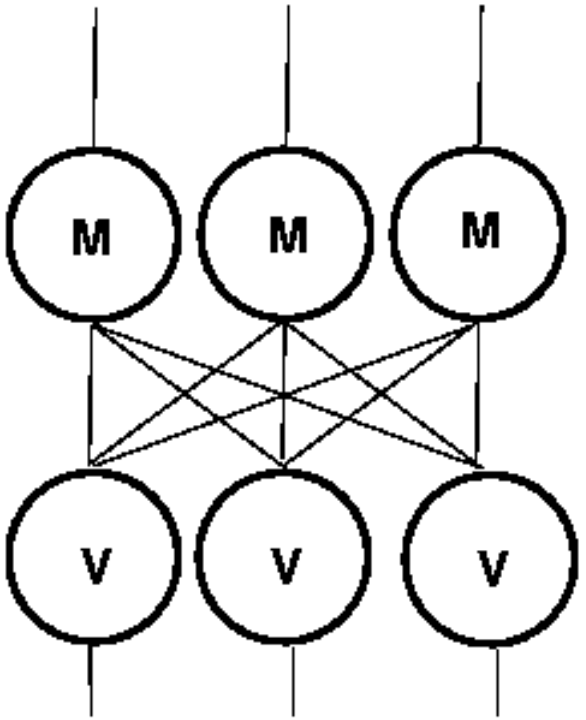


Mitigation and Repair Methods

- **Even a single SEU can cause the circuit to output bad data**
- **Accumulating SEUs increase the likelihood that output data is corrupted and increase device's current draw**
- **Mitigation and repair of SEUs is needed**
  - To date, best option for mitigation SEUs is to mask them through triple-modular redundancy (TMR)
  - Logical constants can be mitigated through a couple different methods
  - On-line reconfiguration, called *scrubbing*, used to remove SEUs
  - Off-line reconfiguration used to remove SEFIs

Triple-Modular Redundancy (1 of 2)

- **Original concept from von Neumann in 1956**
- **Best practice for space usage of SRAM FPGAs recommends triplicated I/O, global signals, logic and voters**
  - Not triplicating global signals can make TMR-protected signals less reliable than unprotected circuits
  - Masking multiple-bits upsets and multiple-independent upsets is not guaranteed

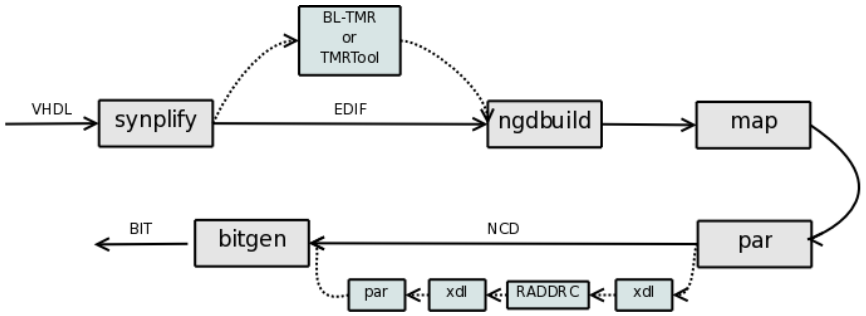


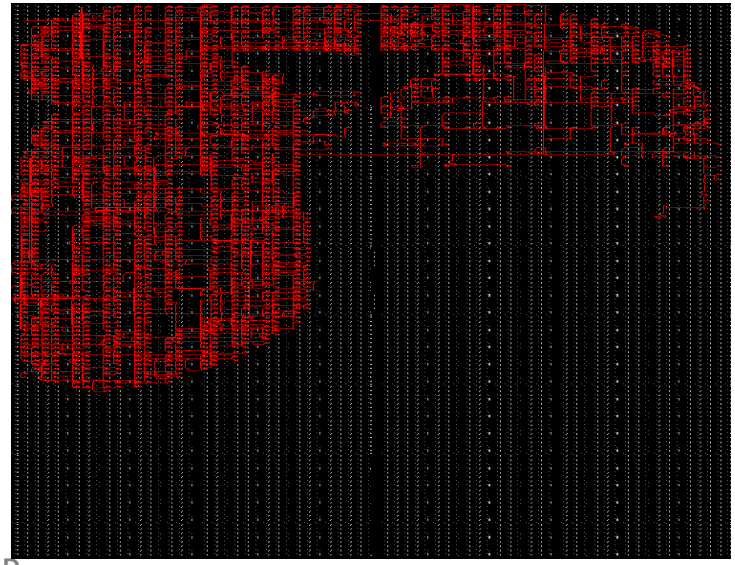
Triple-Modular Redundancy (2 of 2)

- **Applying TMR is not simple; use an automated tool**
  - Xilinx's TMR Tool
  - BYU's and LANL's BL-TMR
- **TMR-protected design should be tested to ensure the circuit meets availability requirements**
- **Three-tiered testing methodology:**
  - Modeling tools: LANL's Scalable Tool for the Analysis of Reliable Circuits (STARC) estimates the hardness assurance issues in the design's EDIF
  - Fault injection tools: BYU's and LANL's SEU Emulator provides a uniform method for injecting artificial SEUs to locate *sensitive bits* in the design that cause output errors
  - Radiation testing at a particle accelerator: final validation of the circuit

Mitigating Half Latches

- **Half latch extraction is available from the RADDRC I and II tools (LANL), BL-TMR tool (BYU-LANL), and TMRTool (Xilinx)**
- **RADDRC I and II:**
  - Half latches are extracted to constant LUTs in XDL
  - Constant LUTs are observable and in this case should not become meshed
- **BL-TMR and TMRTool**
  - Half latches are extracted to constant input pins in EDIF
  - Input pins need to be triplicated, otherwise large cross-section
- **Accelerator testing indicates there is a complete elimination of the half latch cross-section with all methods**





Mitigating Shared Constant LUTs

- **RADDRC II tool (LANL)**
  - Eliminates shared constant LUTs in XDL – works as with previous flow example
  - Each LUT that is sourcing multiple constants is duplicated and the enmeshed network is separated into single-source constant LUTs
  - Is possible to guide the placement with the previous ncd file to minimize the change to the circuit's placement
- **Fault injection testing indicates there is a complete elimination of the shared constant LUT cross-section**

Scrubbing

- **One recommended algorithm:**
  1. Readback the configuration data
  2. Complete a CRC check for each configuration data frame
  3. If the CRC value does not match, scrub the frame
- **SEFIs cannot be scrubbed**
  - Many SEIF signatures look like a number of frames that fail the CRC check
  - Only recovery for a SEFI is a full, off-line reconfiguration of the device
- **LUT RAM, SRL16s, and BlockRAM cannot be scrubbed without corrupting user data**
  - Virtex-I and Virtex-II: scrubbing circuit must “skip” these resources
  - Virtex-4: on-device hardware skips LUT RAM and SRL16s
  - If BlockRAM is used as a ROM, it can be scrubbed using Xilinx’s BlockRAM scrubber
- **“Blind” scrubbing (without readback) is not recommended for the Virtex-4 due to Scrub SEFI**

Conclusions

- **Ubiquitous sensing is an important aspect of national security and reducing global threats**
  - The amount of data collected drives the need for more efficient computational and communication systems
- **A number of radiation-induced faults in SRAM FPGAs were presented**
  - These faults affect the circuit's function, circuit's state, and the device's functionality
- **Mitigation tools are available to automatically apply TMR to mask SEUs and to extract logical constants, improving the reliability of FPGA user designs**
- **Scrubbing can remove SEUs**