

Optimizing Mitigation Strategies for FPGA Critical Applications



Melanie Berg

MEI Technologies- NASA/GSFC Radiation Effects and Analysis Group
Melanie.D.Berg@NASA.gov

What's the Issue?



If something goes wrong...



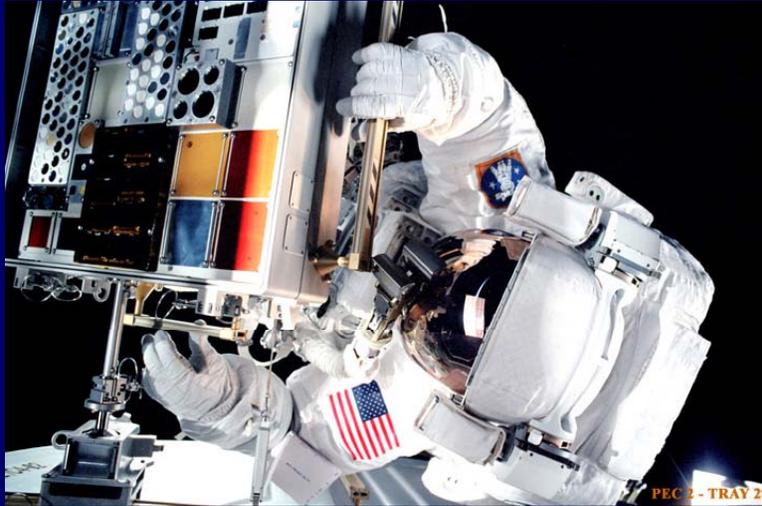
**Increasing number of
FPGA devices inserted
into space missions**

**Harsh Space
Radiation
Environment**

Presented by Melanie Berg MAPLD Albuquerque, New Mexico November 1, 2010

Page 2

We Can't Always do This...



Presented by Melanie Berg MAPLD Albuquerque, New Mexico November 1, 2010

Page 3

Agenda

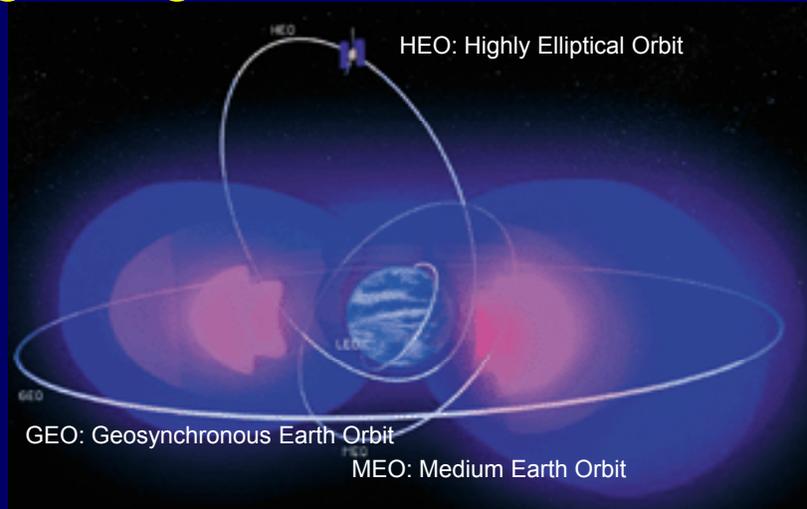


- **Section I:** Single Event Effects in Digital Logic
- **Section II:** FPGA Basics – Architectural Differences
- **Section III:** Reducing System Error: Common Mitigation Techniques
 - Triple Modular Redundancy:
 - Block Triple Modular Redundancy (BTMR)
 - Local Triple Modular Redundancy (LTMR)
 - Global Triple Modular Redundancy (GTMR)
- **Section IV:** The Automation Process and the Mentor Graphics Advantage

Presented by Melanie Berg MAPLD Albuquerque, New Mexico November 1, 2010

Page 4

Section I: Single Event Effects in Digital Logic



HEO: Highly Elliptical Orbit

GEO: Geosynchronous Earth Orbit

MEO: Medium Earth Orbit

Van Allen Radiation Belts:

Illustrated by Aerospace Corp.

Presented by Melanie Berg MAPLD Albuquerque, New Mexico November 1, 2010

Page 5

Source of Faults: SEEs and Ionizing Particles



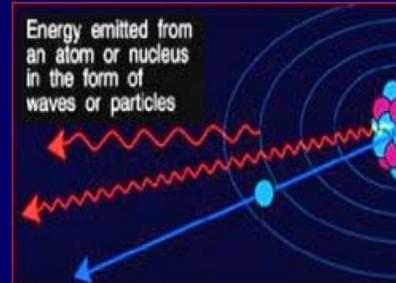
• Single Event Effects (SEEs)

- Terrestrial devices are susceptible to faults mostly due to:

- **alpha particles:** from packaging and doping and
- **Neutrons:** caused by Galactic Cosmic Ray (GCR) Interactions that enter into the earth's atmosphere.

- Devices expected to operate at higher altitude (Aerospace and Military) are more prone to upsets caused by:

- **Heavy ions:** direct ionization
- **Protons:** secondary effects



Energy emitted from an atom or nucleus in the form of waves or particles

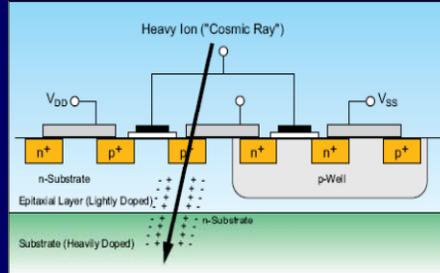
Presented by Melanie Berg MAPLD Albuquerque, New Mexico November 1, 2010

Page 6

Device Penetration of Heavy Ions and Linear Energy Transfer (LET)



- LET characterizes the deposition of charged particles
- Based on Average energy loss per unit path length (stopping power)
- Mass is used to normalize LET to the target material



$$LET = \frac{1}{\rho} \frac{dE}{dx} \cdot \text{MeV} \frac{\text{cm}^2}{\text{mg}}$$

Density of target material
Units
Average energy deposited per unit path length

LET vs. Error Cross Section Graph

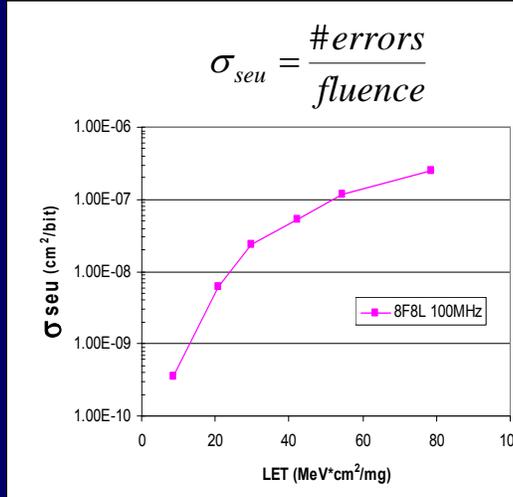


Error Cross Sections are calculated per LET value in order to characterize the number of potential faults and error rates in the space environment

Terminology:

- Flux: Particles/(sec-cm²)
- Fluence: Particles/cm²
- Error cross section(σ): #errors normalized by fluence
- Error cross section is calculated at several LET values (particle spectrum)

LET vs. σ:



Single Event Faults and Common Terminology



- Single Event Latch Up (**SEL**): Device latches in high current state
- Single Event Burnout (**SEB**): Device draws high current and burns out
- Single Event Gate Rupture: (**SEGR**): Gate destroyed typically in power MOSFETs
- **Single Event Transient** (SET): current spike due to ionization. Dissipates through bulk
- **Single Event Upset** (SEU): transient is caught by a memory element
- **Single Event Functional Interrupt** (SEFI) - upset disrupts function

Presented by Melanie Berg MAPLD Albuquerque, New Mexico November 1, 2010

Page 9

Single Event Effects (SEEs) and IC System Error



- **SEUs or SETs can occur in:**
 - Combinatorial Logic (including global routes)
 - Sequential Logic
 - Memory Cells
- Depending on the Device and the design, each fault type will:
 - Have a probability of occurrence
 - Either have a significant or insignificant contribution to system error

Every Device has different Error Responses – We must understand the differences and design appropriately

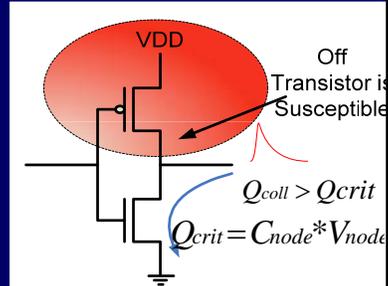
Presented by Melanie Berg MAPLD Albuquerque, New Mexico November 1, 2010

Page 10

Radiation Induced Fault Generation



- SETs can vary in pulse width (T_{pulse}) and amplitude.
- Different FPGA processes and geometries will have different sensitivities

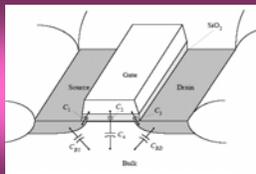


CAPACITANCE

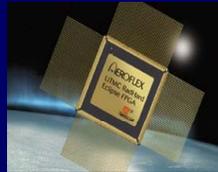
Transistor Cutoff frequencies

$$f_c = \frac{1}{2\pi RC}$$

Each capacitance has its own f_c



- Geometry of Transistors
- Loading of Transistors
- Length of Routes
- Switching Rates



Section II: FPGA Basics – Architectural Differences

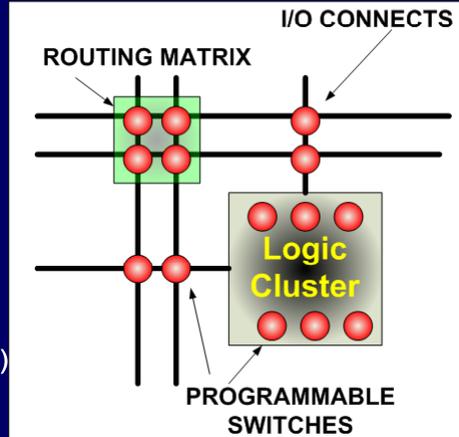


FPGA Configuration



HDL **FPGA MAPPING** Configuration

- **Configuration Defines:**
Arrangement of pre-existing logic via programmable switches
 - Functionality (logic cluster)
 - Connectivity (routes)
- **Programming Switch Types:**
 - **Antifuse:** One time Programmable (OTP)
 - **SRAM:** Reprogrammable (RP)
 - **Flash:** Reprogrammable (RP)

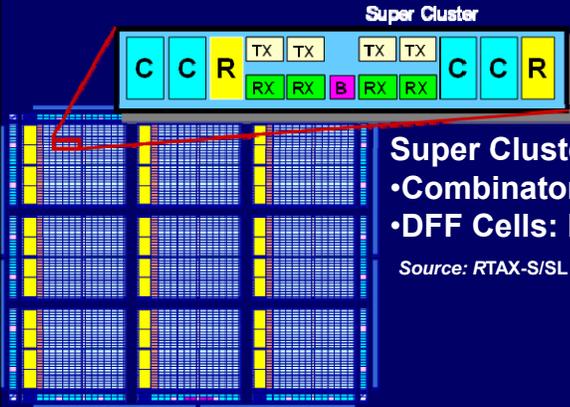


Antifuse FPGA Devices



- Currently the most widely employed FPGA Devices within space applications
- Configuration is hardened due to fuse based technology (Metal to Metal)
- Localized (@ DFF node) Mitigation (TMR or DICE) is employed
- Clock and Reset lines are hardened

ACTEL RTAX-S Architecture Basics

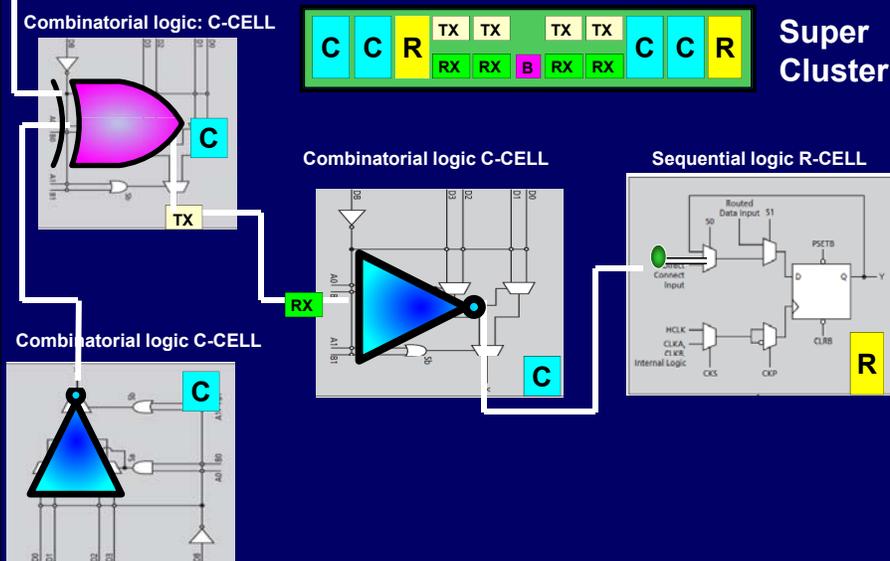


Super Cluster:

- Combinatorial Cells: C CELLS
- DFF Cells: R Cells

Source: RTAX-S/SL RadTolerant FPGAs 2009 Actel.com

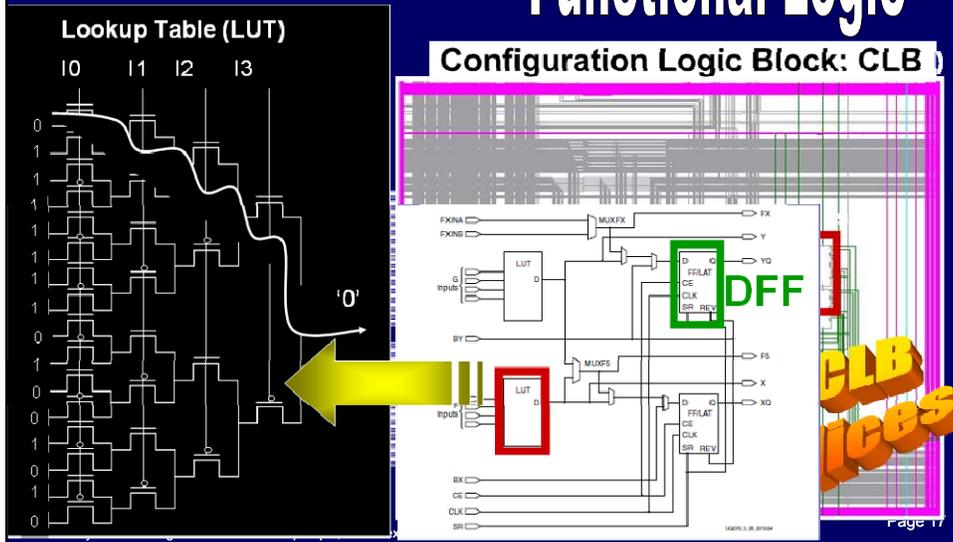
ACTEL RTAX-S Combinatorial and Sequential Logic



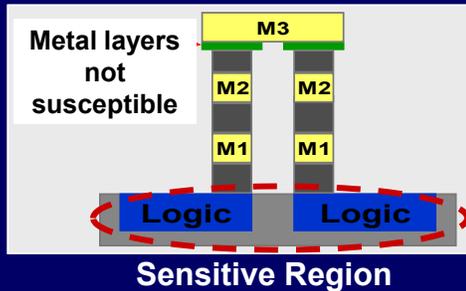
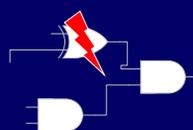
General Xilinx Virtex 4 FPGA Architecture: SRAM Based Configuration



Functional Logic

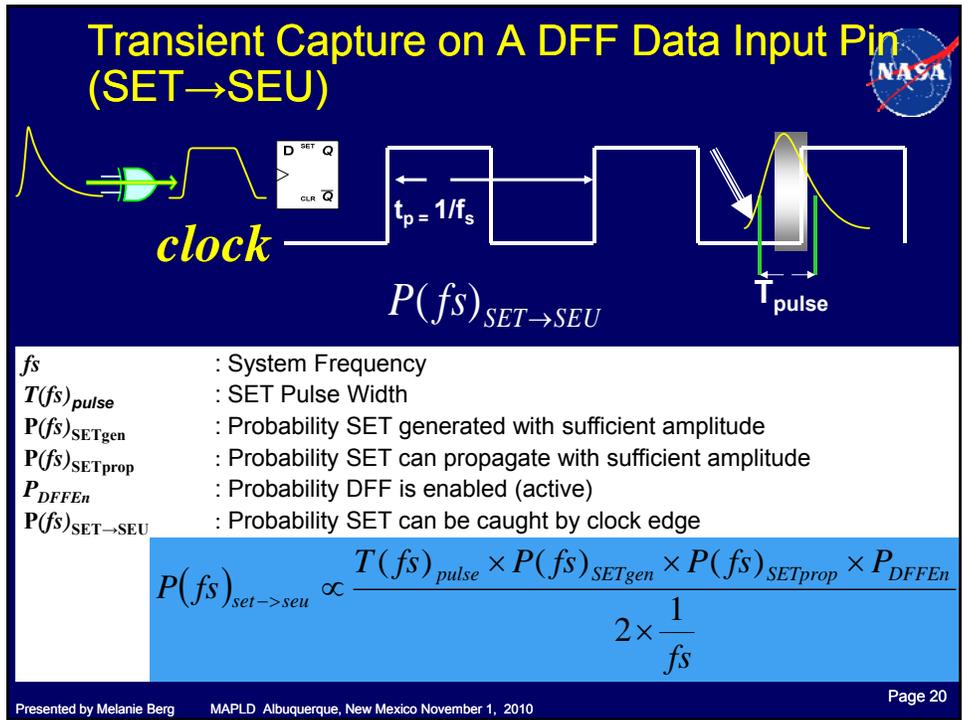
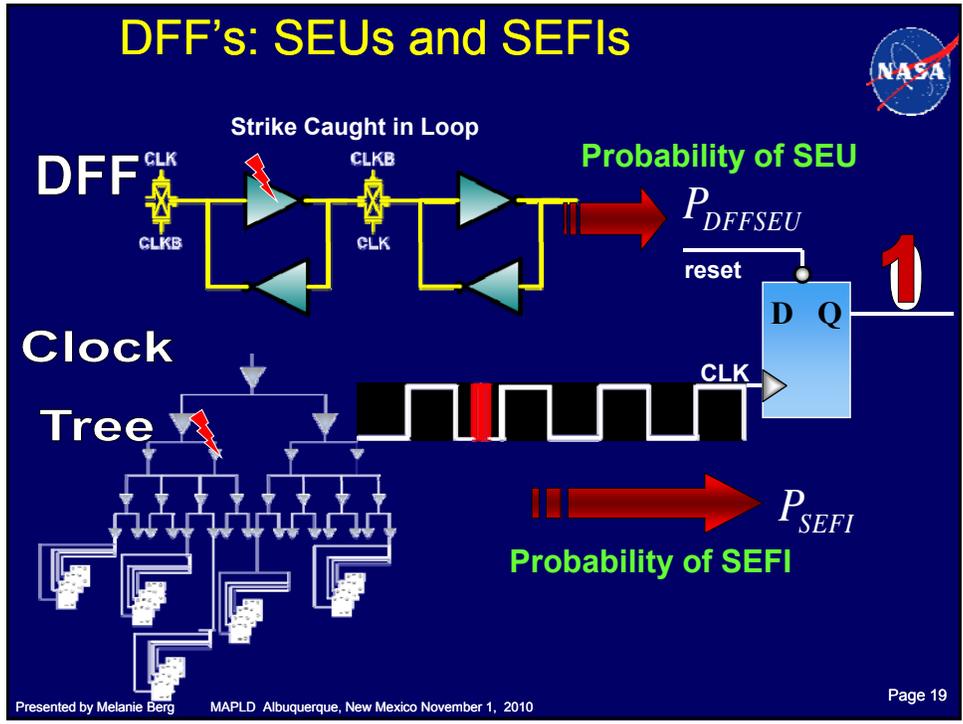


Combinatorial Logic Blocks and Potential Upsets... SETs in ASICs and Anti-fuse FPGAs

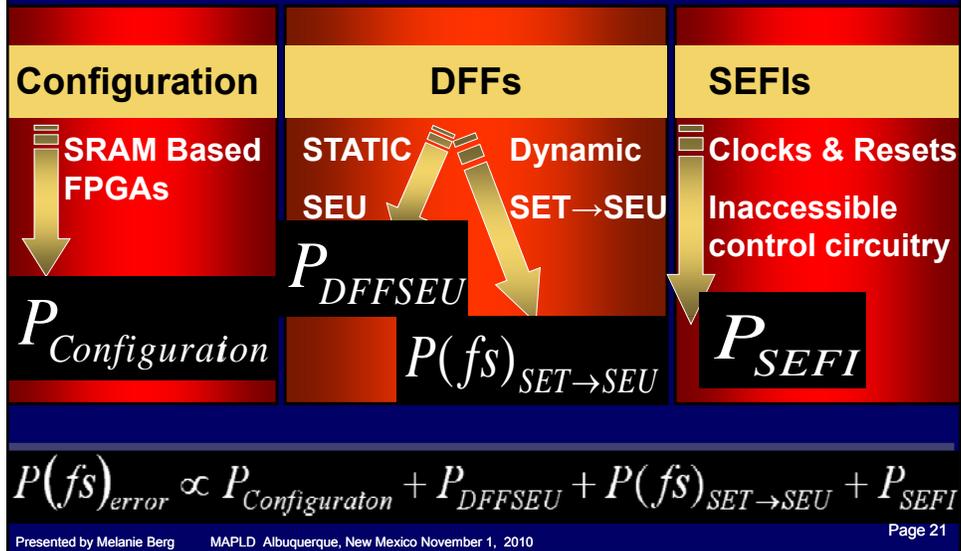


Glitch = Transient





Summary: Most Significant Factors of System Error Probability $P(fs)_{error}$



Section III: Reducing System Error: Common Mitigation Techniques



- Triple Modular Redundancy:
 - Block Triple Modular Redundancy (BTMR)
 - Local Triple Modular Redundancy (LTMR)
 - Global Triple Modular Redundancy (GTMR)
 - Distributed Triple Modular Redundancy (DTMR)

Mitigation



- Error Correction or Error avoidance
- Mitigation can be:
 - **Embedded:** built into the device library cells
 - User does not verify the mitigation – manufacturer does
 - **User inserted:** part of the actual design process
 - User must verify mitigation... Complexity is a RISK!!!!!!!
- Mitigation should reduce error...
 - Generally through redundancy
 - Incorrect implementation can increase error

Want to reduce as many terms as possible:

$$P(fs)_{error} \propto P_{Configuration} + P_{DFESEU} + P(fs)_{SET \rightarrow SEU} + P_{SEFI}$$

Example: TMR Mitigation Schemes will use Majority Voting



$$MajorityVoter = I1 \wedge I2 + I0 \wedge I2 + I0 \wedge I1$$

I0	I1	I2	Majority Voter
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

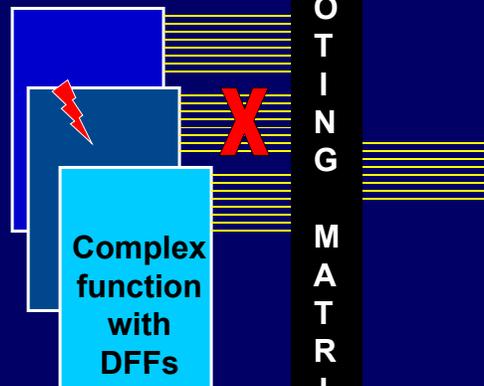
Best 2 out of 3



Reducing System Error: Common Mitigation Techniques

- **Triple Modular Redundancy:**
 - **Block Triple Modular Redundancy (BTMR)**
 - Local Triple Modular Redundancy (LTMR)
 - Global Triple Modular Redundancy (GTMR)
 - Distributed Triple Modular Redundancy (DTMR)

BTMR



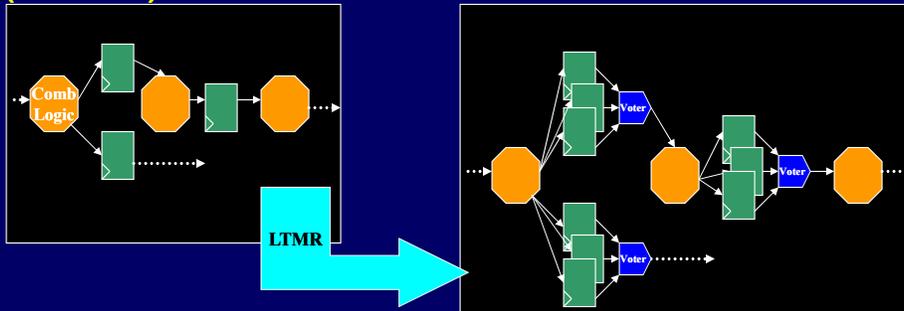
- Need Feedback to Correct
- Generally can not apply internal correction from voted outputs
- Errors can accumulate – not an effective technique



Reducing System Error: Common Mitigation Techniques

- **Triple Modular Redundancy:**
 - Block Triple Modular Redundancy (BTMR)
 - **Local Triple Modular Redundancy (LTMR)**
 - Global Triple Modular Redundancy (GTMR)
 - Distributed Triple Modular Redundancy (DTMR)

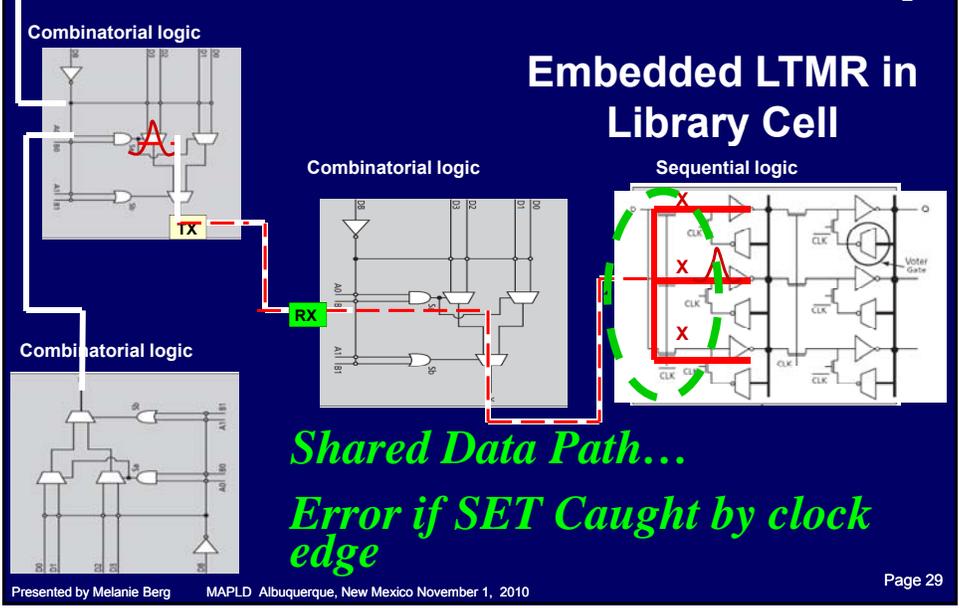
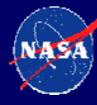
Local Triple Modular Redundancy (LTMR): Voter+Feedback=Correction



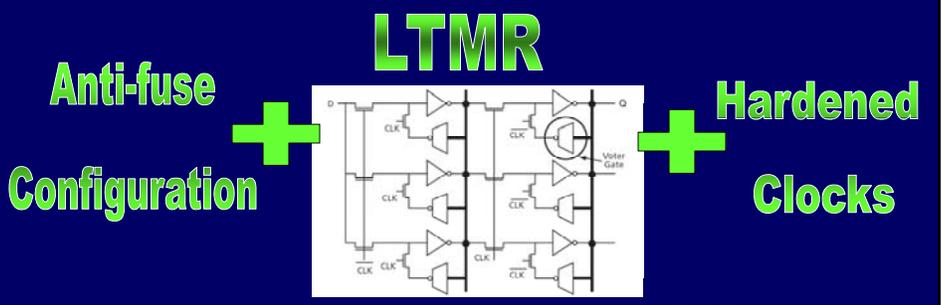
- Triple Each DFF + Vote+ Feedback Correct at DFF
- Unprotected:
 - Clocks and Resets... SEFI
 - Transients (SET->SEU)
 - Internal/hidden device logic: SEFI

$$P(fs)_{error} \propto P_{Configuration} + P_{DFFSEU} \left(P(fs)_{SET \rightarrow SEU} + P_{SEFI} \right)$$

Example... LTMR DFF Library Components and SETs



RTAX Example: Probability of Error Reduction



$$P(fs)_{error} \propto P_{Configuration}^0 + P_{D/FSEU}^{Low} + P(fs)_{SET \rightarrow SEU} + P_{SEFI}^{\sim 0}$$

- Error Rate must reflect frequency of operation
- Low Design implementation Complexity

Example...Upper-Bound Error Prediction for Actel Antifuse FPGA...LTMR + hardened Global Routes RHBD



$$P(fs)_{error} \propto P(fs)_{SET \rightarrow SEU}$$

- Given...15MHz to 120MHz: Dynamic Error Bit Rate

$$P(fs)_{SET \rightarrow SEU}$$

$$1 \times 10^{-9} < \frac{dE_{bit}(fs)}{dt} < 6 \times 10^{-8} \left[\frac{Errors}{bit - day} \right]$$

Source: NASA Goddard

Upper-Bound Error Prediction: Number of Bits x Bit Error Rate



With embedded LTMR Mitigation + Hardened Clocks:

$$P(fs)_{error} \propto P(fs)_{SET \rightarrow SEU}$$

$$\frac{dE}{dt} < \frac{dE_{bit}(fs)}{dt} * (\#UsedDFFs) < 6 \times 10^{-8} \left(\frac{Errors}{bit - day} \right) * 10^4 \left(\frac{bits}{design} \right)$$

10,000 DFFs

$$\frac{dE}{dt} < 6 \times 10^{-4} \left(\frac{Errors}{design - day} \right)$$

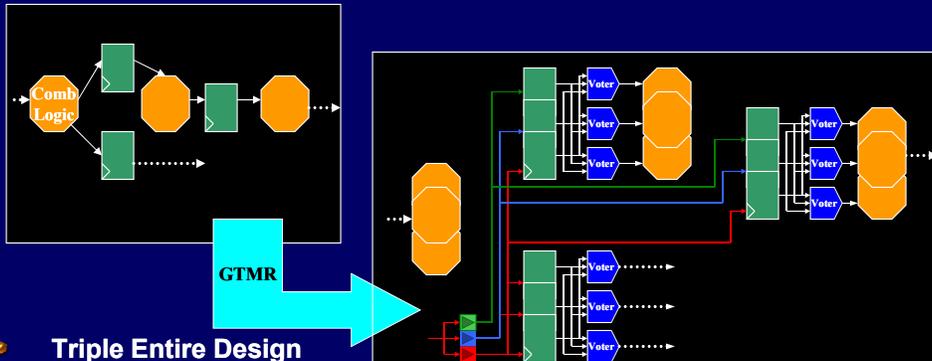
Decades



Reducing System Error: Common Mitigation Techniques

- **Triple Modular Redundancy:**
 - Block Triple Modular Redundancy (BTMR)
 - Local Triple Modular Redundancy (LTMR)
 - **Global Triple Modular Redundancy (GTMR)**
 - Distributed Triple Modular Redundancy (DTMR)

Global Triple Modular Redundancy (GTMR): Largest Area → Complexity



- Triple Entire Design
- Triple I/O and Voters
- Unprotected – hidden device logic SEFIs
- Can not be an embedded strategy: Complex to verify

$$P(fs)_{error} \propto P_{Configurat\text{ion}}^{Low} + P_{DTFSEU}^{Low} + P(fs)_{SET > SEU}^{Low} + P_{SEFI}^{Low}$$

GTMR Proves To be A Great Mitigation Strategy... BUT...



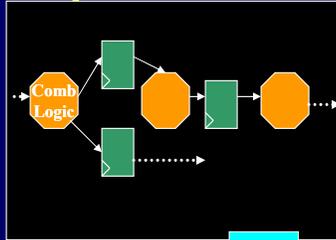
- Triplicating a design and its global routes takes up a lot of power and area
- Not part of the provided and well tested/characterized library elements
- Generally performed after synthesis by a tool— not part of RTL
- Difficult to verify
- Additional complications with Clock Skew and domain crossings
- Can be implemented in an ASIC... but is not considered as a contemporary methodology

Reducing System Error: Common Mitigation Techniques

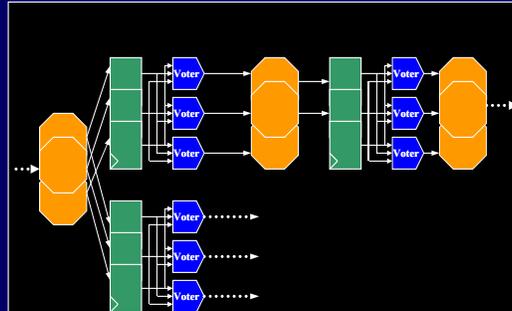


- **Triple Modular Redundancy:**
 - Block Triple Modular Redundancy (BTMR)
 - Local Triple Modular Redundancy (LTMR)
 - Global Triple Modular Redundancy (GTMR)
 - Distributed Triple Modular Redundancy (DTMR)

Distributed Modular Redundancy (DTMR)... GTMR without Clock Replication



DTMR



$$P(fs)_{error} \propto P_{Configurat\textit{on}}^{Low} + P_{DTMRSEU}^{Low} + P(fs)_{SET \rightarrow SEU}^{Low} + P_{SEU}^{Low}$$

Section IV: The Automation Process and the Mentor Graphics Advantage





Section IV: The Automation Process The Mentor Advantage

Presented by Melanie Berg MAPLD Albuquerque, New Mexico November 1, 2010

Page 39

Automation through Synthesis



- Mentor Graphics and Synplicity provide TMR insertion
- It is up to the designer to understand which type of TMR to implement based on the target FPGA and the target space environment

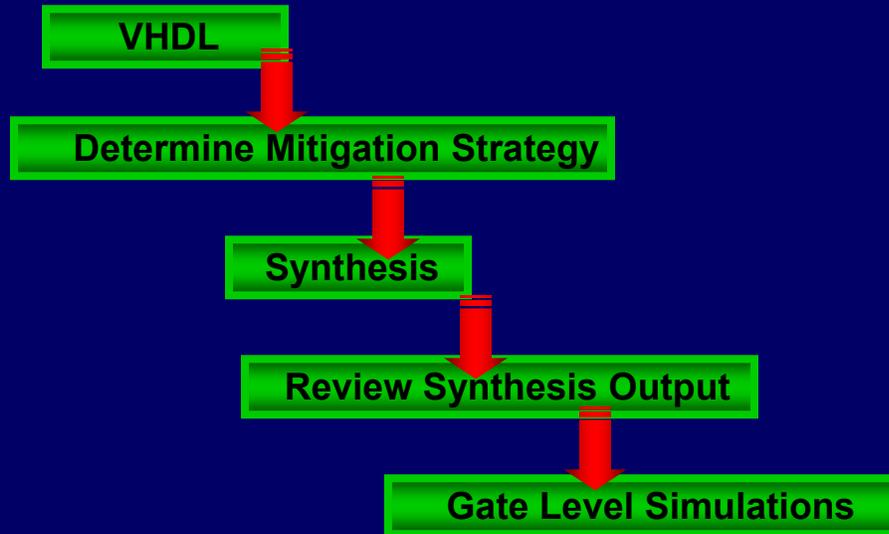
FPGA	LTMR	DTMR	GTMR
Antifuse	Green	Green	Red
Antifuse+LTMR	Orange	Orange	Red
SRAM	Red	Orange	Green
Flash	Green	Green	Red

-  General Recommendation
-  Not Recommended but may be a solution for some situations
-  Will not be a good solution

Presented by Melanie Berg MAPLD Albuquerque, New Mexico November 1, 2010

Page 40

Mitigation Design Process



Presented by Melanie Berg MAPLD Albuquerque, New Mexico November 1, 2010

Page 41

Benefits of Automation

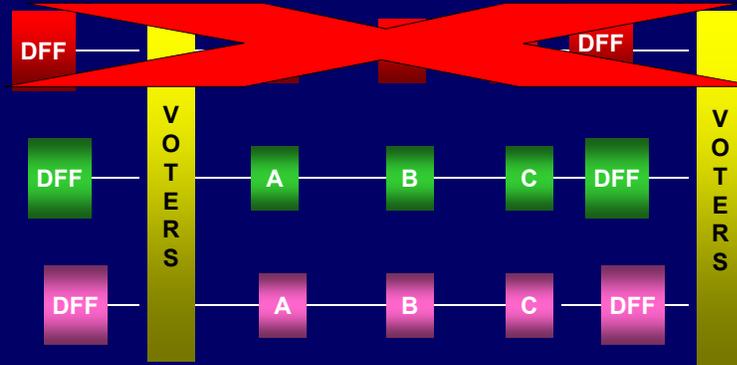


- Difficult to implement mitigation schemes manually with VHDL or Verilog
 - Synthesis Optimization
 - Designer error
 - Mitigation Optimization (voter reduction)
- May reduce the probability of insertion design error:
 - Coding errors are difficult to detect
 - Utilizes a structured and well defined insertion process

Presented by Melanie Berg MAPLD Albuquerque, New Mexico November 1, 2010

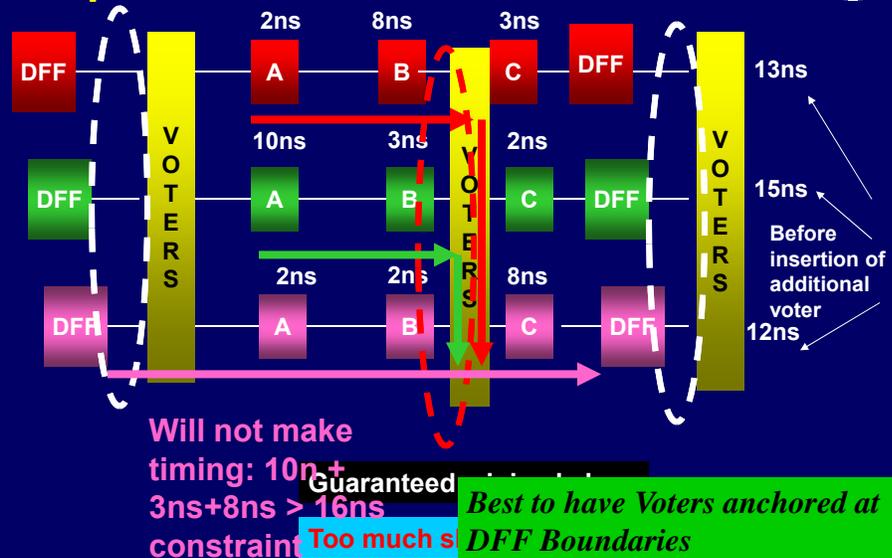
Page 42

Example: Design Error and Mitigation



- Only 2 valid paths at any given Moment due to erroneous manual design
- If an SEE error occurs in one of the functional paths, the voters will not be able to mitigate
- May not be detected during simulation

Incorrect Voter Insertion: Example with 16ns Time Constraint





Section IV: **The Automation Process** **The Mentor Advantage**

Presented by Melanie Berg MAPLD Albuquerque, New Mexico November 1, 2010

Page 45

Up-to-Date Radiation Effects Knowledge



- The best designers can create the worst designs:
 - Must understand radiation effects in order to mitigate properly
 - Each FPGA device has different error modes and signatures
- Mentor has established a close relationship with the radiation effects community
- Knowledge of current FPGA test results is the premise of Precisions mitigation strategies
- Mitigation has been utilized in NASA Goddard Radiation Effects particle accelerator experiments

Presented by Melanie Berg MAPLD Albuquerque, New Mexico November 1, 2010

Page 46

Voter Insertion



- Intelligent handling of many special cases
 - Logic Reduction
 - Primary top-level design outputs
 - Clock Enable Handling
 - Control Domain Crossings
 - Multiply-Accumulate Circuits
 - Latches
 - Combinatorial Loops
 - Black Boxes
- We don't have time to discuss all:
 - Primary top-level design outputs
 - Control Domain Crossings
 - Black Boxes

Section IV: The Automation Process The Mentor Advantage:



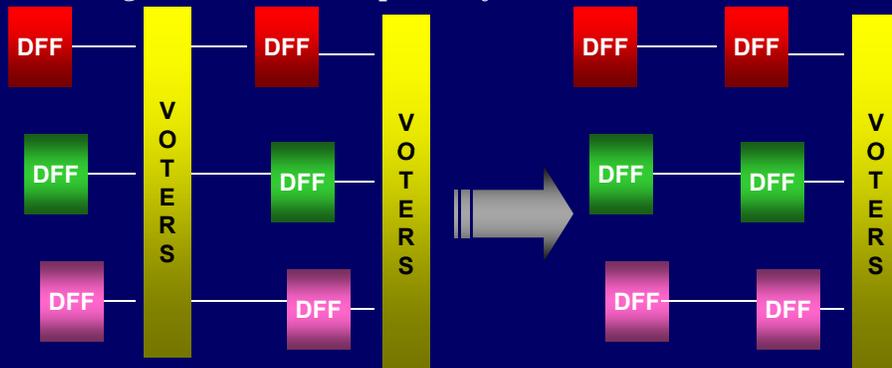
- Logic Reduction (GTMR and DTMR)
- Primary top-level design outputs
- Control Domain Crossings
- Black Boxes



Basis of Automated Process

- Voters are placed after DFFs
- **Logic Reduction:** Voters are not placed in paths of “always enabled” DFFs that are not part of a feedback loop

Logic reduction Example: No feedback – No enable



Presented by Melanie Berg MAPLD Albuquerque, New Mexico November 1, 2010



Section IV: The Automation Process The Mentor Advantage:

- Logic Reduction
- Primary top-level design outputs
- Control Domain Crossings
- Black Boxes

Presented by Melanie Berg MAPLD Albuquerque, New Mexico November 1, 2010

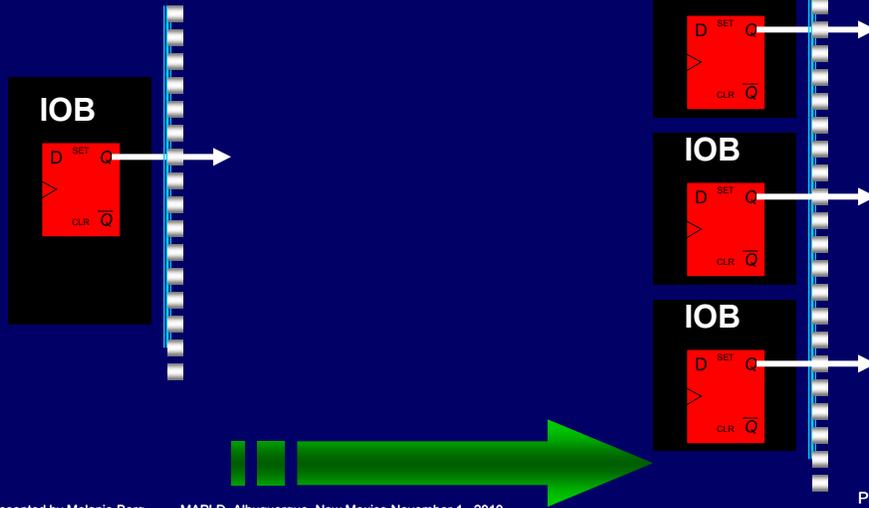
Page 50

Common DTMR and GTMR I/O Strategies



Triple I/O=TRUE, OUTFF=TRUE

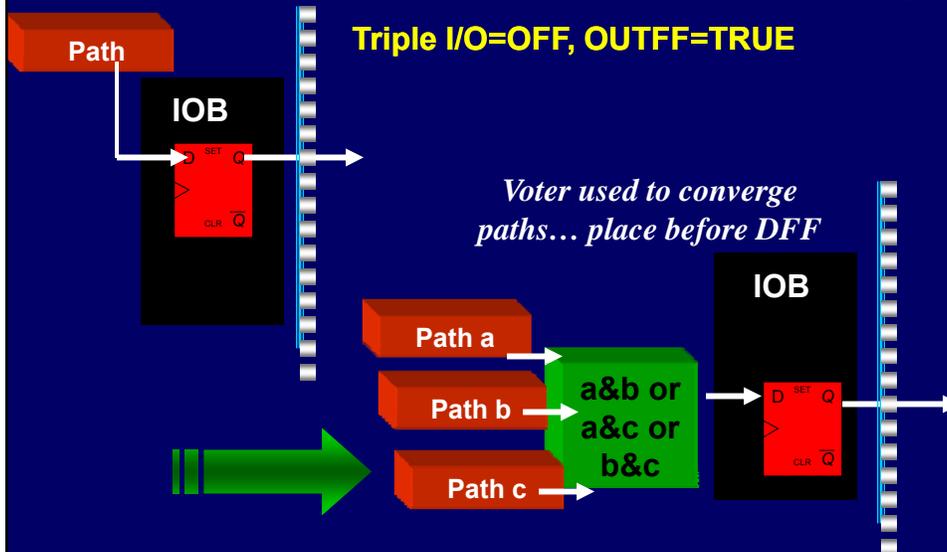
May Produce too many I/O



TMR I/O Strategies: Path Convergence



Triple I/O=OFF, OUTFF=TRUE

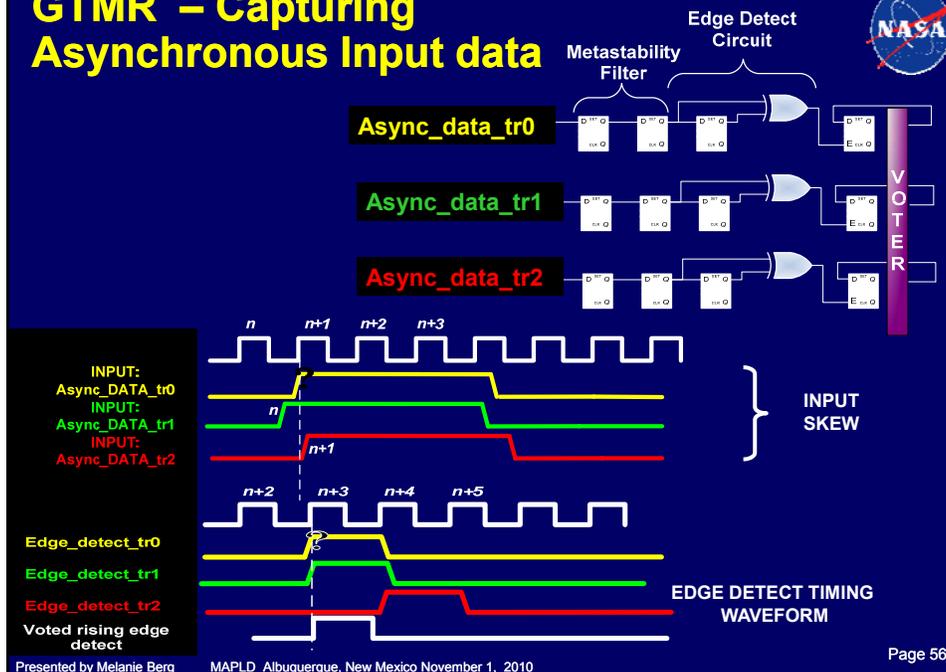




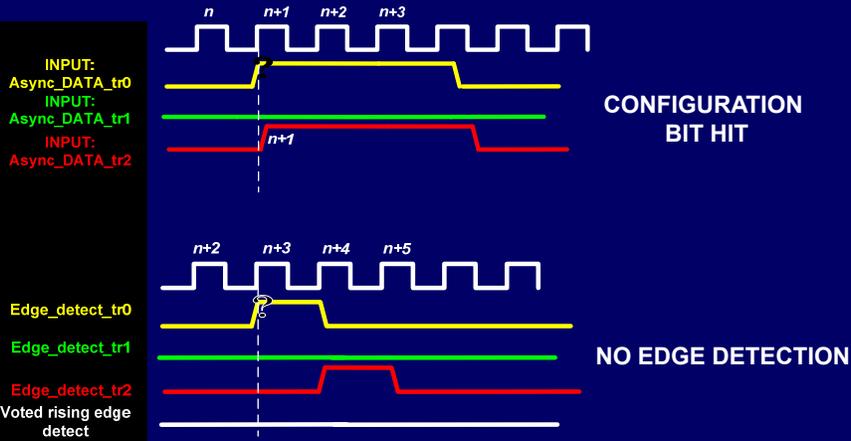
Section IV: The Automation Process The Mentor Advantage:

- Logic Reduction
- Primary top-level design outputs
- Control Domain Crossings... GTMR issue
- Black Boxes

GTMR – Capturing Asynchronous Input data



Time Domain Considerations: GTMR Single Bit Failures ...Not Detected by Static Node Analysis



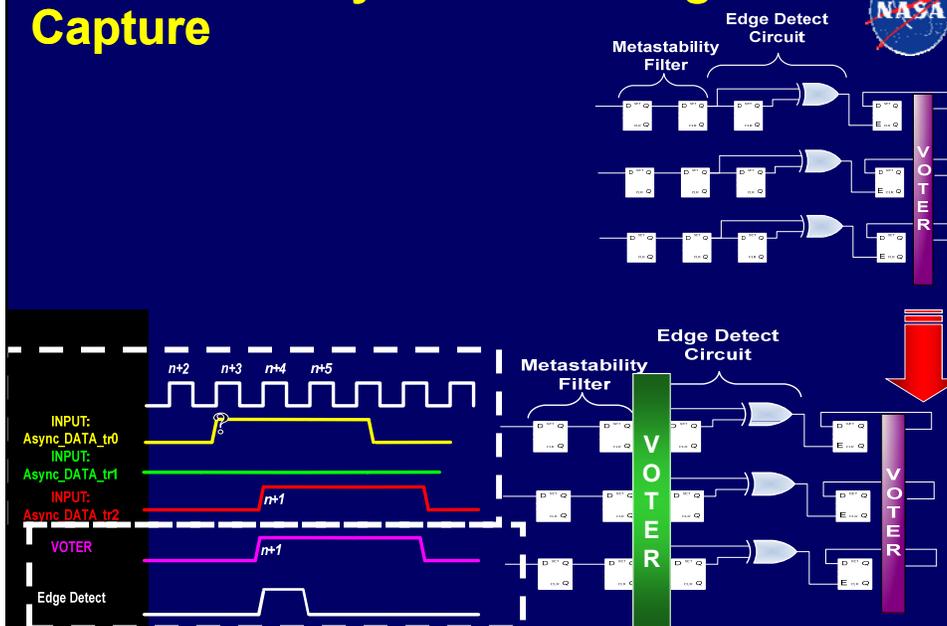
THE IMPORTANCE OF DYNAMIC ANALYSIS

Presented by Melanie Berg

MAPLD Albuquerque, New Mexico November 1, 2010

Page 57

Voters and Asynchronous Signal Capture



Presented by Melanie Berg

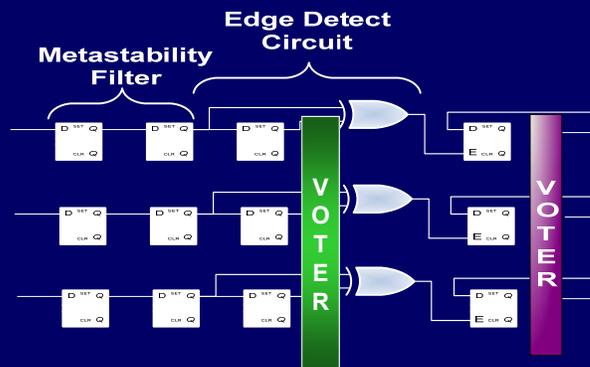
MAPLD Albuquerque, New Mexico November 1, 2010

Page 58

Clock Domain Crossings and Automation



User will want to put an attribute to ensure voter in this asynchronous path



Section IV: The Automation Process The Mentor Advantage:

- Logic Reduction
- Primary top-level design outputs
- Control Domain Crossings... GTMR issue
- Black Boxes



DTMR Black Box Handling



```

module Top(a, b, c, clk, o);
input a, b, c;
input clk;
output o;

wire tmp1;
reg tmp2;

black_box U1_bb(tmp2, clk, o);

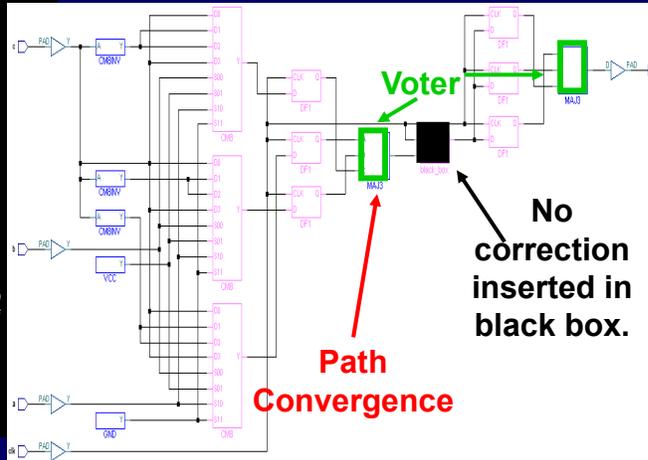
always @(posedge clk)
begin
tmp2 = tmp1;
end

assign tmp1 = a + b + c;

endmodule

module black_box(in1, clk, out1)
/* synthesis black_box */
input in1;
input clk;
output out1;
reg out1;

endmodule
    
```



DTMR result. Voters are used to converge tripled logic at black box inputs. Black box outputs fan out to tripled logic.

Summary



- SEEs will affect FPGAs in space radiation environments
- TMR has been the most effective SEE mitigation technique
- There are many types of TMR:
 - BTMR
 - LTMR
 - DTMR
 - GTMR
- The goal is to select the optimal TMR scheme regarding:
 - SEE requirements
 - Area, Power, Speed

Summary (Continued)



- Mentor has integrated different TMR schemes into their synthesis package.
- The designer must be aware of the target FPGA and its SEE sensitivity before using any automated approach
- Strategies are robust:
 - Flexible based on FPGA susceptibility
 - Many user options
 - Validated via radiation testing
- After TMR insertion, a rigorous review and simulation process must be performed