NEPP Electronic Technology Workshop
June 28-30, 2011

National Aeronautics and Space Administration

Taming the SEU Beast - Approaches and Results for FPGA Devices and How To Apply Them

Melanie Berg, MEI Technologies/NASA GSFC
M. Friendlich, C. Perez, H. Kim: MEI Technologies/NASA GSFC
K. LaBel: NASA GSFC

To be presented by Melanie Berg at the NASA Electronic Parts and Packaging (NEPP) Program Electronic Technology Workshop, Greenbelt, Maryland, June 28-30, 2011, and published on nepp.nasa.gov.

Introduction

- FPGA and ASIC SEE Models have been developed
  - Compartmentalize SEEs to enhance analysis
  - Uses a top-down approach
- Currently, SEEs are falling nicely into their designated categories
  - Evaluate effectiveness of mitigation strategies
  - Determine dominant SEE components
  - Eases the overall analysis process
- Details of SEE generation and other electrical properties are part of ongoing development
- Presented models are only expected to fit synchronous designs as per NASA design guidelines.

To be presented by Melanie Berg at the NASA Electronic Parts and Packaging (NEPP) Program Electronic Technology Workshop, Greenbelt, Maryland, June 28-30, 2011, and published on nepp.nasa.gov.

To be presented by Melanie Berg at the NASA Electronic Parts and Packaging (NEPP) Program Electronic Technology Workshop, Greenbelt, Maryland, June 28-30, 2011, and published on nepp.nasa.gov.

NASA REAG FPGA Top-Level Susceptibility Model

Model has 3 major categories:

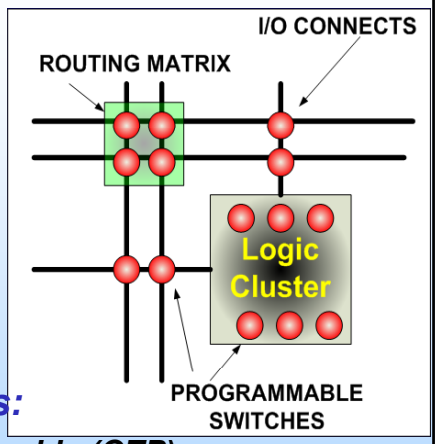
$$P(fs)_{error} \propto P_{Configuration} + P_{functionalLogic}(fs) + P_{SEFI}$$

- $P(fs)_{error}$: *Probability for Design Specific system SEE*
- $P_{Configuration}$: *Probability for Configuration SEE*
- $P_{functionalLogic}(fs)$: *Probability for Functional logic SEE*
- P_{SEFI}: *Probability for Single Event functional Interrupt*

Configuration

$$P_{Configuration}$$

Place, Route, and Gate Utilization are Stored in the FPGA Configuration

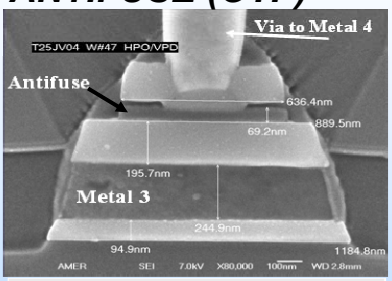


- ***Configuration Defines:***
  Arrangement of pre-existing logic via programmable switches
  - ***Functionality (logic cluster)***
  - ***Connectivity (routes)***
  - ***Placement***
- ***Programming Switch Types:***
  - ***Antifuse:*** ***One time Programmable (OTP)***
  - ***SRAM:*** ***Reprogrammable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

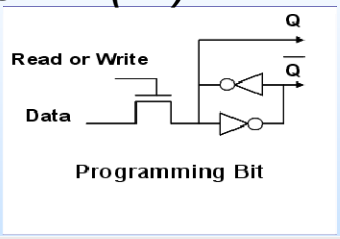
To be presented by Melanie Berg at the NASA Electronic Parts and Packaging (NEPP) Program Electronic Technology Workshop, Greenbelt, Maryland, June 28-30, 2011, and published on nepp.nasa.gov.

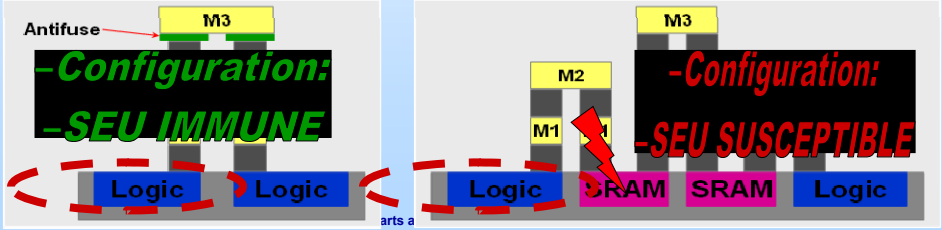
Programmable Switch Implementation and Single Event Upset (SEU) Susceptibility

ANTIFUSE (OTP)



SRAM (RP)





Greenbelt, Maryland, June 28-30, 2011, and published on nepp.nasa.gov.

Configuration Test and Analysis

- **Configuration is static during operation… hence we test and evaluate it statically**

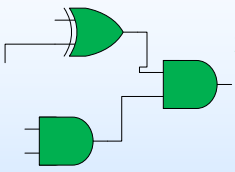
	Antifuse	SRAM	FLASH	Hardened SRAM
Manufacturer	Microsemi Aeroflex	Xilinx Achronix	Microsemi	Xilinx
Upset Signature	Fuse Resistivity	Bit State	Bit State or resistivity	Bit State
Configuration Test	Non-Specific	Readback post-irradiation	Verify Post-irradiation	Readback post-irradiation
Information from Configuration Test	N/A	Upset configuration bits	Pass/Fail	Upset configuration bits
Results	Insignificant	Dominant upsets	Insignificant	Low significance
REAG Tested	Yes	Yes	Yes	No

Impact of Configuration Testing and Analysis to the REAG Model

	REAG Model
Antifuse	$P(fs)_{error} \propto P_{functionalLogic}(fs) + P_{SEFI}$
SRAM **(non-mitigated)**	$P(fs)_{error} \propto P_{Configuration}$
Flash	$P(fs)_{error} \propto P_{functionalLogic}(fs) + P_{SEFI}$
Hardened SRAM	$P(fs)_{error} \propto P_{Configuration} + P_{functionalLogic}(fs) + P_{SEFI}$

Data Path Functional Logic
Concepts of Synchronous Design

Synchronous Design Basic Building Blocks: Combinatorial Logic and DFF's

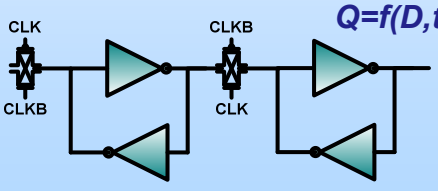


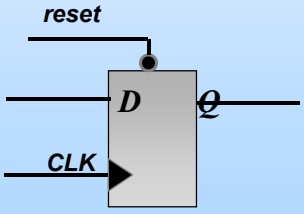
Combinatorial Logic: Output is a function of the inputs after some delay

Output=$f(input,\tau_{dly})$

DFF: Captures data input at clock edge

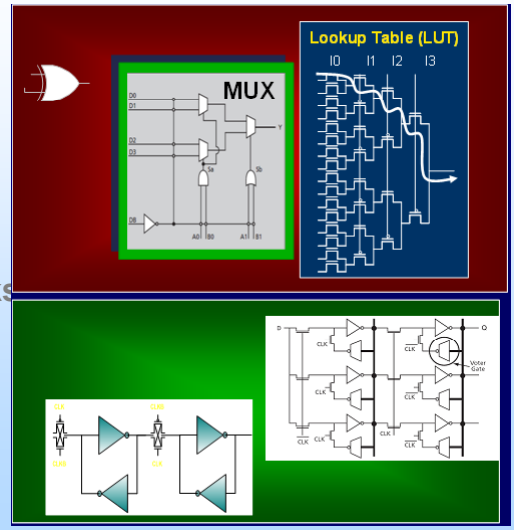
Q=$f(D,t_{clk})$





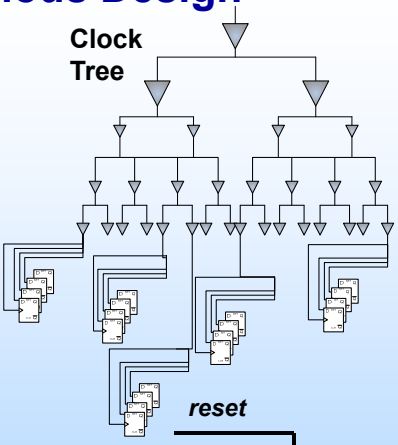
Component Libraries: Basic Designer Building Blocks

- Combinatorial logic blocks
  - Vary in complexity
  - Vary in I/O
- Sequential Memory blocks (DFF)
  - Uses global Clocks
  - Uses global Resets
  - May have mitigation



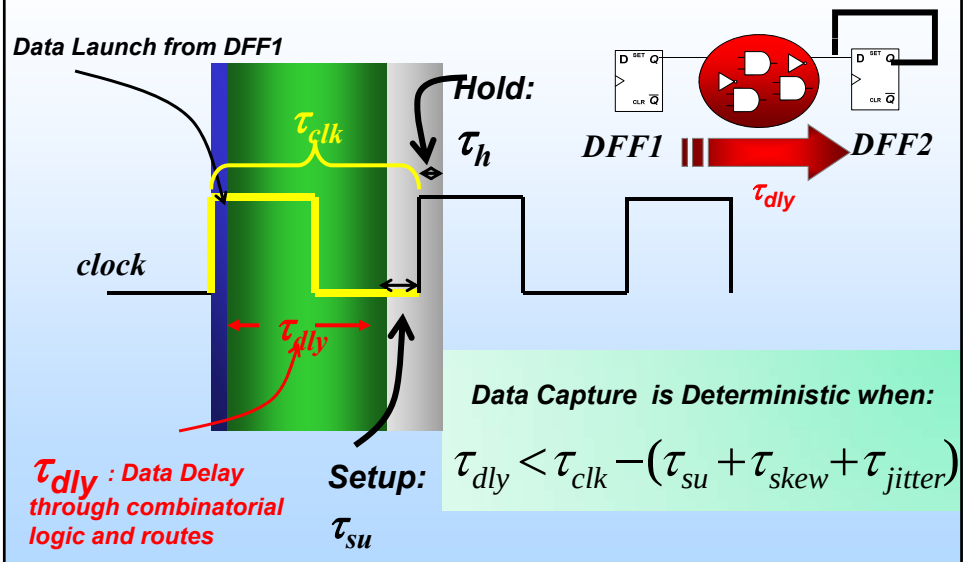
DFF's in a Synchronous Design

- All DFFs are connected to a clock
- A clock tree must be balanced and have minimal skew
- At specified clock edge, DFFs accept input data
- DFFs are expected to be connected to a reset
- Clock periods are limited by the amount of delay between DFF to DFF



DFFs are BOUNDARY POINTs in a synchronous design

Deterministic Data Capture…Adhering to Setup and Hold Time for a DFF



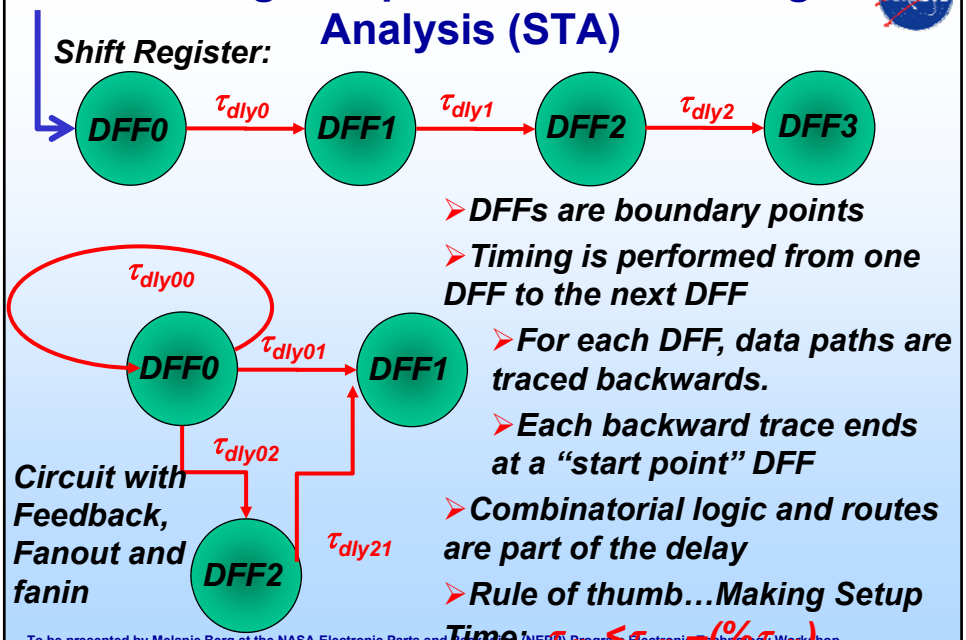
τ_{dly} : Data Delay through combinatorial logic and routes

Data Capture is Deterministic when:

$$\tau_{dly} < \tau_{clk} - (\tau_{su} + \tau_{skew} + \tau_{jitter})$$

To be presented by Melanie Berg at the NASA Electronic Parts and Packaging (NEPP) Program Electronic Technology Workshop, Greenbelt, Maryland, June 28-30, 2011, and published on nepp.nasa.gov.

Making Setup Time: Static Timing Analysis (STA)

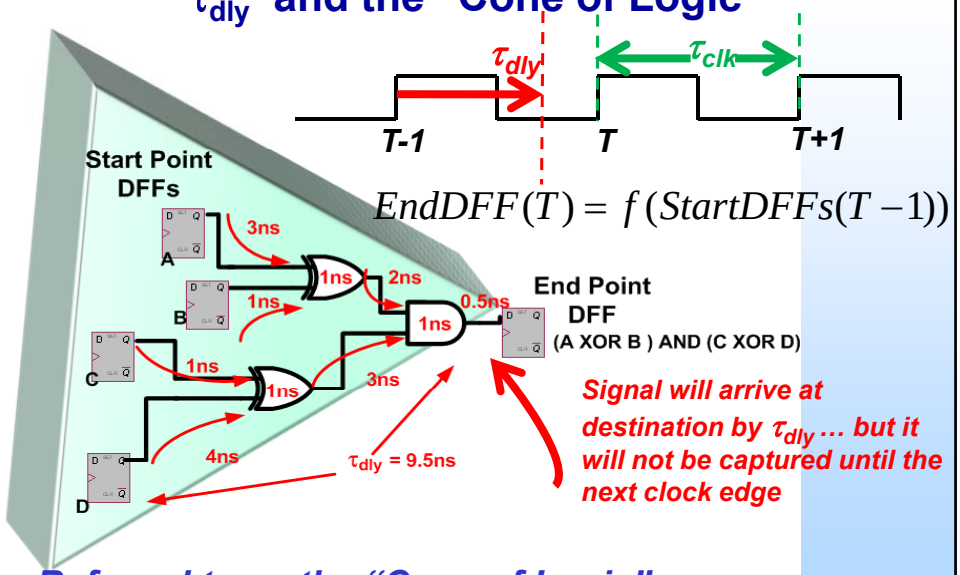


- DFFs are boundary points
- Timing is performed from one DFF to the next DFF
  - For each DFF, data paths are traced backwards.
  - Each backward trace ends at a "start point" DFF
- Combinatorial logic and routes are part of the delay
- Rule of thumb…Making Setup Time: $\tau_{dly} < \tau_{clk} - (\%\,\tau_{clk})$

To be presented by Melanie Berg at the NASA Electronic Parts and Packaging (NEPP) Program Electronic Technology Workshop, Greenbelt, Maryland, June 28-30, 2011, and published on nepp.nasa.gov.

14

Start Point DFFs → End Point DFFs
τ_{dly} and the "Cone of Logic"



$$EndDFF(T) = f(StartDFFs(T-1))$$

Referred to as the "Cone of Logic"

15

System States

- System state is defined by the logic values within all DFFs
- System state is captured at each rising clock edge.
- In between clock edges (intermediate points)
  - (transient … unsettled)
  - Computations are occurring (combinatorial logic)
  - Computations must be settled and signals must arrive at the input data pin of their next DFF with

$$\tau_{dly} < \tau_{clk} - (\tau_{su} + \tau_{skew} + \tau_{jitter})$$

Note: Upsets occur at intermediate points. They become part of the system state if they are captured into the next state

To be presented by Melanie Berg at the NASA Electronic Parts and Packaging (NEPP) Program Electronic Technology Workshop, Greenbelt, Maryland, June 28-30, 2011, and published on nepp.nasa.gov.

16

Synchronous Design Take Away Points

- Basic Blocks: DFFs and Combinatorial logic
- DFFs are boundary points
  - For each DFF (end point) there is a backwards trace to start point DFFs
  - There is delay between start point DFFs and endpoint DFFs
    - Combinatorial logic
    - Routes
- SEE analysis has traditionally been based on utilized DFFs because they are upset capture points

The question is… If an upset occurs will it reach an endpoint DFF?

Data Path Functional Logic

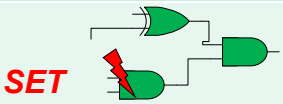
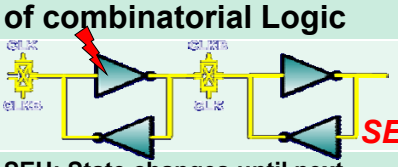
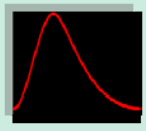
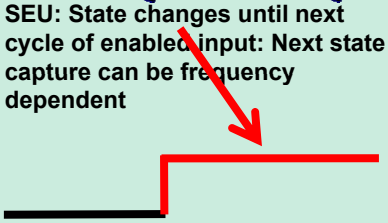
$$P_{functional\ Logic}$$

Configuration versus Data Path (Functional Logic) SEE

- Configuration and Functional logic are separate logic
- Can be implemented with different technology within one device (e.g. antifuse versus CMOS)
- Configuration is static and data paths are not. Requires a different test and analysis approach

SEU and SET Background

Primary functional logic components can be classified into:

Combinatorial	Sequential
Logic function generation (computation)	Captures and holds state of combinatorial Logic
	
SET: Glitch in the combinatorial logic: Capture is frequency dependent 	SEU: State changes until next cycle of enabled input: Next state capture can be frequency dependent 

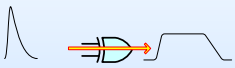
SET effects are nonlinear and are heavily design and state dependent.

Implications of Non-Linear SET Effects

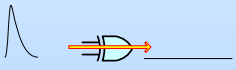
- SET width and amplitude (energy) are non-linear across:
  - LET
  - Clock Frequency
  - Propagation path
  - Number of gates
  - Temperature and voltage?????
- Detailed SET response cannot be extrapolated
- Example: Determining the width of an SET generated from an inverter after it propagates through N-levels of other inverters:
  - Will not supply the width of an SET generated in a different transistor scheme (e.g. XOR, NAND)
  - Will not tell you if the SET will be attenuated or elongated in a different circuit or different frequency
  - Heavily dependent on the original shape of the generated SET

But We Do Know The Basics… Engineering 101!

- An SET in a data path is not an error unless it gets captured by its destination flip-flop (DFF)
- Increase in probability of capture: An SET has the potential to be elongated



- Decrease in probability of capture: An SET can be attenuated.



- How does this apply to real circuits in real systems?

We may not be able to fully characterize the width of the SET for your design… but we can get an idea of its impact to your design

Data Path Functional Logic: Test and Analysis using Shift Regsiters

$$P_{functional\ Logic}$$

Data Path Model and DFF Logic Cones

$$P(fs)_{functional\ Logic}$$

Probability for Functional logic SEE

Evaluate for Each DFF

$$\exists_{DFF}\left(\sum_{j=1}^{\#StartPoint\ DFFs} P(fs)_{DFFSEU \rightarrow SEU(j)} + \sum_{i=1}^{\#Combinatorial\ Logic\ Gates} P(fs)_{SET \rightarrow SEU(i)}\right)$$

Probability for Captured DFF Events

Probability for Captured Combinatorial logic

DFF_k Cone of Logic

All Start Point DFFs and Combinatorial Logic gates that feed into End Point DFF under Evaluation (DFF_k):

Combinatorial Logic Contribution to System Error in a Synchronous System

SETs and a Synchronous System

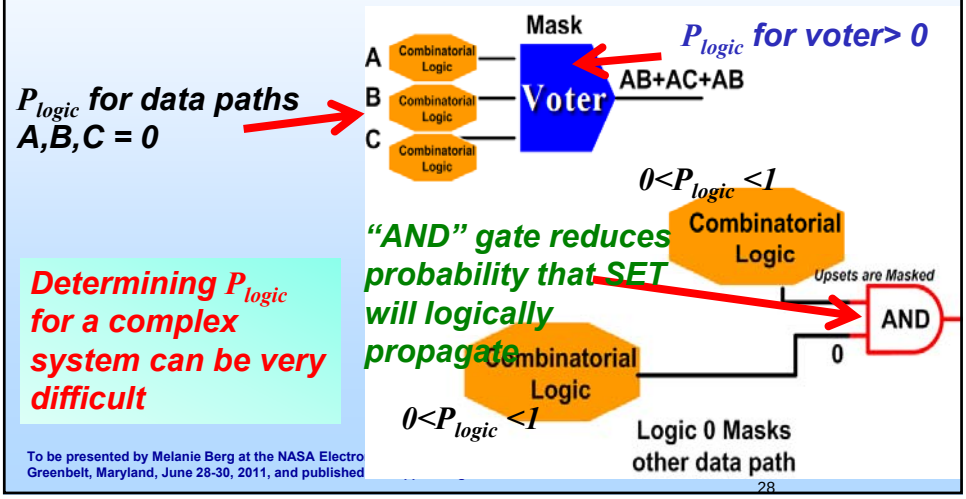
- Generation (P_{gen})
- Propagation (P_{prop})
- Logic Masking ($P_{logic.}$)
- Capture

SET Propagation: P_{prop}

- In order for the data path SET to become an upset, it must propagate to its destination DFF
- In REAG model, P_{prop} is defined as the probability that an SET can propagate through an electrical medium to reach a destination REGARDLESS of logic path.
  - The SET will travel through routes and potentially other combinatorial logic gates before it reaches its destination DFF.
  - Due to the capacitance of the routes and other combinatorial logic blocks, SETs will be reshaped as they propagate.
    - Attenuation
    - Elongation

SET Logic Masking: P_{logic}

- ***P_{logic}: Probability that a SET can logically propagate through a cone of logic. Based on state of the combinatorial logic gates and their potential masking.***



SET Capture at Destination DFF

Each combinatorial element can generate a transient. The transient width will be a fraction of the clock period for a synchronous design in a CMOS process.

$$P(fs)_{SET \rightarrow SEU} \propto \frac{\tau_{width}}{\tau_{clk}}$$

Probability of capture is proportional to the width of the transient as seen from the destination DFF

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

Data Path Model and SETs

$$P(fs)_{functional\ Logic}$$

$$\exists_{DFF}\left(\sum_{j=1}^{\#FaninDFFs} P(fs)_{DFFSEU \rightarrow SEU} + P(fs)_{SET \rightarrow SEU}\right)$$

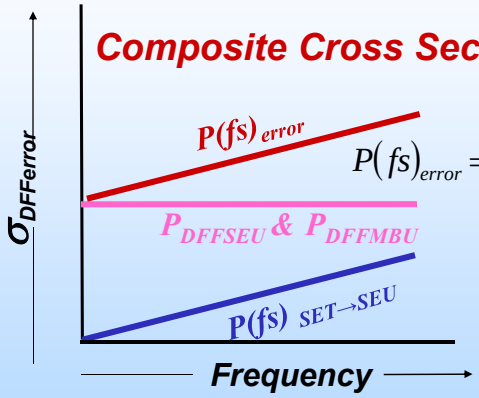
$$\sum_{i=1}^{\#CombinatorialCells} P_{gen(i)} P_{prop(i)} P_{logic(i)} \tau_{width(i)}\ fs$$

$$\sum_{i=1}^{\#CombinatorialCells} P_{gen(i)} P_{prop(i)} \tau_{width(i)}\ fs$$

Upper Bound SET P_{logic}=1

DFF Contribution to System Error in a Synchronous System

Conventional Theory: DFF Upsets Have a Static Component+Dynamic Component



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

~0

Takes into account upsets from combinatorial logic in DFF data path and the DFF potential for flipping its state

Does not fully characterize DFF upsets as they pertain to a synchronous system

SEUs and a Synchronous System: New Stuff

- Generation (P_{DFFSEU})
- Propagation (P_{prop})
- Logic Masking ($P_{logic.}$)
- Capture

All Components comprise:

$$P_{DFFSEU \rightarrow SEU}$$

Generation P_{DFFSEU}

- DFF SEU generation pertains to a SET flipping the state of a start point DFF.
- P_{DFFSEU} is a frequency independent event that can occur anywhere within a clock period.
- However, $P_{DFFSEU \rightarrow SEU}$ the probability of this event manifesting into the system as an upset is frequency dependent.
- Upper-bound calculations can assume frequency independent capture

System Level Evaluation:

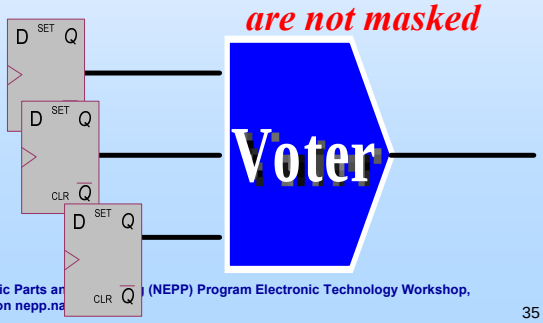
The effects of SEUs in DFFs are frequency dependent

Logic Masking DFFs… P_{logic}

- Logic masking for DFF start points is similar to logic masking of combinatorial logic.
- DFF logic masking is generally the point where TMR is inserted

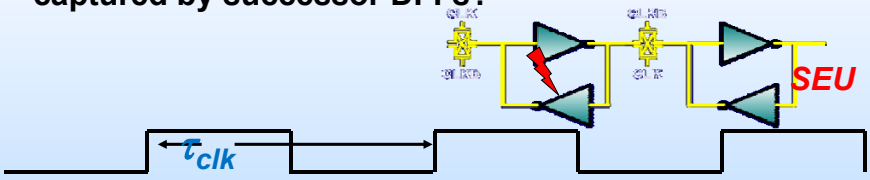
P_{logic}>0 for Voter… its upsets are not masked

P_{logic}=0 for DFFs… their upsets are masked

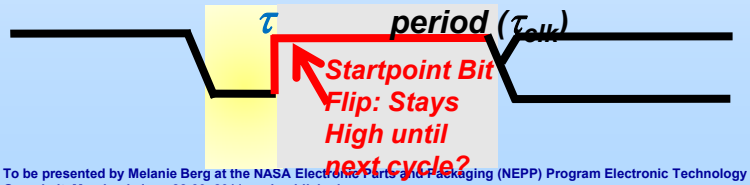


Capture: SEU System Effects

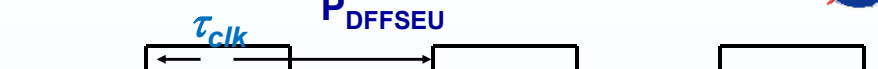
- If a DFF is affected by an SEU it will change its state somewhere within a clock cycle at time τ
- The question is… will this change-in-state be captured by successor DFFs?



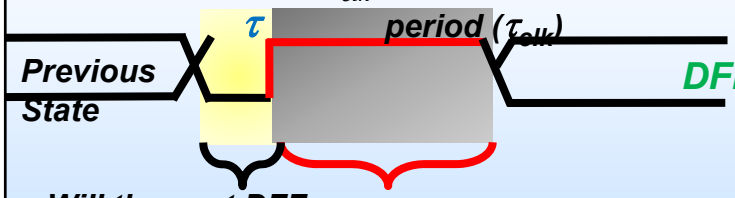
$0 < \tau < \tau_{clk}$ *τ Is defined to be within 1 clock period (τ_{clk})*



DFF Upsets (SEUs) and Next State Capture: P_{DFFSEU}



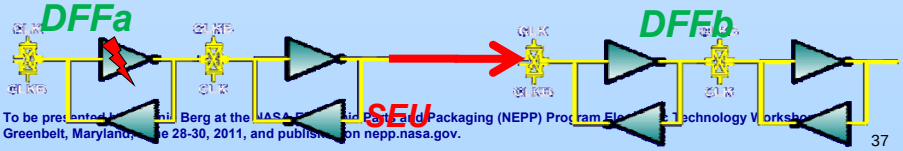
$0 < \tau < \tau_{clk}$ *τ Is defined to be within 1 clock period (τ_{clk})*



Will the next DFF capture the correct value?

OR…

Will the next DFF capture the upset (P_{DFFSEU})?

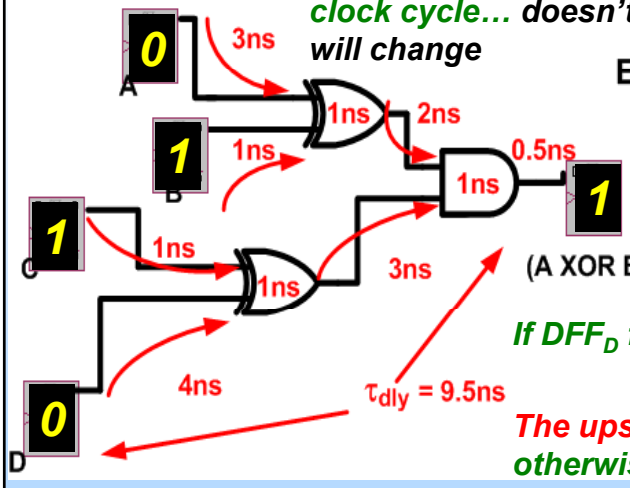


SEU Capture Example: Assume τ_{clk}=15ns

Start Point DFFs

A,B,C, and D will capture new data every clock cycle… doesn't mean their values will change

End Point DFF



(A XOR B) AND (C XOR D)

If DFF_D flips its state…

$0<\tau<(5.5)$ns

The upset will get caught… otherwise it's as if the event never occurred

Percentage of Clock Cycle for SEU Capture:

$$\tau < \tau_{clk} - \tau_{dly}$$

Upset is caught within this timeframe

$$\frac{\tau_{clk} - \tau_{dly}}{\tau_{clk}}$$

Percentage of clock period for upset capture

$$1 - \frac{\tau_{dly}}{\tau_{clk}}$$

$$1 - \tau_{dly}\ fs$$

Percentage of clock period for upset capture wrt to frequency

Data Path Upsets and Start Point DFFs

$$P(fs)_{functionalLogic}$$

$$\exists_{DFF}\left(\sum_{j=1}^{\#StartPoint\ DFFs} P(fs)_{DFFSEU \rightarrow SEU} + P(fs)_{SET \rightarrow SEU}\right)$$

$$\sum_{j=1}^{\#StartPoint\ DFFs} P_{DFFSEU(j)} P_{logic(j)} (1 - \tau_{dly(j)}\ fs)$$

$$\sum_{j=1}^{\#StartPoint\ DFFs} P_{DFFSEU(j)} (1 - \tau_{dly(j)}\ fs)$$

If No redundancy or masking is implemented P_{logic}=1 (e.g. unmitigated shift register)

NASA REAG FPGA Susceptibility Model (Probability of Capture)

$$P(fs)_{functional\ \ Logic}$$

$$\exists_{DFF}\left(P(fs)_{DFFSEU \rightarrow SEU} + P(fs)_{SET \rightarrow SEU}\right)$$

$$\exists_{DFF}\left(\left(\sum_{j=1}^{\#FaninDFFs} P_{DFFSEU(j)}(1-\tau_{dly(j)}fs)P_{logic(j)}\right) + \sum_{i=1}^{\#CombinatorialLogicGates}(P_{gen(i)}P_{prop(i)}P_{logic}\tau_{width(i)}fs)\right)$$

NASA REAG FPGA Upper Bound Susceptibility Model

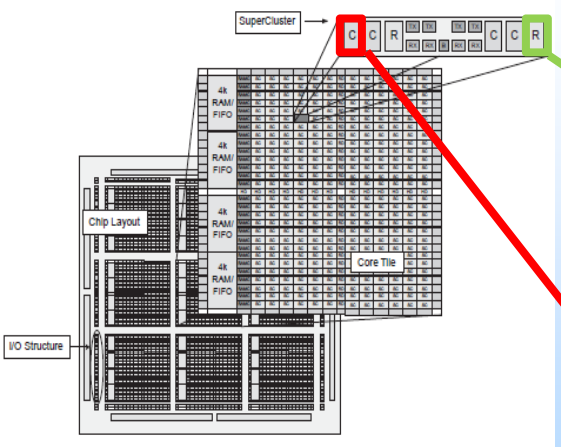
$$\exists_{DFF}\left(\left(\sum_{j=1}^{\#StartPointDFFs} P_{DFFSEU(j)}(1-\tau_{dly(j)}fs)P_{logic(j)}\right) + \sum_{i=1}^{\#CombinatorialLogicGates}(P_{gen(i)}P_{prop(i)}P_{logic}\tau_{width(i)}fs)\right)$$

Upper-bound assumes Plogic=1 (no mitigation) and NO DFF fs dependency

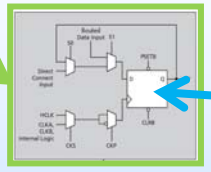
$$\exists_{DFF}\left(\sum_{j=1}^{\#StartPointDFFs} P_{DFFSEU} + \sum_{i=1}^{\#CombinatorialLogicGates}(P_{gen(i)}P_{prop(i)}\tau_{width(i)}fs)\right)$$

NASA REAG Models + Heavy Ion Data: RTAXs

RTAXs FPGA Core Logic: Basic Building Blocks are R-CELLs and C-CELLs

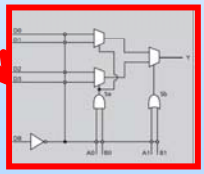


R-CELL: Sequential + Combinatorial



DFF

C-CELL: Combinatorial



Combinatorial Logic is susceptible to Single Event Transients (SETs)

(RCELLs, CCELLs, and buffers are susceptible to SETs)

RTAXs has Localized TMR at DFF (LTMR)… P_{logic}=0 hence $P_{DFFSEU \rightarrow SEU}$=0

$$P(fs)_{functionalLogic}$$

$$\exists_{DFF}\left(\sum_{j=1}^{\#FaninDFFs} P(fs)_{DFFSEU \rightarrow SEU} + P(fs)_{SET \rightarrow SEU} \right)$$

$$\sum_{j=1}^{\#FaninDFFs} P_{DFFSEU(j)} P_{logic(j)} (1 - \tau_{dly(j)} fs)$$

0

0

Localized TMR (LTMR) at every DFF: P_{logic}=0

Model Application to RTAXs

$$P(fs)_{error} \propto P_{Configuration} + P(fs)_{functionalLogic} + P_{SEFI}$$

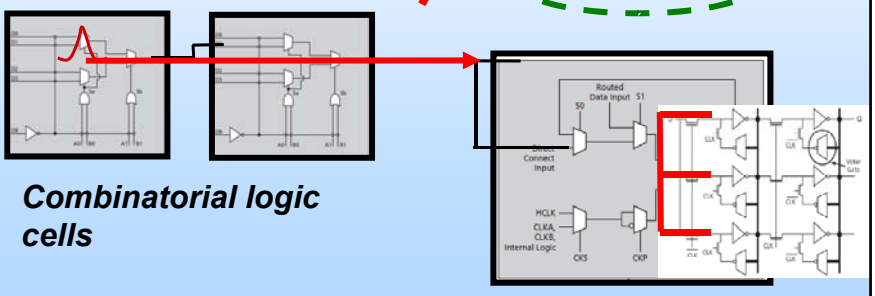
Design Specific SEE upset rate

Configuration SEE upset rate

Functional logic SEE upset rate

Single Event functional interrupt

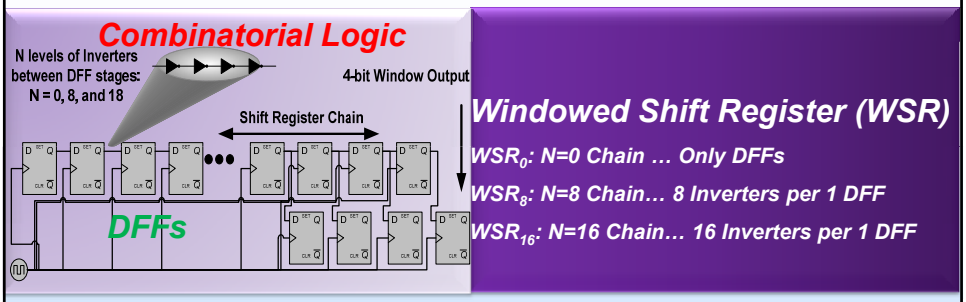
$$P_{DFFSEU \rightarrow SEU} + P(fs)_{SET \rightarrow SEU}$$



Combinatorial logic cells

DFF

Testing Combinatorial Logic Contributions to SEU Cross-Sections: WSRs with Inverters



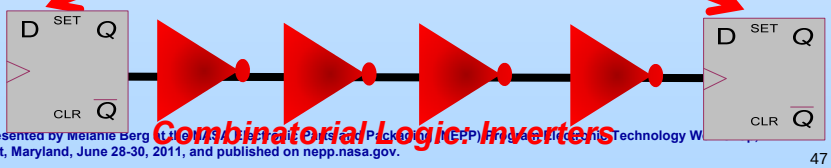
Windowed Shift Register (WSR)

- WSR_0: N=0 Chain … Only DFFs
- WSR_8: N=8 Chain… 8 Inverters per 1 DFF
- WSR_{16}: N=16 Chain… 16 Inverters per 1 DFF

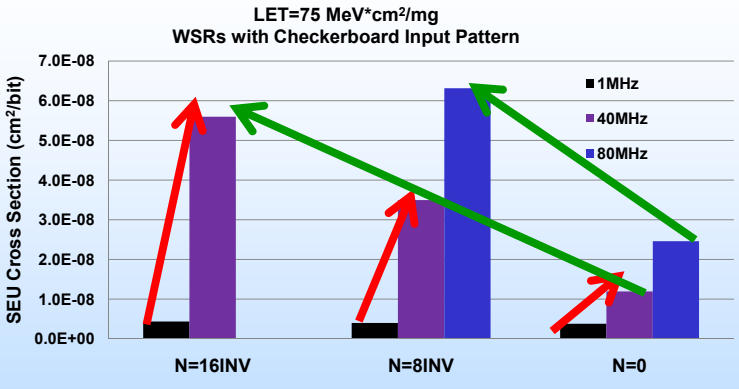
Think of DFFs as your boundary points… They capture SETs

$$P_{SET\rightarrow SEU}(fs) \propto \sum_{j=1}^{\#RCELLCombinatorial} P_{gen(j)} P_{prop(j)} \tau_{width(j)}\, fs + \sum_{i=1}^{\#CCELLCombinatorial} P_{gen(i)} P_{prop(i)} \tau_{width(i)}\, fs$$

DFFs DFFs



RTAXs: SET Capture across LET affects SEU Cross Sections

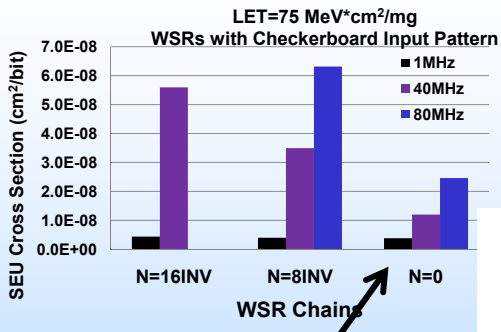


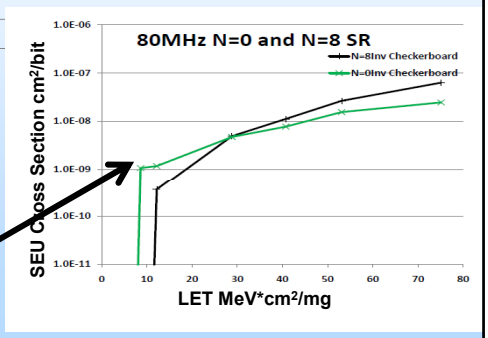
$$P(fs)_{SET\rightarrow SEU} \propto \sum_{j=1}^{\#RCELLCombinatorial} P_{gen(j)} P_{prop(j)} \tau_{width(j)}\, fs + \sum_{i=1}^{\#CCELLCombinatorial} P_{gen(i)} P_{prop(i)} \tau_{width(i)}\, fs$$

Frequency: Increase Frequency→ Increase Cross section

Combinatorial logic gates: Increase CCELLs → Increase Cross section

RTAXs: SET Capture across LET affects SEU Cross Sections





Non-Linear Effects:

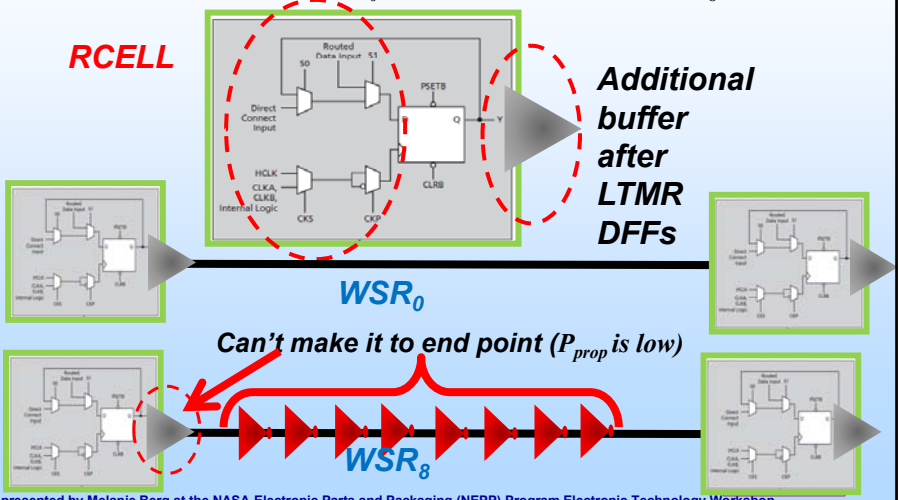
- ***WSR_0 has the lowest SEU Cross Sections at High LETs***
- ***WSR_0 > WSR_8 SEU Cross Sections at Low LETs***

Low LETs: attenuation of Single Event Transients (SETs) at low LET values

Why is WSR_0 > WSR_8 for Low LET?

$$(P(fs)_{SET \rightarrow SEU})\big|_{WSR_0} > (P(fs)_{SET \rightarrow SEU})\big|_{WSR_8}$$

RCELL



Additional buffer after LTMR DFFs

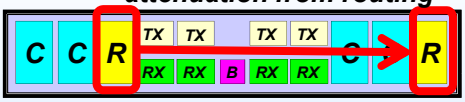
WSR_0

Can't make it to end point (P_{prop} is low)

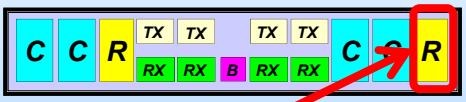
WSR_8

In addition… Low LET P_{prop} for WSR_8?

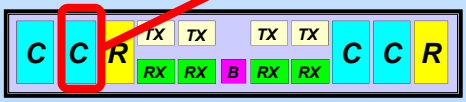
More WSR_0 Chains share the same super cluster (less attenuation from routing



Super Cluster



Super Cluster



Super Cluster

To be presented by Melanie Berg at the NASA Electronic Parts and Packaging (NEPP) Program Electronic Technology Workshop, Greenbelt, Maryland, June 28-30, 2011, and published on nepp.nasa.gov.

51

Why is WSR_0 > WSR_8 for Low LET? Proof using the REAG Model

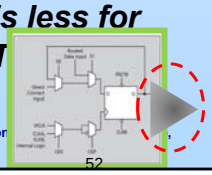
P_{prop} is low

$$\sum_{j=1}^{\#RCELLCombinatorial} P_{gen(j)} P_{prop(j)} \tau_{width(j)} fs + \sum_{i=1}^{\#CCELLCombinatorial} P_{gen(i)} P_{prop(i)} \tau_{width(i)} fs \Big|_{WSR_8} < \sum_{j=1}^{\#RCELLCombinatorial} P_{gen(j)} P_{prop(j)} \tau_{width(j)} fs \Big|_{WSR_0}$$

WSR_8 ***WSR_0***

$$\sum_{j=1}^{\#RCELLCombinatorial} P_{gen(j)} P_{prop(j)} \tau_{width(j)} fs \Big|_{WSR_8} < \sum_{j=1}^{\#RCELLCombinatorial} P_{gen(j)} P_{prop(j)} \tau_{width(j)} fs \Big|_{WSR_0}$$

Number of combinatorial logic gates that contribute to the SEU cross section is less for WSR_8 because the RCELL Buffer SET get through the long chain



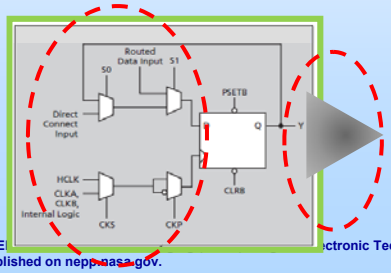
To be presented by Melanie Berg at the NASA Electronic Parts and Packaging (NEPP) Program Electron
Greenbelt, Maryland, June 28-30, 2011, and published on nepp.nasa.gov.

52

Summary…
Why is WSR_0 > WSR_8 for Low LET?

- RTAXs RCELLs have combinatorial logic
- SETs that occur in the buffer of the RCELL cannot make it down a WSR_8 chain
- WSR8 also has longer routes for its last CCELL block connection to its RCELL block

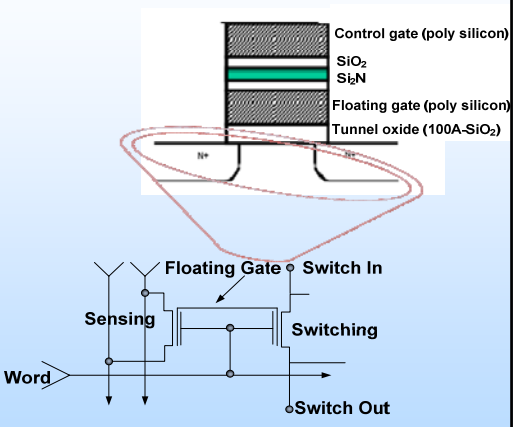
RCELL



NASA REAG Models + Heavy Ion Data: ProASIC

Background: Micro-Semi (Actel) ProASIC3 Flash Based FPGA

- Commercial device
- Configuration is flash based and has proven to be almost immune to SEUs
- No embedded mitigation in device
- Evaluation of user mitigation insertion has been performed



Actel ProASIC3 Study

- Functional Logic Designs Under Test:
  - Six WSR strings with various levels of combinatorial logic
  - Counter Arrays
  - Hamming Code 3 State Machine.

$$P(fs)_{error} \propto P_{Configuration} + P(fs)_{functionalLogic} + P_{SEFI}$$

$$\exists_{DFF} \left(P(fs)_{DFFSEU \rightarrow SEU} + P(fs)_{SET \rightarrow SEU} \right)$$

- Apply LTMR and DTMR Mitigation to WSR strings in order to reduce overall error
- Heavy Ion Testing

ProASIC Non-Mitigated Shift Register Data Path Model

Evaluate for Each DFF

$$\exists_{DFF}\left(\sum_{j=1}^{\#FaninDFFs} P(fs)_{DFFSEU\rightarrow SEU(j)} + \sum_{i=1}^{\#Combinatorial LogicGates} P(fs)_{SET\rightarrow SEU(i)}\right)$$

DFF_k Cone of Logic

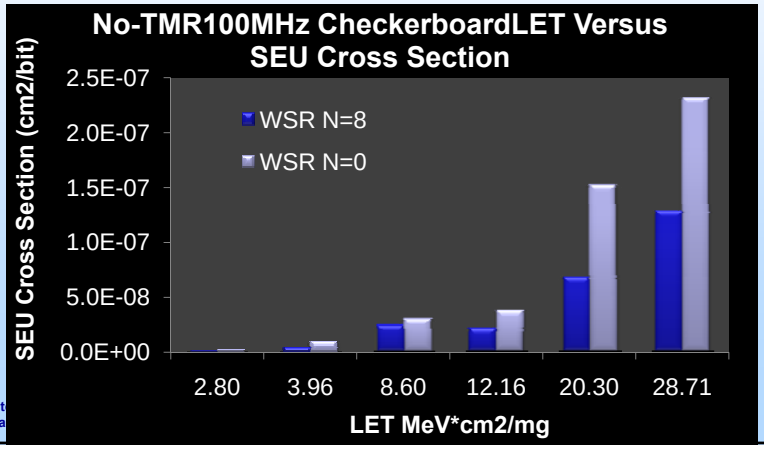
Each End point only has one DFF start point.

WSR chains dictate number of combinatorial logic gates

$$\exists_{DFF}\left(P(fs)_{DFFSEU\rightarrow SEU} + \sum_{i=1}^{\#Combinatorial LogicGates} P(fs)_{SET\rightarrow SEU(i)}\right)$$

SEE Test Results: Windowed Shift Registers (WSRs) No-TMR

- **N=0: WSR with only DFFs**
- **N=8: WSR with 8 inverters between each DFF stage**
- ***No Mitigation: $P(fs)_{DFFSEU\rightarrow SEU} > P(fs)_{SET\rightarrow SEU}$***

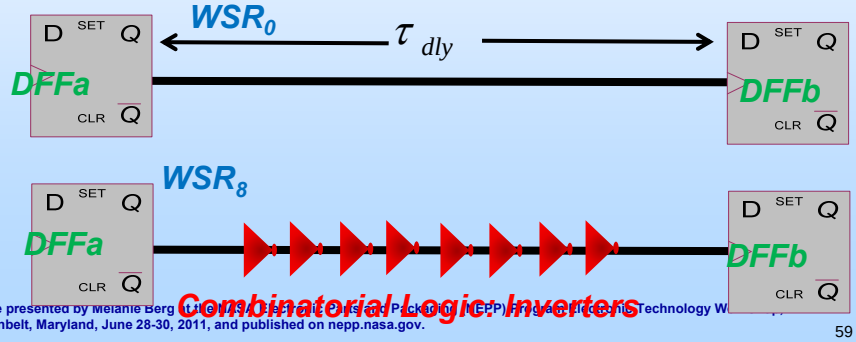


Why is WSR0 > WSR8 for Non-Mitigated ProASIC: τ_{dly}

$WSR_0\ \tau_{dly} << WSR_8\ \tau_{dly}$

For a clock period = τ_{clk}, if DFFa flips @ time $\tau > (\tau_{clk} - \tau_{dly})$ then DFFb will never capture the upset.

$(0 < \tau < \tau_{clk})$



New Research!...Why is WSR_0 > WSR_8 for The Non-Mitigated ProASIC3 Design?

No combinatorial in WSR_0

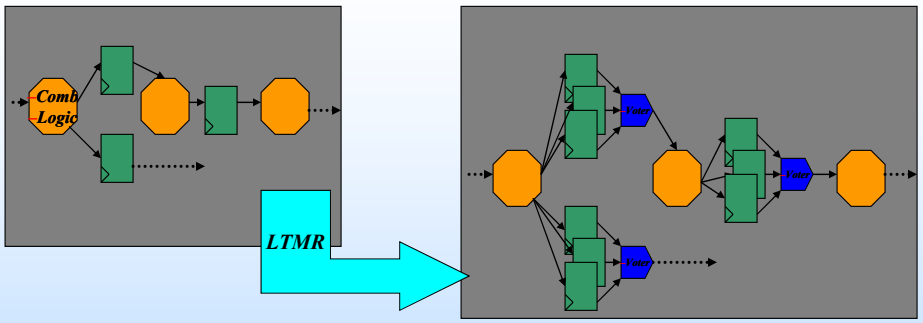
$$(P(fs)_{DFFSEU\rightarrow SEU} + P(fs)_{SET\rightarrow SEU})\big|_{WSR_0} > (P(fs)_{DFFSEU\rightarrow SEU} + P(fs)_{SET\rightarrow SEU})\big|_{WSR_8}$$

Upsets in WSR_0 Chain — *Upsets in WSR_8 Chain*

$$P_{DFFSEU}\left(1 - \frac{\tau_{dly}\big|_{WSR_0}}{\tau_{clk}}\right) > P_{DFFSEU}\left(1 - \frac{\tau_{dly}\big|_{WSR_8}}{\tau_{clk}}\right) + \sum_{i=1}^{8} P(fs)_{SET\rightarrow SEU(i)}$$

$$P_{DFFSEU} > \frac{\tau_{clk}}{\tau_{dly}\big|_{WSR_8} - \tau_{dly}\big|_{WSR0}} \sum_{i=1}^{8} P(fs)_{SET\rightarrow SEU(i)}$$

Local Triple Modular Redundancy (LTMR): Triple DFFs Only

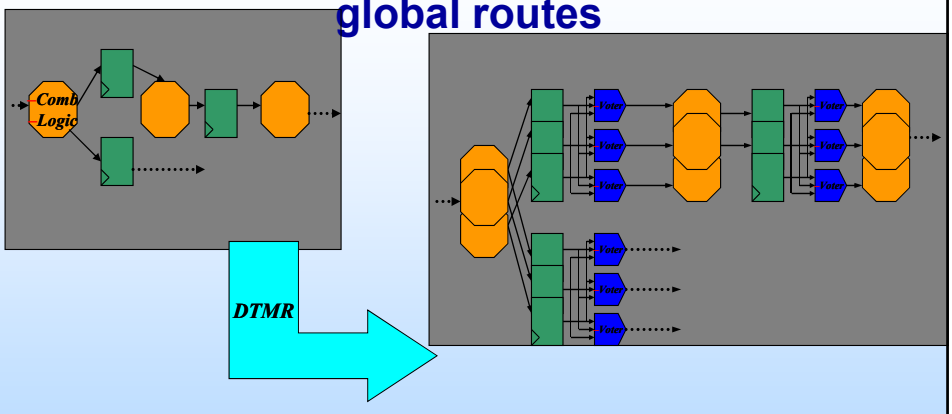


- Triple Each DFF + Vote+ Feedback Correct at DFF
- Unprotected:
  - Clocks and Resets… SEFI
  - Transients (SET->SEU)
  - Internal/hidden device logic: SEFI

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

? Low ?

Distributed Modular Redundancy (DTMR)… Triple everything except global routes



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

Low Low Low

ProASIC LTMR Shift Register Data Path Model

DFF_k Cone of Logic

Each End point only has one DFF start point.

WSR chains dictate number of combinatorial logic gates

Evaluate for Each DFF

$$\exists_{DFF}\left(\sum_{j=1}^{\#StartPo\ int\ DFFs} P(fs)_{DFFSEU\rightarrow SEU(j)} + \sum_{i=1}^{\#Combinator\ ialLogicGa\ tes} P(fs)_{SET\rightarrow SEU(i)}\right)$$

$$\exists_{DFF}\left(P(fs)_{DFFSEU\rightarrow SEU} + \sum_{i=1}^{\#Combinator\ ialLogicGa\ tes} P(fs)_{SET\rightarrow SEU(i)}\right)$$

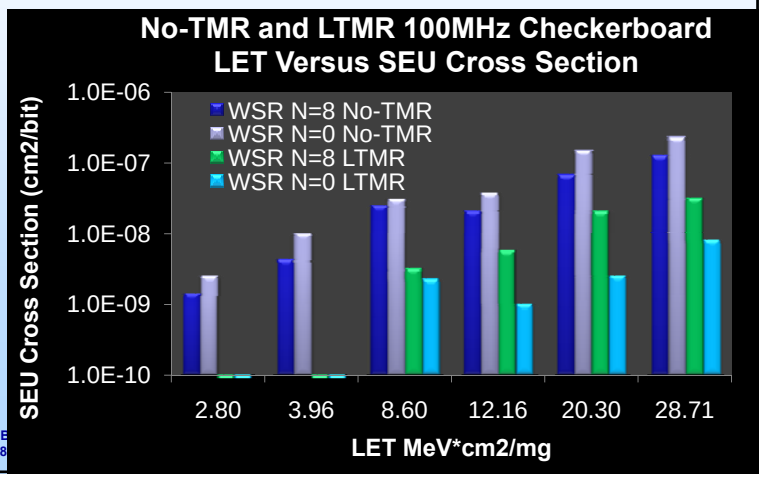
$$\exists_{DFF}\left(\sum_{i=1}^{\#Combinator\ ialLogicGa\ tes} P(fs)_{SET\rightarrow SEU(i)}\right)$$

$P_{DFFSEU\rightarrow SEU}$ =0 if ($\tau > \tau_{clk}-\tau_{dly}$)
What Else Does This Mean?

- **As frequency increases (decrease τ_{clk}), $P_{DFFSEU\rightarrow SEU}$ decreases**
  - **Systems with unmitigated DFFs that operate at higher frequencies or that contain data paths with large τ_{dly}, will have lower SEU cross Sections**
  - **Systems with unmitigated DFFs that run at slower speeds can be more susceptible to upsets… Yes!**
  - **Opposite behavior than combinatorial logic**
- **For RHBD DFFs, this can be used to determine the RHBD effectiveness**
  - **RHBD DFFs should not be the dominant source of SEU upsets.**
  - **If SEU cross sections decrease as frequency increases, then DFFs are dominant and RHBD is not as effective as projected.**

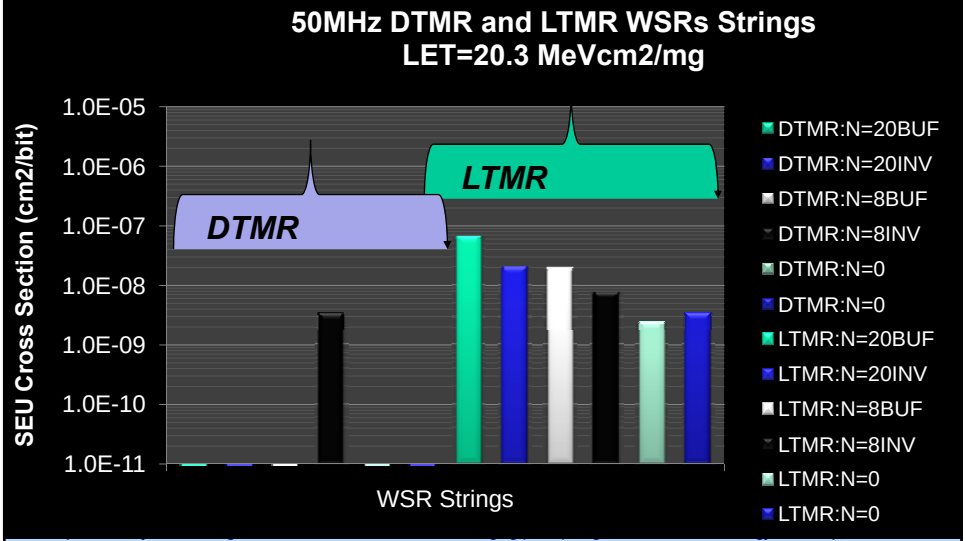
SEE Test Results: Windowed Shift Registers (WSRs) No-TMR versus TMR

- LTMR is effective and has reduced P_{DFFSEU}
- $P(fs)_{DFFSEU} < P(fs)_{SET \rightarrow SEU}$



To be presented by Melanie B
Greenbelt, Maryland, June 28

DTMR vs. LTMR at LET=20.3MeVcm²/mg



Greenbelt, Maryland, June 28-30, 2011, and published on nepp.nasa.gov.

66