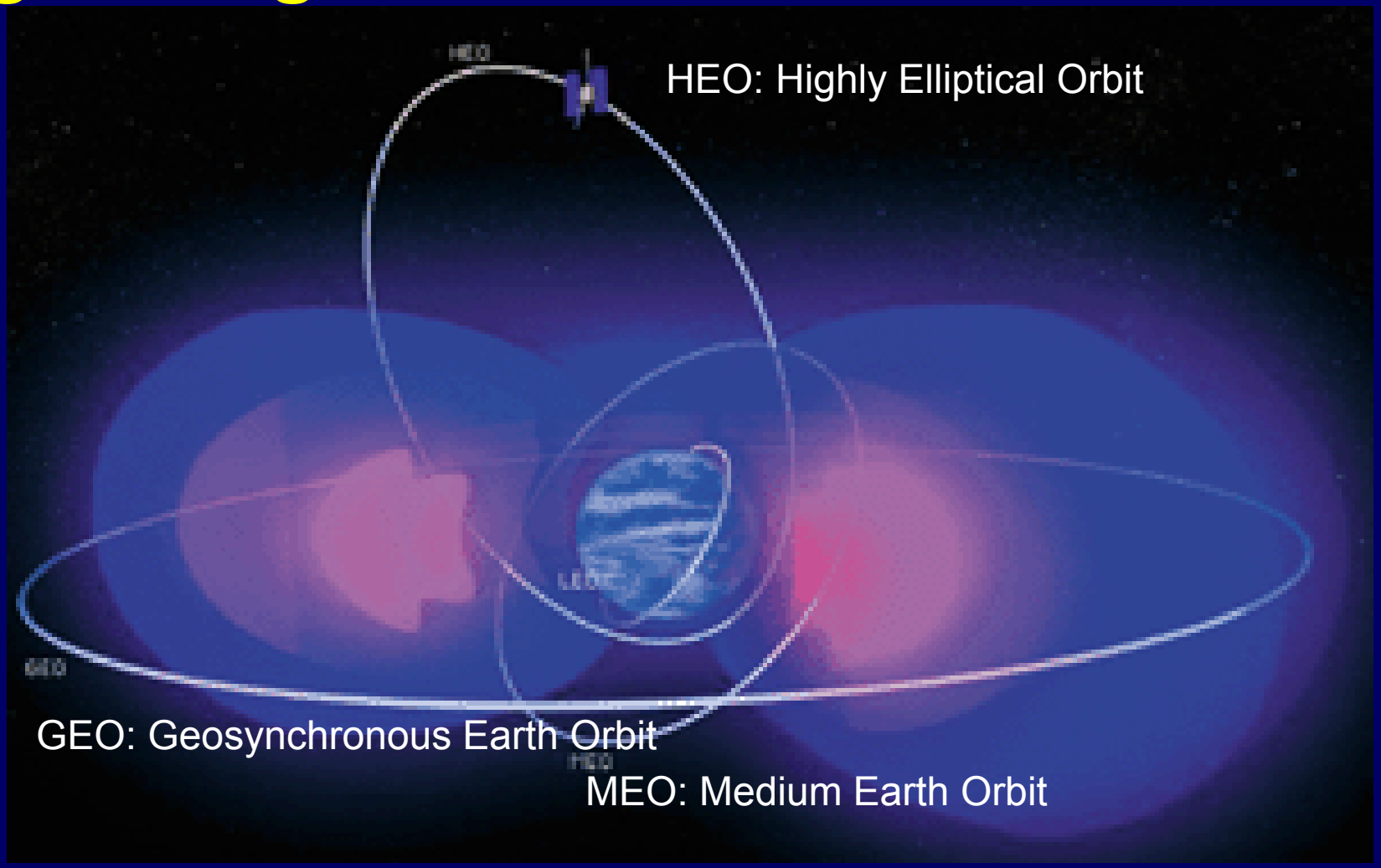
Complexity Management and Design Optimization Regarding a Variety of Triple Modular Redundancy Schemes through Automation

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

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

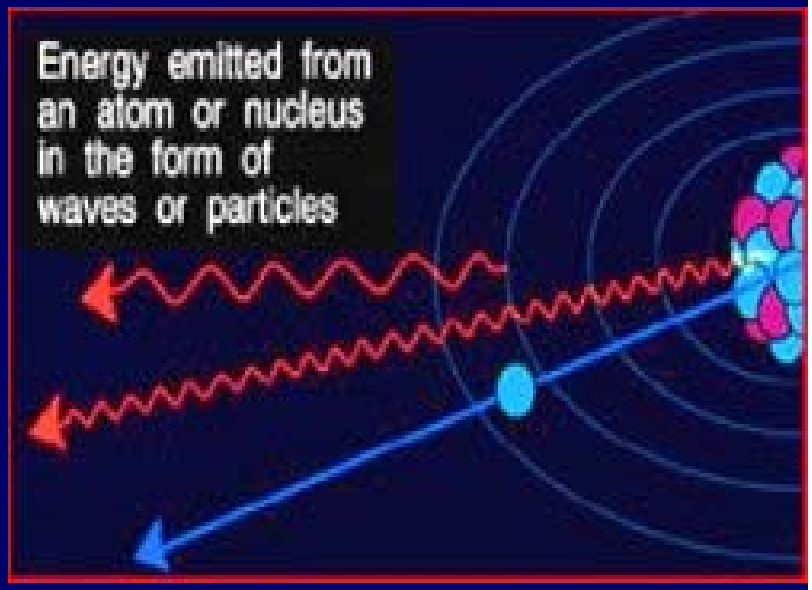
Section I: Single Event Effects in Digital Logic



Van Allen Radiation Belts:

Illustrated by Aerospace Corp.

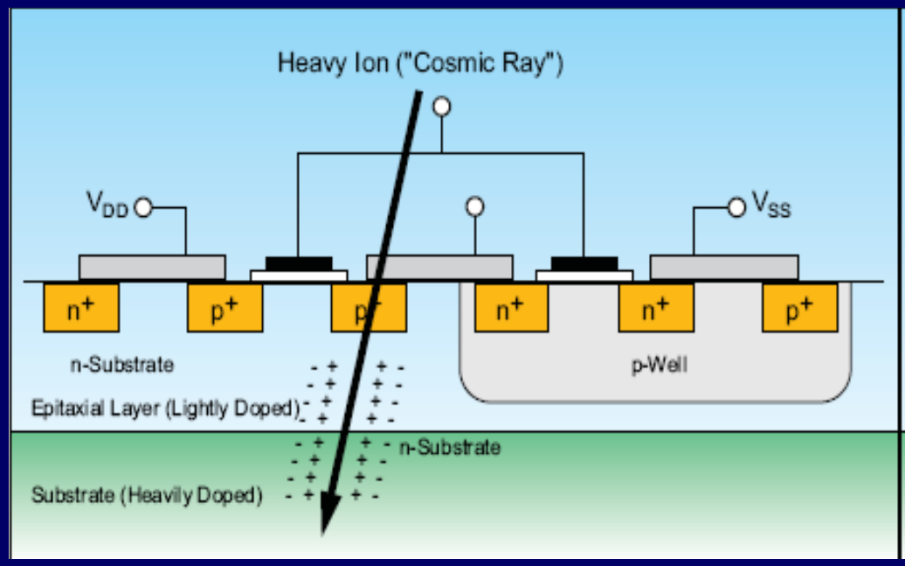
Source of Faults: SEEs and Ionizing Particles



- 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

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



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

Average energy deposited per unit path length

Units

Density of target material

LET vs. Error Cross Section Graph

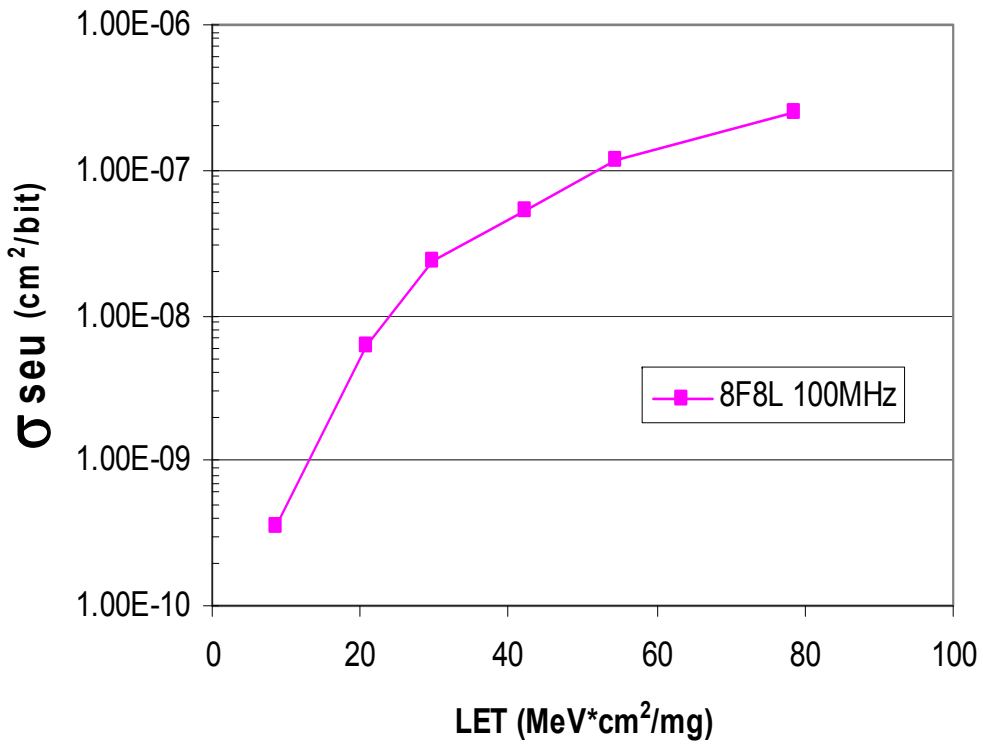
Error Cross Sections are calculated per LET value in order to characterize the number of potential faults and error rates in the space environment

Terminology:

- Flux: Particles/(sec-cm^2)
- Fluence: Particles/cm^2
- Error cross section(σ): #errors normalized by fluence
- Error cross section is calculated at several LET values (particle spectrum)

LET vs. σ:

$$\sigma_{seu} = \frac{\#errors}{fluence}$$

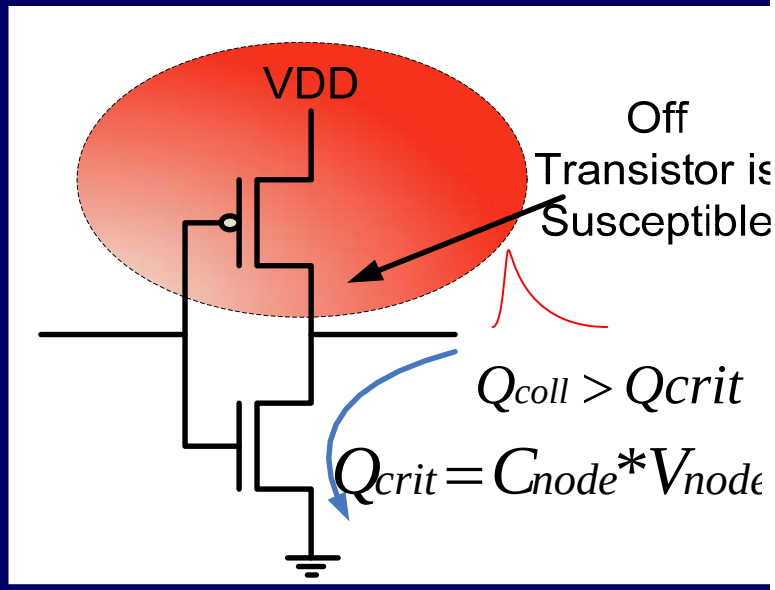


Single Event Faults and Common Terminology

- 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

Radiation Induced Fault Generation

- (SET)s can develop in combinatorial logic
- SETs can vary in pulse width (T_{pulse}) and amplitude.

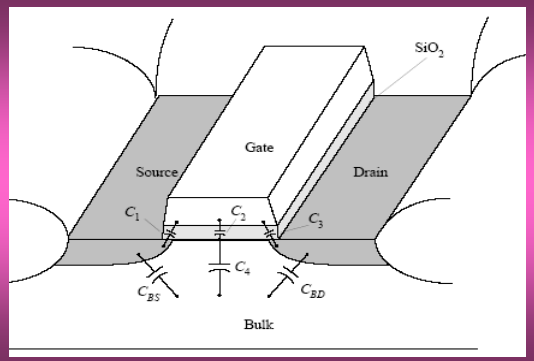


CAPACITANCE

Transistor Cutoff frequencies

$$f_c = 1/2\pi RC$$

Each capacitance has its own f_c

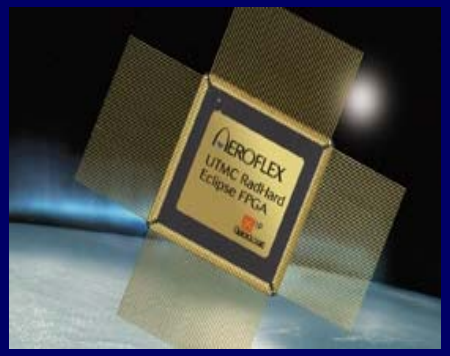


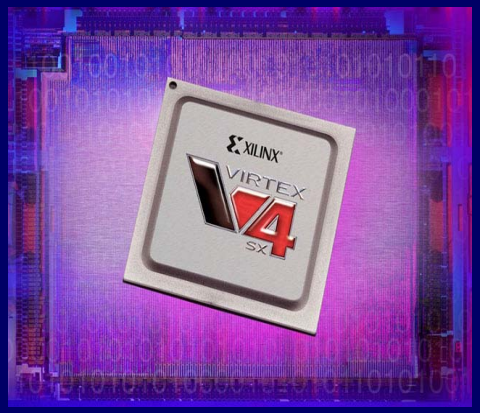
- Geometry of Transistors
- Loading of Transistors
- Length of Routes
- Switching Rates

Single Event Effects (SEEs) and IC System Error

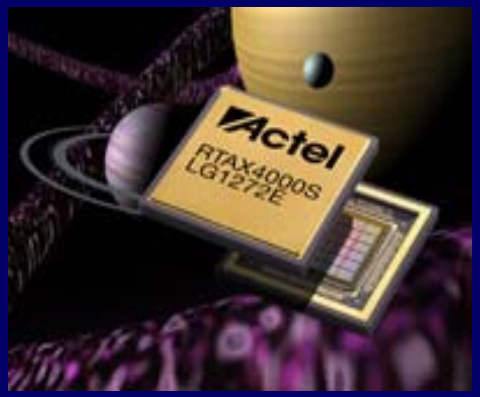
- **SEUs or SETs can occur in:**
  - Combinatorial Logic (including global routes)
  - Sequential Logic
  - Memory Cells
- Depending on the Device and the design, each fault type will:
  - Have a probability of occurrence
  - Either have a significant or insignificant contribution to system error

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





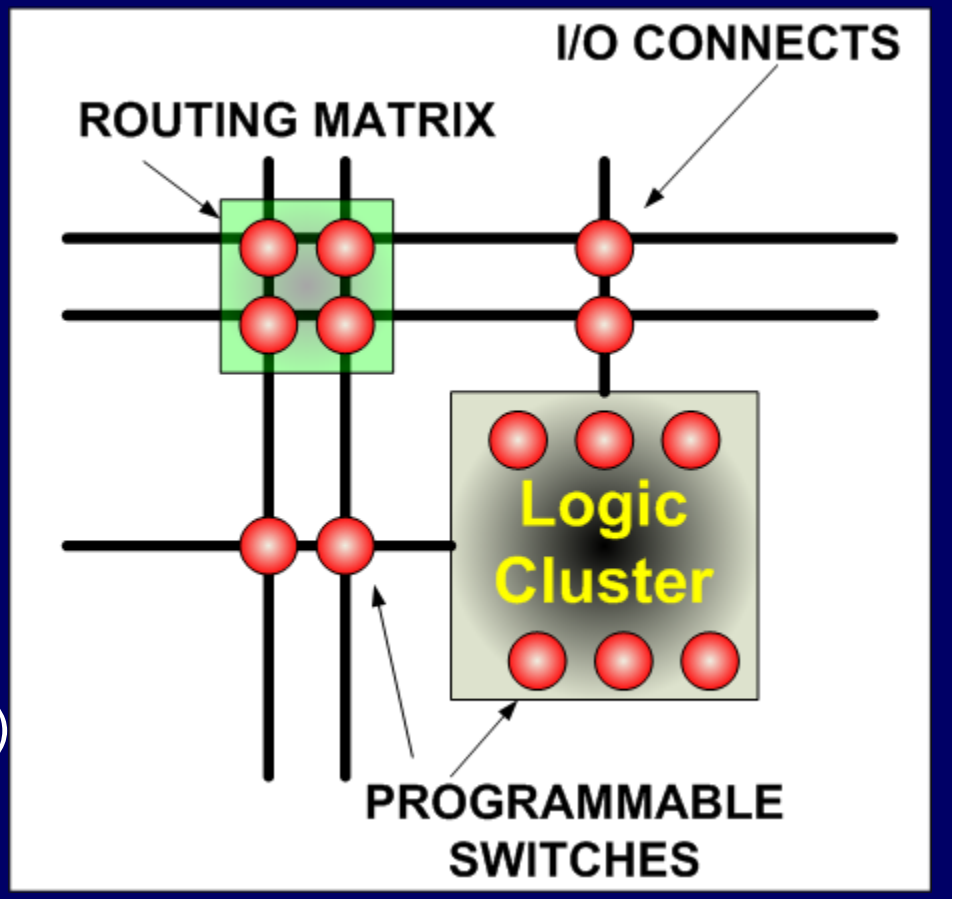
Section II: FPGA Basics – Architectural Differences



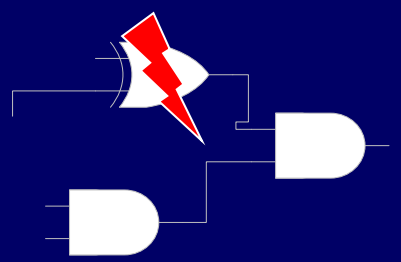
Configuration… Only FPGAs

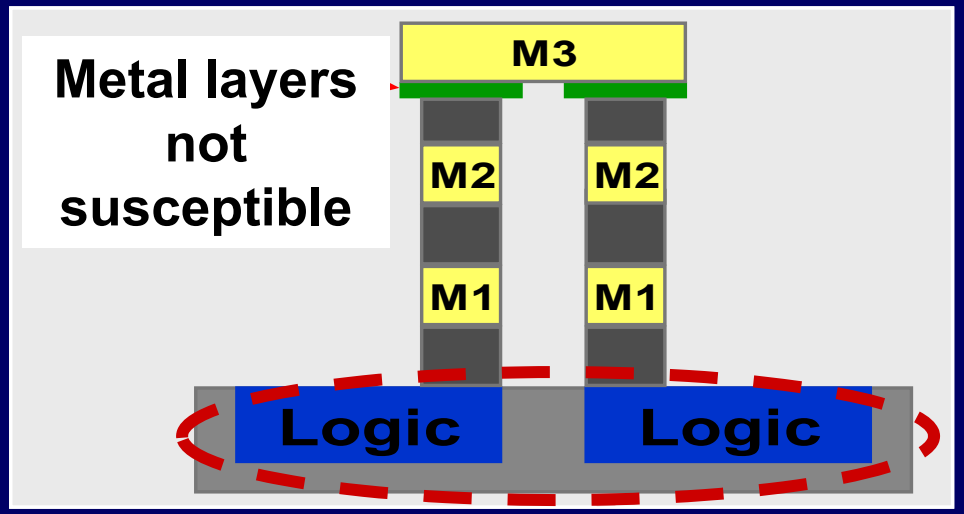
HDL → **FPGA MAPPING** → **Configuration**

- **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)



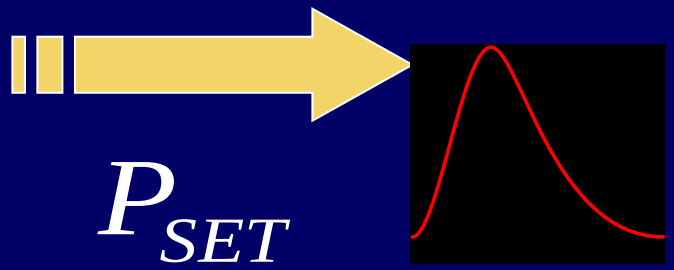
Combinatorial Logic Blocks and Potential Upsets… SETs in ASICs and Anti-fuse FPGAs





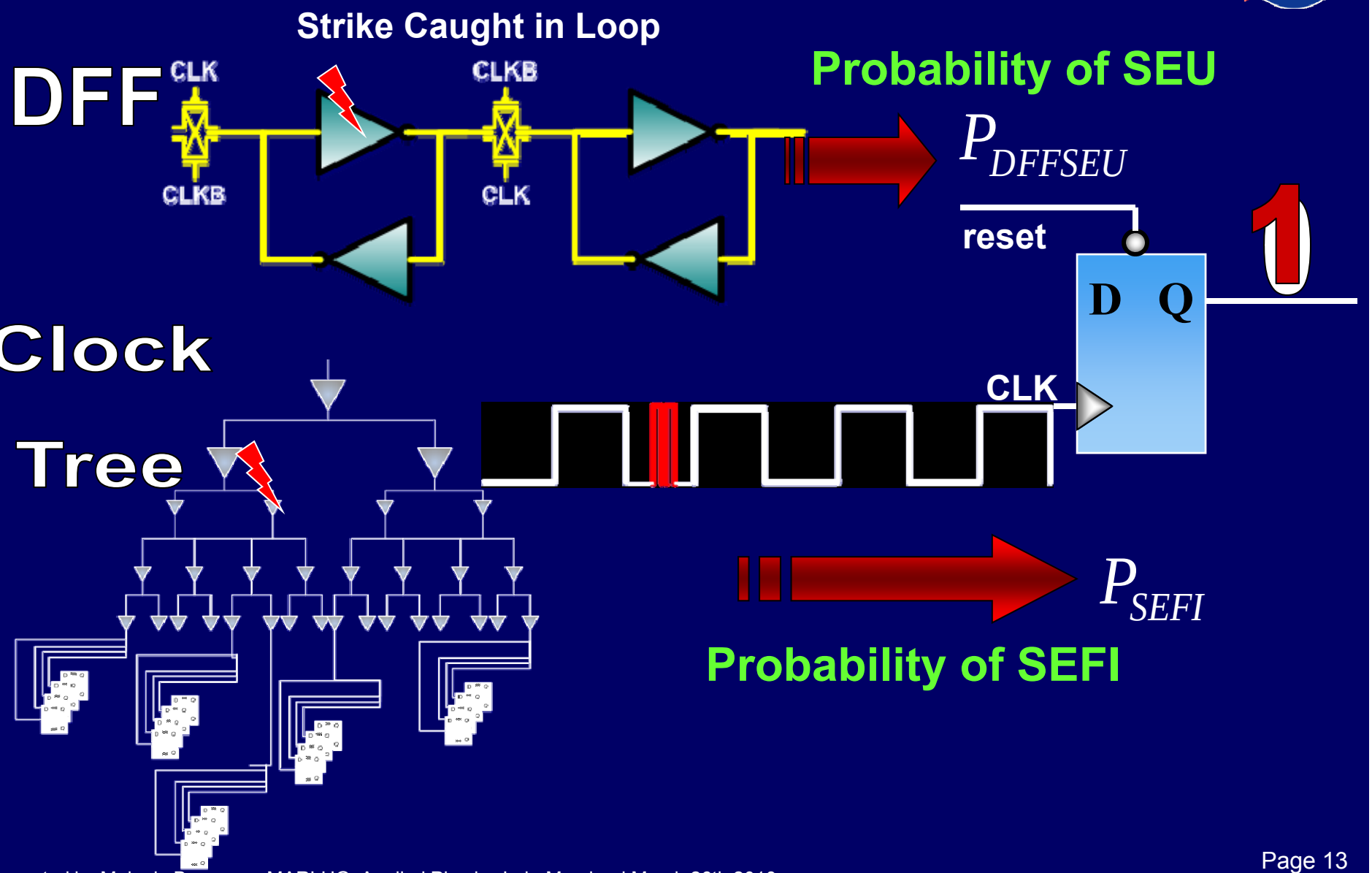
Sensitive Region

Glitch = Transient

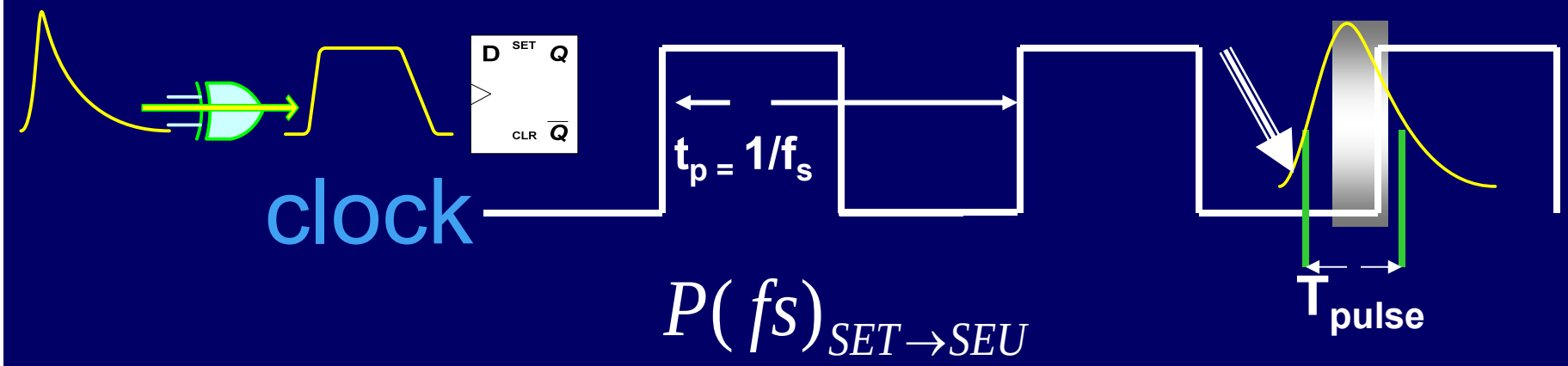


P_{SET}

DFF's: SEUs and SEFIs



Transient Capture on A DFF Data Input Pin (SET→SEU)



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

fs	: System Frequency
$T(fs)_{pulse}$	: SET Pulse Width
$P(fs)_{SETgen}$	: Probability SET generated with sufficient amplitude
$P(fs)_{SETprop}$	: Probability SET can propagate with sufficient amplitude
P_{DFFEn}	: Probability DFF is enabled (active)
$P(fs)_{SET \rightarrow SEU}$	: Probability SET can be caught by clock edge

$$P(fs)_{set->seu} \propto \frac{T(fs)_{pulse} \times P(fs)_{SETgen} \times P(fs)_{SETprop} \times P_{DFFEn}}{2 \times \frac{1}{fs}}$$

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

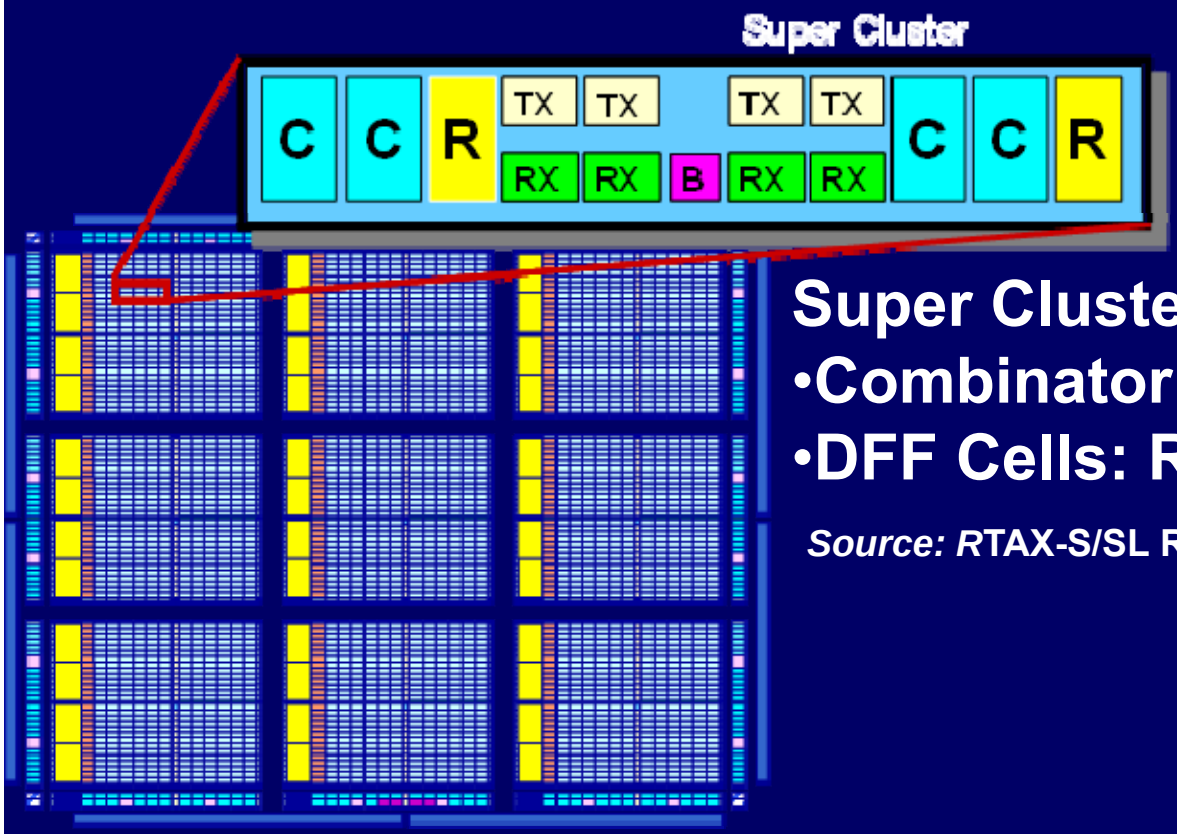
Configuration	DFFs	SEFIs
SRAM Based FPGAs	STATIC SEU	Dynamic SET→SEU	Clocks & Resets Inaccessible control circuitry
$P_{Configuration}$	P_{DFFSEU}	$P(fs)_{SET \rightarrow SEU}$	P_{SEFI}

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

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

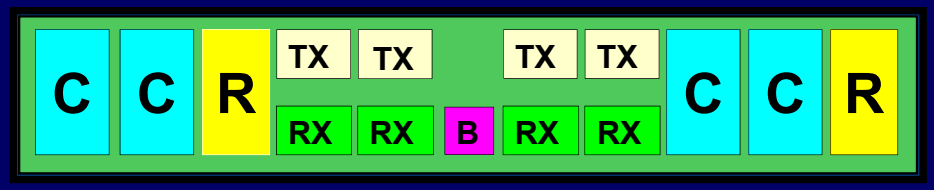


Super Cluster:

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

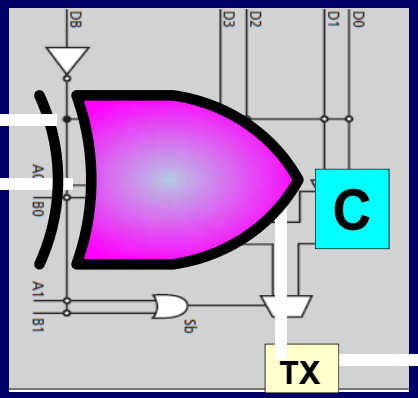
Source:* RTAX-S/SL RadTolerant FPGAs *2009 Actel.com

ACTEL RTAX-S Combinatorial and Sequential Logic

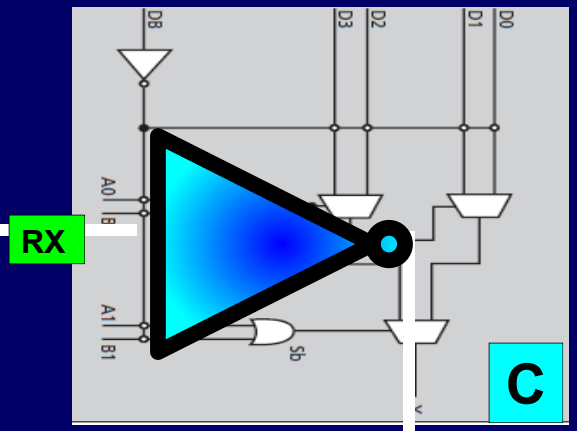


Super Cluster

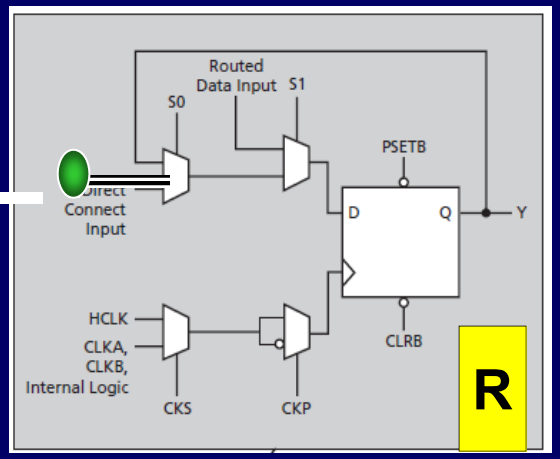
Combinatorial logic: C-CELL



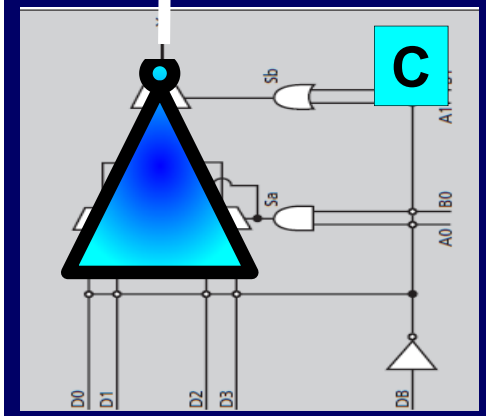
Combinatorial logic C-CELL



Sequential logic R-CELL

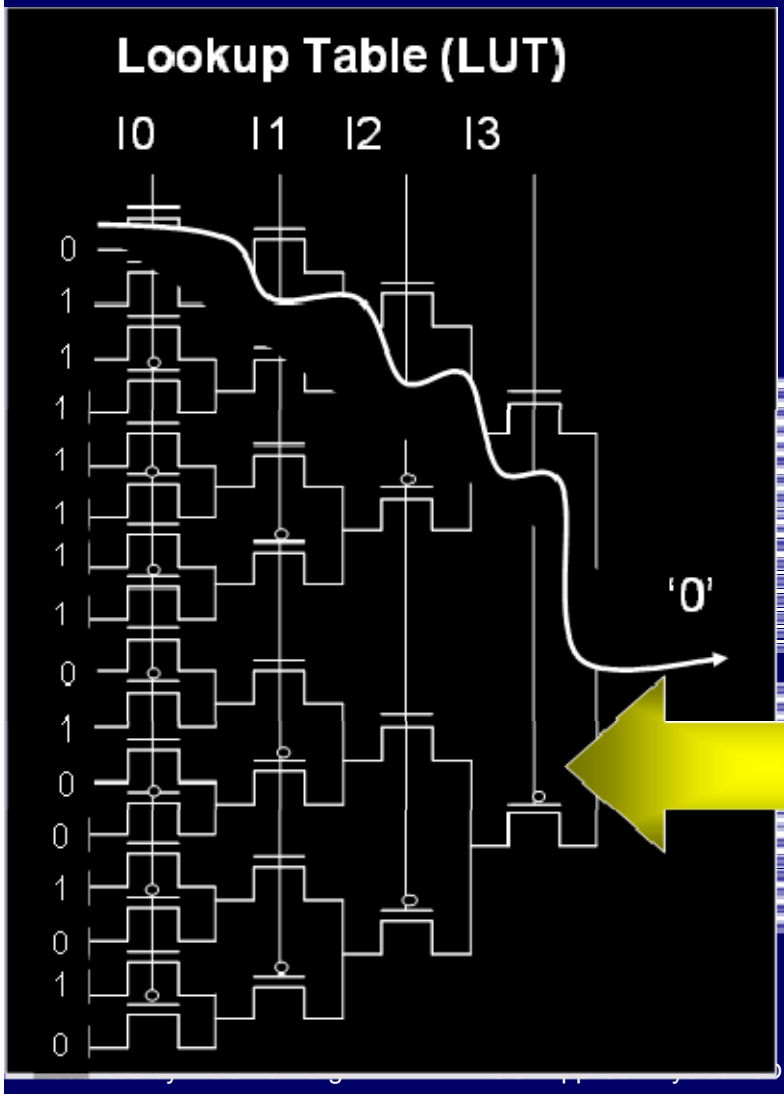


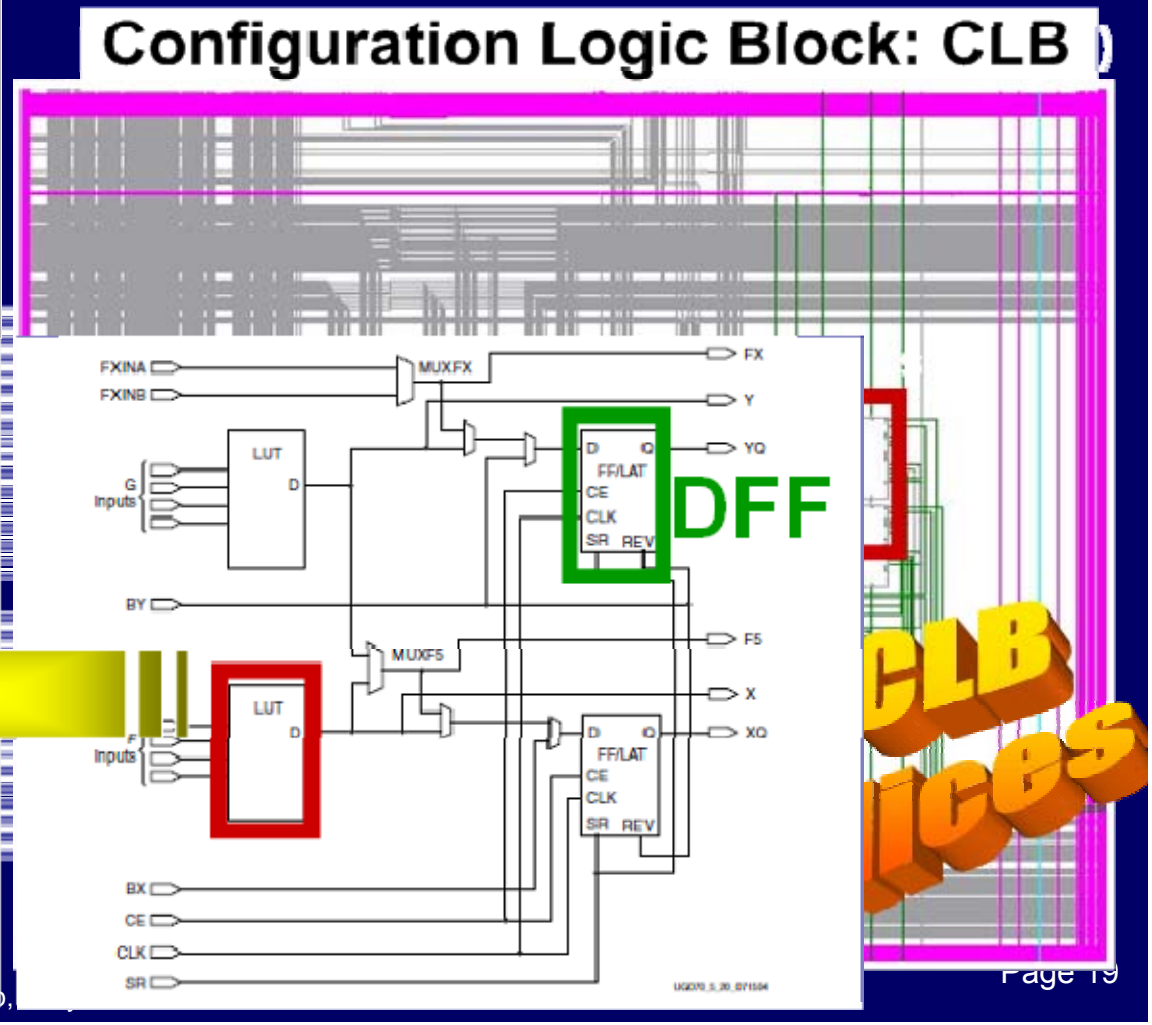
Combinatorial logic C-CELL



General Xilinx Virtex 4 FPGA Architecture: SRAM Based Configuration

Functional Logic





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:

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

Example: TMR Mitigation Schemes will use Majority Voting

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

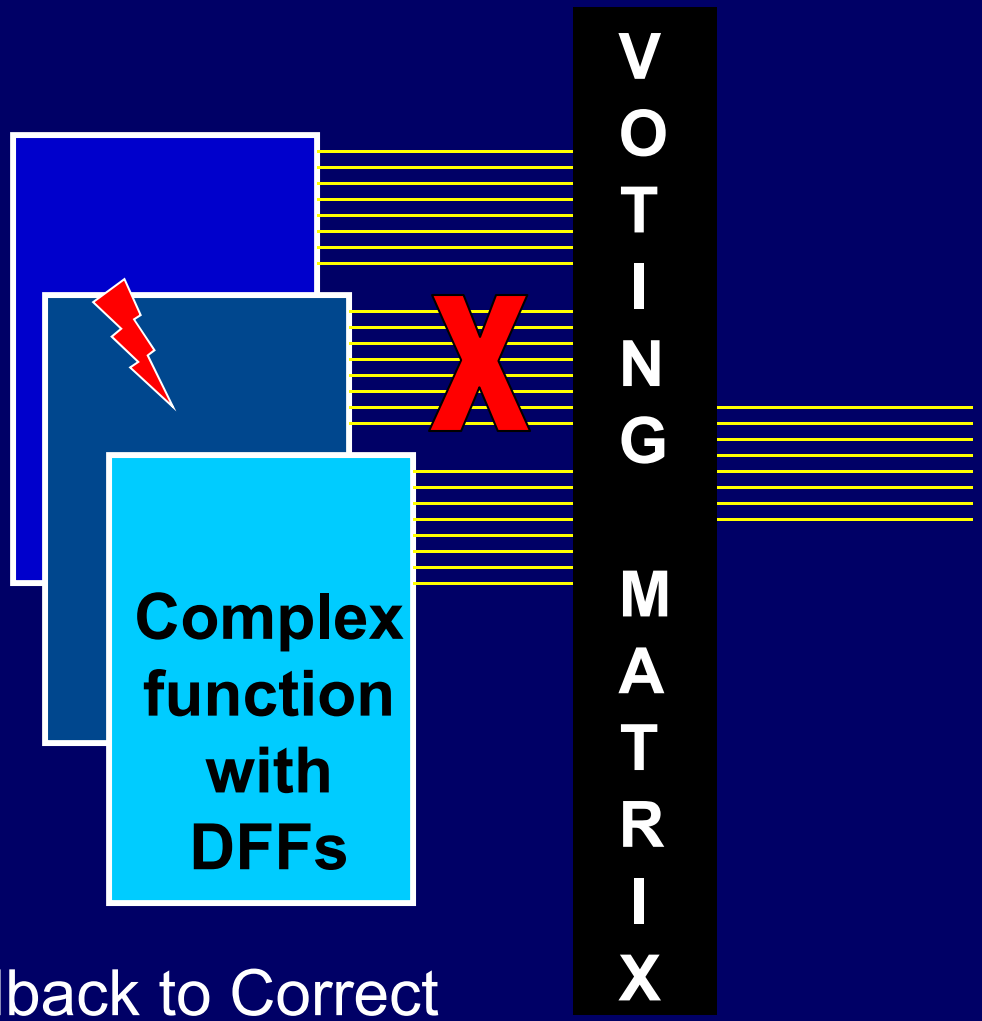
I0	I1	I2	Majority Voter
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

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

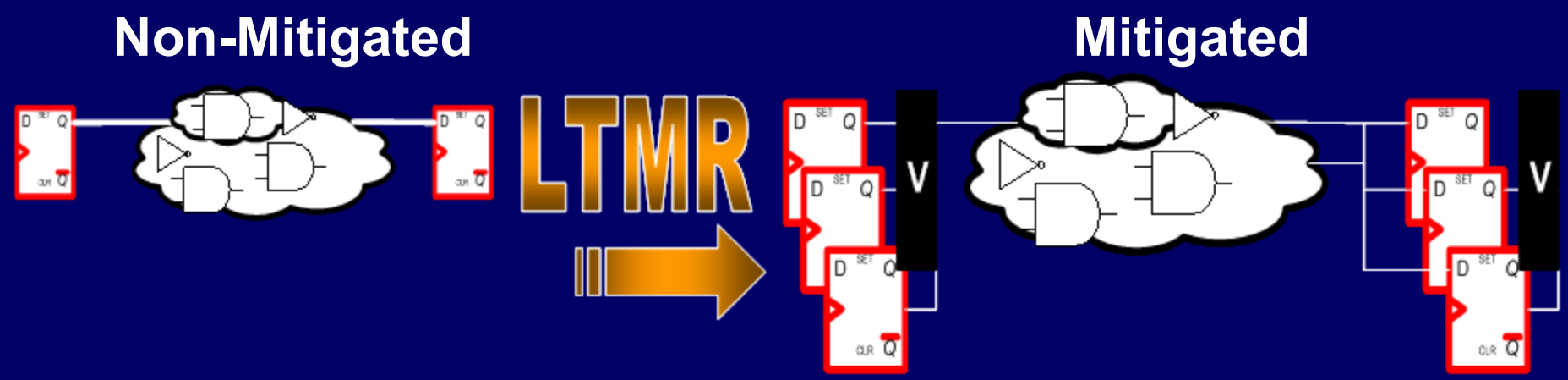


- Need Feedback to Correct
- Generally can not apply internal correction from voted outputs
- Errors can accumulate – not an effective technique

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

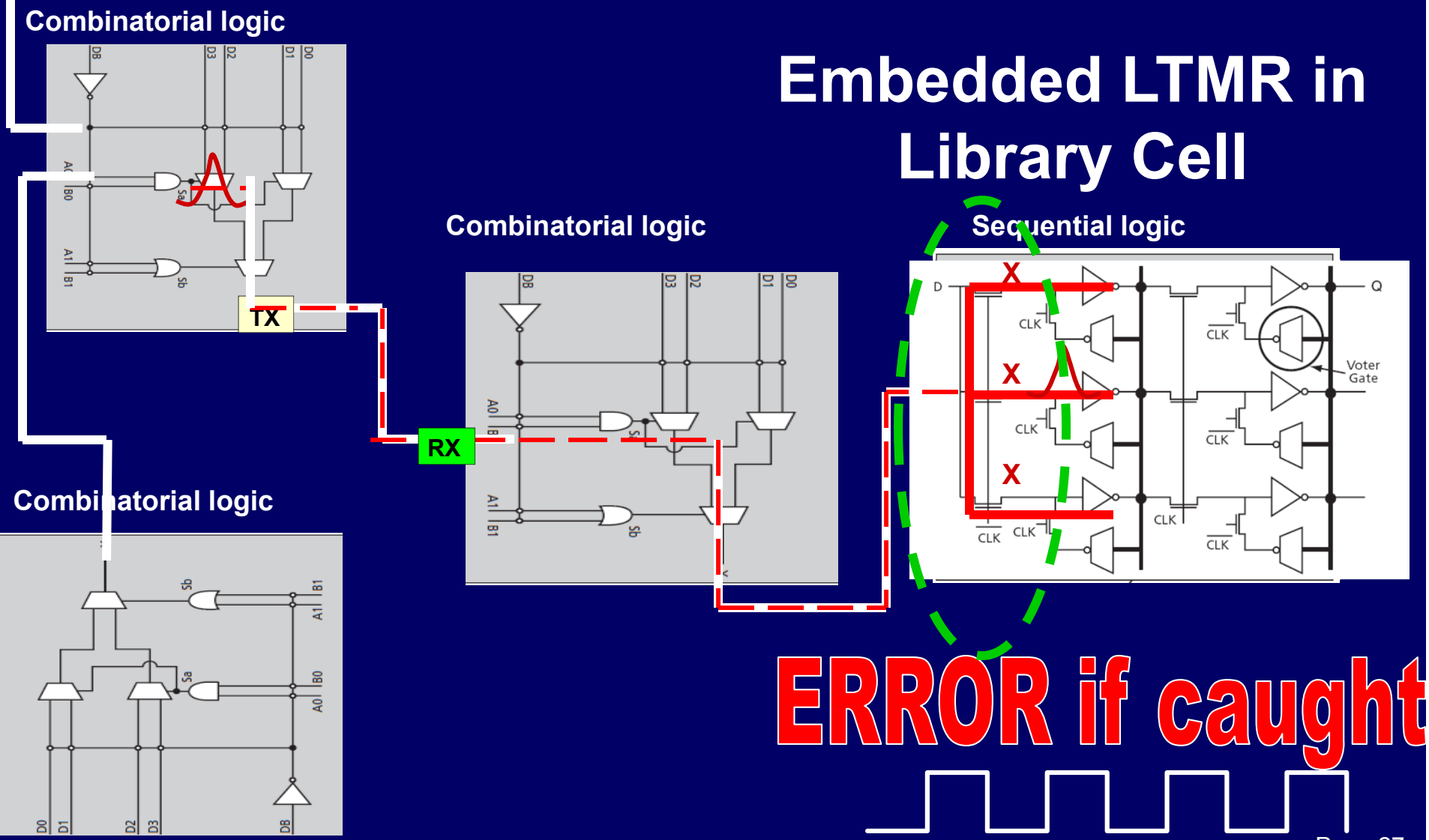


- Triple Each DFF + Vote+ Feedback Correct at DFF
- Unprotected:
  - Clocks and Resets… SEFI
  - Transients (SET->SEU)
  - Internal/hidden device logic: SEFI

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

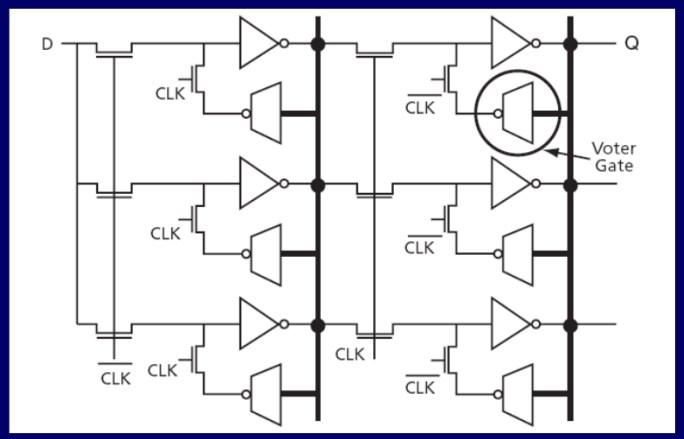
Low

Example… LTMR DFF Library Components and SETs



RTAX Example: Probability of Error Reduction

LTMR



+

Hardened Clocks

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

0 Low ~0

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

Example…Upper-Bound Error Prediction for LTMR + hardened Global Routes RHBD

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

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

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

Source: NASA Goddard

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

Upper-Bound Error Prediction Actel RHBD Anti-fuse FPGA

With embedded LTMR Mitigation + Hardened Clocks:

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

50,000 DFFs

$$\frac{dE}{dt} < \frac{dE_{bit}(fs)}{dt} * (\#UsedDFFs) < 6x10^{-8}\left(\frac{Errors}{bit-day}\right) * 5x10^{4}\left(\frac{bits}{design}\right)$$

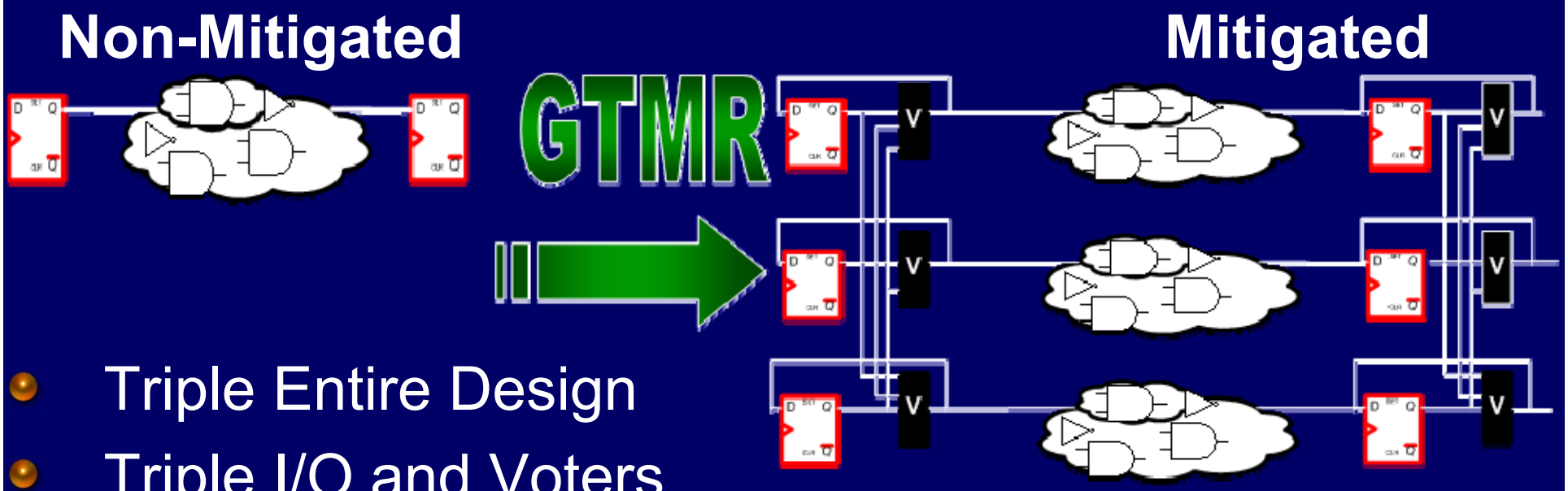
$$\frac{dE}{dt} < 3x10^{-3}\left(\frac{Errors}{design-day}\right)$$

Years

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

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

Low Low Low 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)**

DTMR

- Looks a lot like GTMR only difference is that the Global routes and I/O are not triplicated
- Small reduction in area vs. GTMR
- Small reduction in power vs. GTMR
- Can be slightly slower than GTMR because all circuitry share the same clock

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

Low Low Low

Section IV: The Automation Process

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			
SRAM			
Flash			

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

Automation Process

VHDL

Select Mitigation

Synthesis

Review Synthesis Output

Gate Level Simulations

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
- Vendors have 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
- After TMR insertion, a rigorous review and simulation process must be performed