Incorporating Probability Models of Complex Test Structures to Perform Technology Independent FPGA Single Event Upset Analysis

M. Berg, Member IEEE, H. Kim, M. Friendlich, C. Perez, C. Seidleck, K. LaBel, Member IEEE

Abstract—We present SEU test and analysis of the Microsemi ProASIC3 FPGA. SEU Probability models are incorporated for device evaluation. Included is a comparison to the RTAXS FPGA illustrating the effectiveness of the overall testing methodology.

Index Terms—FPGA, ProASIC3 versus RTAXS, SEU, Test and Analysis

I. INTRODUCTION

An effective method for modeling Single Event Upset (SEU) probabilities in Field Programmable Gate Array (FPGA) devices has been developed and presented[1][2] by NASA Goddard Radiation Effects and Analysis Group (REAG). The SEU Probability model is used by REAG to characterize and analyze upper-bound SEU cross sections (σ_{SEU}) for synchronous digital systems.

The main control of a synchronous design is its clock. The clocking scheme and speed dictate circuit interface techniques and the amount of computation performed per clock period. It has been shown that while synchronous designs are strictly dependent on their clock period (or frequency of operation), their SEU cross sections are also dependent on their operational frequency [1]-[7]. Subsequently, FPGA σ_{SEU} models must take frequency into account. Note that operational frequency (f_s) is understood to be the inverse of clock period (τ_{clk}) as in (1).

\[
\tau_{clk} = \frac{1}{f_s}
\]

With the REAG FPGA test methodology, SEUs that occur during radiation testing are differentiated and categorized in order to enhance device evaluation. The REAG-FPGA-SEU model is based on a top-down approach. The top-level of the model (P(f_s)_{error}=σ_{SEU}) contains three major components (2):

- Configuration SEU cross section (P_{configuration})
- Data path or functional logic SEU cross section (P(f_s)_{FunctionalLogic})
- Single Event Functional Logic SEU cross section (P_{SEF})

\[
P(f_s)_{error} \propto P_{configuration} + P(f_s)_{FunctionalLogic} + P_{SEF}
\]

The importance of this subject matter is to present Microsemi ProASIC3 FPGA SEU behavior under a variety of conditions while illustrating how the REAG-FPGA-SEU model facilitates a detailed analysis that spans across FPGA device technologies. Microsemi RTAXS data [2][6] are used as a comparison.

II. P(f_s)_{error} MODEL COMPONENTS

Before SEU radiation testing is performed, general models of expected SEU probabilities based on mitigation and device logic structure are constructed. The top-level models are used as reference points during testing. During the analysis phase, lower levels of the model are developed to reflect SEU data obtained during radiation testing.

The following is a more detailed discussion of each element in (2).

A. FPGA Configuration and P_{configuration}

Although contained within the FPGA device, the configuration is a separate portion of technology than the functional logic. Accordingly, it has its own categorization of upsets [1]. The RTAXS has an antifuse configuration [8] while the ProASIC3 has a flash configuration [9]. It has been shown through Configuration SEE radiation testing of Antifuse[6][8] and Flash technologies[7][9] that P_{configuration} is considered near zero as in (3).

**Antifuse and Flash Configuration:**

\[
P_{configuration} \rightarrow 0;
\]
Because $P_{\text{configuration}}$ is essentially zero for these devices, the following discussion focuses on $P(\text{fs})_{\text{functional Logic}}$ and $P_{\text{SEFI}}$.

### B. Functional Logic Data Path Upsets and $P(\text{fs})_{\text{functional Logic}}$

The functional logic data path of a synchronous design is comprised of: Combinatorial Logic, Flip-Flops (DFFs), and Routes. Table 1 illustrates upset types that can potentially occur in an FPGA data path. In a synchronous design, every DFF is connected to a global clock signal. Because DFFs are master-slave edge-triggered flip-flops, their internal structure uses both a global clock (CLK) and its logical inverse (CLKB), as shown in Table 1.

#### Table 1: Combinatorial Logic Versus Sequential Logic

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logic function generation</td>
<td>Captures and holds state of combinatorial logic</td>
</tr>
<tr>
<td>SET: Glitch in the combinatorial logic: Capture its frequency of occurrence</td>
<td></td>
</tr>
<tr>
<td>SEU: Next state capture can be frequency dependent</td>
<td></td>
</tr>
</tbody>
</table>

1) **Synchronous Design Concepts and the Functional Data Path**

The essence of synchronous design considers DFFs as boundary points. In a design, each boundary point DFF will have a cone of logic feeding it. The cone is defined to be a backwards trace from an End-Point DFF that stops at its previous stage DFFs (Start-Point DFFs). The trace includes the Start-Point DFFs and all combinatorial logic within the path. One cone of logic is illustrated in Fig. 1.

#### Table 2: Definition of Terms

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P(\text{fs})_{\text{DIFFSEU} \rightarrow \text{SEU}}$</td>
<td>Probability that the Start-Point DFF will incur a SEU and that it will be captured by an End-Point</td>
</tr>
<tr>
<td>$P(\text{fs})_{\text{SET} \rightarrow \text{SEU}}$</td>
<td>Probability that the Start-Point DFF will incur a SEU and it will be captured by an End-Point</td>
</tr>
<tr>
<td>$P_{\text{DIFFSEU}}$</td>
<td>Probability the Start-Point DFF will incur a SEU</td>
</tr>
<tr>
<td>$\tau$</td>
<td>Moment in within a clock period when a Start-Point DFF flips its state: $0 &lt; \tau &lt; \tau_{\text{clk}}$</td>
</tr>
<tr>
<td>$1 - \tau_{\text{dly}}$</td>
<td>Portion of clock cycle that the End-Point DFF can capture a Start-Point DFF SEU before the next clock edge. Assumes the SEU Start-Point DFF is always enabled and will have a valid value at the next clock edge</td>
</tr>
<tr>
<td>$P_{\text{set}}$</td>
<td>Probability a combinatorial gate will incur a SET</td>
</tr>
<tr>
<td>$P_{\text{prop}}$</td>
<td>Probability the SET can propagate to an End-Point DFF</td>
</tr>
<tr>
<td>$\tau_{\text{width}}$</td>
<td>SET width to clock period ratio</td>
</tr>
</tbody>
</table>

2) **$P(\text{fs})_{\text{functional Logic}}$ Evaluation for Synchronous Designs**

In order to analyze $P(\text{fs})_{\text{functional Logic}}$, each DFF is evaluated as an End-Point with a cone of logic backwards trace. Equation (4) is a breakdown of $P(\text{fs})_{\text{functional Logic}}$ by Start-Point DFF and combinatorial logic.

3) **Capturing Start-Point DFF Upsets ($P(\text{fs})_{\text{DIFFSEU} \rightarrow \text{SEU}}$)**

If a Start-Point DFF incurs an SEU ($P_{\text{DIFFSEU}}$), it will occur at time $\tau$ as a single sided function (see Table 1 and Table 2) somewhere within a clock period ($\tau_{\text{clk}}$). Because the SEU occurs between clock edges, it is only considered as an intermediate state. It will not manifest as a system upset and become a part of the state space unless an End-Point DFF captures the effects of the single sided upset. DFFs only capture the state of its data-pin on clock edges.

The key point to this analysis is that it takes time for the flipped Start-Point state to make its way through the cone of logic to the End-Point. $\tau_{\text{dly}}$ is the delay from a Start-Point DFF to an End-Point DFF within a cone of logic. There is a unique $\tau_{\text{dly}}$ for every Start-Point to End-Point. By definition of synchronous design: $\tau_{\text{dly}} < \tau_{\text{clk}}$.

An End-Point will only capture the Start-Point upset if it occurs at $\tau$ such that after propagating through the delay path ($\tau_{\text{dly}}$), the single sided upset arrives at the End-Point data-pin prior to the next clock edge as shown in Fig. 2 and (5).
If DFF can be captured by an End-Point DFF is shown (6).

Fig. 2: Will the End-Point DFF capture the Start-Point SEU? Capture occurs if \( \tau < \tau_{\text{clk}} - \tau_{\text{dly}} \), giving the one-sided signal enough time to reach the End-Point DFF. Otherwise it's as if the event never occurred.

\[
\tau < \tau_{\text{clk}} - \tau_{\text{dly}} \tag{5}
\]

The portion of the clock period that a Start-Point DFF SEU can be captured by an End-Point DFF is shown (6).

\[
\frac{\tau}{\tau_{\text{clk}}} < 1 - \frac{\tau_{\text{dly}}}{\tau_{\text{clk}}} = 1 - \tau_{\text{dly}}.f_{\text{s}} \tag{6}
\]

The probability that \( P_{\text{DFFSEU}} \) will manifest as a system error \( (P(f_{\text{s}})_{\text{DFFSEU→SEU}}) \) is reflected (7).

\[
P(f_{\text{s}})_{\text{DFFSEU→SEU}} \propto \exists_{\text{DFF}} \left( \sum_{j=1}^{\#\text{StartPoint DFFs}} P_{\text{DFFSEU}(j)}(1 - \tau_{\text{dly}(j)}.f_{\text{s}}) \right) \tag{7}
\]

4) System upsets due to combinatorial logic \( (P(f_{\text{s}})_{\text{SET→SEU}}) \)

If an SET occurs in a combinatorial logic gate within the cone of logic for an End-Point DFF, it has the possibility of being captured by its End-Point with a probability of \( P(f_{\text{s}})_{\text{SET→SEU}} \). It has been shown [1][2] that the upper-bound \( P(f_{\text{s}})_{\text{SET→SEU}} \) for a synchronous design is proportional to the following probabilities: generation of a SET \( (P_{gen}) \), propagation of the SET \( (P_{prop}) \), and capture of the SET. In addition, the SET capture is proportional to the width \( (\tau_{\text{width}}) \) of the SET with respect to the \( f_{\text{s}} \) as shown in (8).

\[
P(f_{\text{s}})_{\text{SET→SEU}} \propto \exists_{\text{DFF}} \left( \sum_{i=1}^{\#\text{CombinatorialCells}} (P_{gen(i)} P_{prop(i)} \tau_{\text{width}(i)}.f_{\text{s}}) \right) \tag{8}
\]

5) Putting it all together: DFF and Combinatorial Logic Upsets

As previously mentioned, upper-bound data path susceptibility \( (P(f_{\text{s}})_{\text{functionalLogic}}) \) is based on cone of logic Start-Point DFF capture \( (P(f_{\text{s}})_{\text{DFFSEU→SEU}}) \) and combinatorial logic gate capture \( (P(f_{\text{s}})_{\text{SET→SEU}}) \) as shown in (9).

\[
P(f_{\text{s}})_{\text{functionalLogic}} \propto \exists_{\text{DFF}} \left( \sum_{j=1}^{\#\text{StartPoint DFFs}} P_{\text{DFFSEU}(j)}(1 - \tau_{\text{dly}(j)}.f_{\text{s}}) + \sum_{i=1}^{\#\text{CombinatorialCells}} P_{gen(i)} P_{prop(i)} \tau_{\text{width}(i)}.f_{\text{s}} \right) \tag{9}
\]

C. Single Event Functional Interrupt \( (P_{\text{SEFI}}) \)

A Single Event Functional Interrupt (SEFI) is a SEU that forces the FPGA to be inoperable. According to the REAG-FPGA-SEU model, \( P_{\text{SEFI}} \) has two major categories:

1) Global Route SEFI: \( P_{\text{GlobalRoutes}} \)

As previously mentioned, all DFFs must be connected to a clock. In addition, all DFFs should be connected to a reset. Clock and reset signals are categorized as global routes because they are connected to a large number of components. The global routes are constructed as trees. Subsequently, if an SEU occurs, there are portions of the tree that can simultaneously affect multiple gates (global event) or individual gates (local event). Depending on the mitigation technique, local and/or global tree events can be masked, hence decreasing \( P_{\text{SEFI}} \).

2) Hidden Logic SEFI: \( P_{\text{HiddenLogic}} \)

Some FPGA devices have additional logic that is inaccessible to the designer. Hidden logic is used for a variety of operations depending on the manufacturer. As an example, the ProASIC3 and RTAXS contain Joint Test Action Group (JTAG) circuitry [8][9]. If the circuitry were to incur a SEU, it is possible for the FPGA’s I/O to become inoperable and hence cause catastrophic responses, i.e. a SEFI. This SEFI has been observed during radiation testing. However, if the JTAG clock and reset inputs are properly grounded during operation, it has been shown that JTAG I/O SEFIs are insignificant [6]-[9].

3) ProASIC3 and RTAXS \( P_{\text{SEFI}} \) Equation

Regarding the ProASIC3 and RTAXS FPGA devices, the hidden logic contribution to \( P_{\text{SEFI}} \) is near zero. Hence, global routes are the significant contributor to \( P_{\text{SEFI}} \). \( P_{\text{SEFI}} \) can be approximated as the sum of global and hidden logic contributions.

\[
P_{\text{SEFI}} \propto P_{\text{GlobalRoutes}} + P_{\text{HiddenLogic}} \tag{10}
\]

III. ANALYSIS OF MODEL COMPONENTS

It is intuitive to expect that a non-mitigated design will have
a significantly higher $\sigma_{\text{SEU}}$ than a mitigated design. It is not necessarily intuitive to determine the strength of the mitigation or the dominant source of SEUs. However, component significance can be determined using Table 3 and $\sigma_{\text{SEU}}$ data.

### Table 3: Analysis of SEU Capture Effects: $P(f_s)_{\text{DFFSEU-SEU}}$ versus $P(f_s)_{\text{SET-SEU}}$

<table>
<thead>
<tr>
<th>Logic</th>
<th>$P(f_s)_{\text{DFFSEU-SEU}}$</th>
<th>$P(f_s)_{\text{SET-SEU}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capture percentage of clock period</td>
<td>1 - $\tau_{dly} f_s$</td>
<td>As frequency increases, $P(f_s)_{\text{DFFSEU-SEU}}$ increases</td>
</tr>
<tr>
<td>Frequency Dependency</td>
<td>Increase in frequency decreases $P(f_s)_{\text{DFFSEU-SEU}}$</td>
<td>Increase in frequency increases $P(f_s)_{\text{SET-SEU}}$</td>
</tr>
<tr>
<td>Combinatorial Logic Effect</td>
<td>Increase in Combinatorial logic increases $\tau_{dly}$ and decreases $P(f_s)_{\text{DFFSEU-SEU}}$</td>
<td>Increase in Combinatorial logic increases $P(f_s)_{\text{SET-SEU}}$</td>
</tr>
</tbody>
</table>

Based on Table 3, the following is a list of trends used for evaluating $\sigma_{\text{SEU}}$ data and determining dominant sources of susceptibility:

- **$P(f_s)_{\text{DFFSEU-SEU}}$ Dominance – Most SEUs stem from captured Start-Point DFFs.** This is true when:
  - There is an increase in the number of combinatorial logic blocks or $\tau_{dly}$ and the $\sigma_{\text{SEU}} (P(f_s)_{\text{error}})$ decreases in response.
  - There is an increase in frequency and the $\sigma_{\text{SEU}}$ decreases in response.

- **$P(f_s)_{\text{SET-SEU}}$ Dominance – Most SEUs stem from captured Combinatorial Logic SETs.** This is true when:
  - There is an increase in frequency and $\sigma_{\text{SEU}}$ increases in response.
  - There is an increase in combinatorial logic and $\sigma_{\text{SEU}}$ increases in response.

Local Mitigation Strength: if the design has been mitigated using a localized-DFF mitigation scheme such as Localized Triple Modular Redundancy (LTMR)[1] or Dual Inter Cell (DICE)[6]:

- It is expected that the DFFs are masked from $\sigma_{\text{SEU}}$ contribution. $P(f_s)_{\text{DFFSEU-SEU}}$ should be insignificant, and subsequently $\sigma_{\text{SEU}}$ is lower.
- However, if $P(f_s)_{\text{DFFSEU-SEU}}$ has the most significant error contribution for a localized-DFF mitigation scheme, then the mitigation scheme is considered weak because it is not fully masking DFF upsets.

### IV. Reducing System Error: Triple Modular Redundancy Schemes

For the ProASIC3 and RTAXS, as previously mentioned, $P(f_s)_{\text{configuration}}$ is near zero. Substituting $P(f_s)_{\text{DFFSEU-SEU}}$ and $P(f_s)_{\text{SET-SEU}}$ in (2) for $P(f_s)_{\text{functional Logic}}$, a non-mitigated ProASIC3 or RTAXS design is expected to have a $\sigma_{\text{SEU}}$ cross as reflected in (11).

\[
\text{No – Mitigation } \sigma_{\text{SEU}}: \quad P(f_s)_{\text{error}} \propto P(f_s)_{\text{DFFSEU-SEU}} + P(f_s)_{\text{SET-SEU}} + P_{\text{SEFI}} \tag{11}
\]

In order to reduce $\sigma_{\text{SEU}}$, mitigation is applied to the FPGA design. A common form of mitigation is Triple Modular Redundancy (TMR). TMR is a scheme such that a group of circuitry is triplicated and then voted. The mitigation is a majority voter: i.e., best-two-out-of-three. It is important to differentiate and signify the TMR scheme based on which circuits are redundant so that the user is aware of the strength of the mitigation strategy. The following is a discussion of one of the TMR schemes used during radiation testing: Localized TMR (LTMR).

#### A. Localized TMR (LTMR)

![Localized TMR (LTMR)](image)

Fig. 3: Localized Triple Modular Redundancy (LTMR). DFFs are triplicated and a voter is inserted into the data path.

LTMR is the process of triplicating each DFF of a design and inserting a voter after each DFF triplication [1][12]. The LTMR process is illustrated in Fig. 3. A limitation of LTMR is that shared data paths exist as inputs to the triplicated group of circuitry is triplicated and then voted. The mitigation is a majority voter: i.e., best-two-out-of-three. It is important to differentiate and signify the TMR scheme based on which circuits are redundant so that the user is aware of the strength of the mitigation strategy. The following is a discussion of one of the TMR schemes used during radiation testing: Localized TMR (LTMR).

#### B. LTMR and the REAG-FPGA-SEU Model

As a synopsis of the mitigation power of ProASIC3 and RTAXS LTMR, DFFs ($P(f_s)_{\text{DFFSEU-SEU}}$) are mitigated, but data paths ($P(f_s)_{\text{SET-SEU}}$) are not. It follows that (11) is reduced to (12) with LTMR insertion.

\[
\text{LTMR } \sigma_{\text{SEU}}: \quad P(f_s)_{\text{error}} \propto P(f_s)_{\text{SET-SEU}} + P_{\text{SEFI}} \tag{12}
\]

### V. ProASIC3 and RTAXS SEE Test Structures

The Device-Under-Test (DUT) test structures followed the NASA REAG FPGA testing methodology [2][6][7] implementing Windowed Shift Registers (WSR) strings and
Counter Arrays. Only data pertaining to WSR chains are presented in this manuscript.

A WSR is a shift register with a different output scheme as illustrated in Fig. 4. Instead of outputting the last DFF once every clock cycle, a WSR outputs the last 4 DFFs once every 4 clock cycles. The parallel output has proven successful for high speed transmission [2].

The following is WSR Nomenclature:

- \( \text{WSR}_{0} \): N=0 Chain … Only DFFs
- \( \text{WSR}_{8} \): N=8 Chain… 8 Inverters per 1 DFF
- \( \text{WSR}_{16} \): N=16 Chain… 16 Inverters per 1 DFF

The mitigated ProASIC3 designs used the automated synthesis tool: Mentor Precision-RTL [12] to implement LTMR. RTAXS contains embedded LTMR, therefore, no additional mitigation was inserted into RTAXS designs.

VI. HEAVY ION SEU TESTING

Heavy-Ion testing has been performed at Texas A&M Cyclotron Facility using the NASA REAG Low Cost Digital Testing (LCDT) System [6][7].

A. SEU Cross Section Calculation

While the ProASIC3 is exposed to an active heavy-ion beam, designs are operating and outputs are compared to expected values for each system clock cycle. If an output is not equivalent to its expected state, then an upset is recorded. \( \sigma_{\text{SEU}} \)'s are based on the number of observed upsets normalized by the active beam particle fluence. Depending on the evaluation, an additional normalization step may be implemented to enhance analysis.

B. WSR Chains

Each WSR chain (e.g. N=0, N=8, and N=16) has a unique SEU cross section (\( \sigma_{\text{WSR}_{N}} \)) and is normalized by the number of DFFs (bits) contained in the string. Equation (13) shows \( \sigma_{\text{WSR}_{N}} \):

\[
\sigma_{\text{WSR}_{N}} = \frac{\# \text{ WSR upsets}}{\# \text{Particles} \times \# \text{WSR bit}} \quad \text{cm}^2 \quad \text{bit}^{-1}
\]  

C. Global Routes

Because global routes are connected to multiple DFF cells, one upset can affect a significant number of DFFs. Subsequently, SEU events that occur on global routes as global events are not normalized by bit. SEU cross sections are measured by device. Equation (14) shows \( \sigma_{\text{SEFI}} \). It is important to note that \( \sigma_{\text{SEFI}} \) only reflects global events on global route structures. Global route local events are currently being analyzed and are not included in (14).

\[
\sigma_{\text{SEFI}} = \sigma_{\text{Global Route}} = \frac{\# \text{ Global Events}}{\# \text{Particles} \times \text{cm}^2} \quad \text{device}^{-1}
\]

D. SEU Cross Section Analysis

After SEU cross sections are calculated, comparisons are performed to their expected models and across designs. WSRs are evaluated to determine:

- \( P(f_s)_{\text{SEU}} \) versus \( P(f_s)_{\text{SET}} \) dominance: i.e., which elements mostly contribute to the overall \( \sigma_{\text{SEU}} \): DFFs or combinatorial logic? The results are used to develop lower SEU susceptible FPGA designs.
- Frequency dependency: Is there a strong \( P(f_s)_{\text{SET}} \) component? If frequency dependence is found to be significant, frequency based \( \sigma_{\text{SEU}} \) data should be used as input to error rate calculations.
- Other SEU Model effects and trends as previously described in Section III.

VII. HEAVY ION TEST RESULTS AND ANALYSIS

A. ProASIC- No-TMR Analysis

One would expect that WSRs \( \sigma_{\text{SEU}} \) (\( \sigma_{\text{WSR}_{N}} \)) will always be greater than WSRs \( \sigma_{\text{SEU}} \) (\( \sigma_{\text{WSR}_{0}} \)) because WSR chains have more logic. However, \( \sigma_{\text{SEU}} \) data reveals that this is not always a valid assumption. Fig. 6 illustrates that for ProASIC3-No-TMR WSRs, \( \sigma_{\text{WSR}_{0}} \geq \sigma_{\text{WSR}_{N}} \) across all LETs.
Fig. 6: ProASIC3 WSR₀ and WSR₈ with No-TMR and user-inserted LTMR. For No-TMR WSR₀ has higher SEU Cross section than WSR₈. With LTMR the trend is switched, WSR₈ has a higher SEU cross section than WSR₀.

Why are No-TMR-ProASIC3 $\sigma_{WSR₀\_SEU} > \sigma_{WSR₈\_SEU}$ for every LET? Consider $\tau_{dy}$. With No-TMR, the DFFs are not mitigated. Hence $P(f\_SET\_→\_SEU) > 0$ and there is a $\tau_{dy}/f$ dependence. It is known that:

$$\tau_{dy\_WSR₀} < \tau_{dy\_WSR₈} \quad \text{and} \quad \sigma_{SEU} \propto (1-\tau_{dy}/f) \quad \text{(as shown in (7))}$$

Hence, due to the inverse correlation between $\sigma_{SEU}$ and $\tau_{dy}$, it follows that No-TMR: $\sigma_{WSR₀\_SEU} > \sigma_{WSR₈\_SEU}$. This can be further observed using the REAG-FPGA-SEU Model and $\sigma_{SEU}$ data. Equation (15) reflects the $\sigma_{SEU}$ heavy ion data in Fig. 6 and the fact that $\sigma_{WSR₀\_SEU} > \sigma_{WSR₈\_SEU}$.

$$\left(P\left(f\_SET\_\rightarrow\_SEU\right) + P\left(f\_DFF\_\rightarrow\_SEU\right)\right)_{WSR₀} > \left(P\left(f\_SET\_\rightarrow\_SEU\right) + P\left(f\_DFF\_\rightarrow\_SEU\right)\right)_{WSR₈} \quad \text{(15)}$$

Substitutions are made for $P(f\_DFF\_\rightarrow\_SEU)$ and $P(f\_SET\_\rightarrow\_SEU)$ in (16) to form (17):

$$P\left(f\_DFF\_\rightarrow\_SEU\right) \left(1 - \frac{\tau_{dy\_WSR₀}}{\tau_{ck}}\right) > \quad \text{(17)}$$

Equation (17) reveals the $\tau_{dy}$ significance with respect to the $\sigma_{SEU}$. In addition, rearranging (17) leads to (18) and shows that DFFs are more SEU susceptible than combinatorial logic.

$$P\left(f\_DFF\_\rightarrow\_SEU\right) > \frac{\tau_{ck}}{\tau_{dy\_WSR₈} - \tau_{dy\_WSR₀}} \sum_{i=1}^{8} P\left(f\_SET\_\rightarrow\_SEU\right) \quad \text{(18)}$$

A more detailed inspection of relative $\sigma_{SEU}$’s for the ProASIC3-No-TMR WSR₀ and WSR₈ is illustrated in Fig. 7. It can be seen that as LET increases, the ratio of WSR₀ to WSR₈ slightly decreases. This can be explained using (16) or (17). As LET increases, SETs increase in significance. Consequently, the $P(f\_SET\_\rightarrow\_SEU)$ component becomes more significant and subsequently reduces the relative difference between $\sigma_{WSR₀\_SEU}$ and $\sigma_{WSR₈\_SEU}$.

B. ProASIC3-LTMR-WSRs: $P(f\_SET\_\rightarrow\_SEU)$

Fig. 8: ProASIC3-LTMR-WSR₀ and ProASIC3-LTMR-WSR₈. As the frequency increases or the number of combinatorial blocks increases, the $\sigma_{SEU}$ increases.

Fig. 6 illustrates that with user-inserted LTMR, the overall $\sigma_{SEU}$ is reduced and now $\sigma_{WSR₀\_SEU} < \sigma_{WSR₈\_SEU}$. This trend follows conventional theory: as the number of combinatorial logic blocks increases so does the SEU cross section.

Regarding the REAG-FPGA-SEU model, it is expected that $\sigma_{WSR₀\_SEU} < \sigma_{WSR₈\_SEU}$ because $P(f\_DFF\_\rightarrow\_SEU)$ is mitigated with LTMR. Consequently, with LTMR insertion, $P(f\_SET\_\rightarrow\_SEU)$ is now the significant component. In addition, the $\sigma_{SEU}$ data in Fig. 6 and Fig. 8 show the dominance of $P(f\_SET\_\rightarrow\_SEU)$ for a LTMR design. Given the $\sigma_{SEU}$ data, the
dominance of $P(\text{fs})_{\text{SET} \rightarrow \text{SEU}}$ and the effects of (8), the following hold true for ProASIC3-LTMR designs:

- As the number of combinatorial logic gates increases, $P(\text{fs})_{\text{SET} \rightarrow \text{SEU}}$ increases and hence $\sigma_{\text{SEU}}$ increases; i.e., LTMR $\sigma_{\text{WSR}0_{\text{SEU}}} < \sigma_{\text{WSR}8_{\text{SEU}}}$ as illustrated in Fig. 6
- As frequency increases, $\sigma_{\text{WSR}N_{\text{SEU}}}$ also increases, as illustrated in Fig. 8

C. ProASIC3 versus RTAXS Analysis

1) RTAXS Embedded LTMR versus LTMR-ProASIC3

Fig. 9 is a comparison between RTAXS WSRs (containing embedded LTMR) with the ProASIC3 WSRs (containing user inserted LTMR). It is shown that although the RTAXS has an overall lower $\sigma_{\text{SEU}}$, the LTMR-ProASIC3 $\sigma_{\text{SEU}}$ are not drastically higher. In addition, the data shows that the LET threshold (LET$_{\text{th}}$) for the LTMR-ProASIC3 is statistically similar to the RTAXS.

ProASIC3 $\sigma_{\text{WSR}N_{\text{SEU}}}$ are higher than RTAXS $\sigma_{\text{WSR}N_{\text{SEU}}}$ for two major reasons:

1. The ProASIC3 is a commercial grade part containing gates with switching rates considerably higher than the RTAXS[4][6]. In addition, the routing network of the ProASIC3 has less capacitive loading than the RTAXS as fan-out and length increases. By definition, faster switching rates and less capacitance lead to a higher SET $P_{\text{prop}}$ than slower circuits that contain significant capacitive loading.

2. The RTAXS embedded mitigation scheme uses a wired–or as a voter [3][4]. The wired–or does not contribute to the $\sigma_{\text{SEU}}$ because it does not use transistors to perform the voting. However, the ProASIC3 voters utilize a number of transistors to perform the “best-two-out-of-three function and hence have a significant contribution to the overall $\sigma_{\text{SEU}}$. Fig. 6 illustrates the difference between RTAXS and ProASIC3 mitigation schemes.

Fig. 10: RTAXS embedded LTMR[4] versus ProASIC3 user-inserted LTMR

2) $\sigma_{\text{SEU}}$ reduction with an increase in combinatorial logic: ProASIC3 versus RTAXS

As previously mentioned, one would expect that $\sigma_{\text{WSR}0_{\text{SEU}}} < \sigma_{\text{WSR}8_{\text{SEU}}}$ because WSR$_8$ chains have more logic (i.e. WSR$_8$ contains more combinatorial logic between DFF stages than WSR$_0$). However, depending on the mitigation strategy, we have shown that this is not always the case for the ProASIC3. In support, Fig. 6 illustrates that across all LET values the ProASIC3-No-TMR $\sigma_{\text{WSR}0_{\text{SEU}}} > \sigma_{\text{WSR}8_{\text{SEU}}}$. This trend is due to the dominance of $P(\text{fs})_{\text{DFFSEU} \rightarrow \text{SEU}}$ and $\tau_{\text{dly}}$ in a non-mitigated design.

By inserting LTMR, the data show that the trend reverses; i.e., for ProASIC3-LTMR: $\sigma_{\text{WSR}0_{\text{SEU}}} < \sigma_{\text{WSR}8_{\text{SEU}}}$ across all tested LET values due to the mitigation of $P(\text{fs})_{\text{DFFSEU} \rightarrow \text{SEU}}$. Hence, as the number of combinatorial logic cells increases in a ProASIC3-LTMR design, so does the $\sigma_{\text{SEU}}$.

Regarding the SEU response to increasing combinatorial logic in the RTAXS, it has also been observed that an increase in combinatorial logic at LET <10 MeV*cm$^2$/mg can reduce $\sigma_{\text{SEU}}$ ($\sigma_{\text{WSR}0_{\text{SEU}}} < \sigma_{\text{WSR}8_{\text{SEU}}}$) due to attenuation of SETs [2][6]. However, for LET >10 MeV*cm$^2$/mg, the LTMR properties become dominant and the trend is reversed: ($\sigma_{\text{WSR}0_{\text{SEU}}} < \sigma_{\text{WSR}8_{\text{SEU}}}$).

To reiterate, it is unconventional to expect that as the number of combinatorial logic gates increases, the $\sigma_{\text{SEU}}$ decreases ($\sigma_{\text{WSR}0_{\text{SEU}}} > \sigma_{\text{WSR}8_{\text{SEU}}}$). Although the RTAXS and the No-TMR-ProASIC3 both have trends where $\sigma_{\text{WSR}0_{\text{SEU}}} > \sigma_{\text{WSR}8_{\text{SEU}}}$, the conditions and rationales for the unexpected SEU response are completely different. Table 4 summarizes the conditions and factors that influence the $\sigma_{\text{WSR}0_{\text{SEU}}} > \sigma_{\text{WSR}8_{\text{SEU}}}$ response for ProASIC3-No-TMR WSRs versus RTAXS WSRs.
The refined FPGA SEU models have proven to reliably reflect the irradiation, SEU test results were analyzed and applied to the end-point DFF. Post-radiation, SEU test results were analyzed and applied to the original SEU probability models to develop more precise top-down models. The refined FPGA SEU models have proven to reliably reflect the \( \sigma_{SEU} \) data, mitigation strategy, and synchronous design component effects (DFFs and combinatorial logic).

Regarding heavy-ion data, ProASIC3-LTMR has proven to improve SEU performance with respect to ProASIC3-No-TMR designs by increasing the LET of 8.6 MeV·cm²/mg and reducing the overall \( \sigma_{SEU} \).

When comparing the LTMR ProASIC3 to the RTAXs SEU data, it has been shown that the ProASIC3-LTMR LET is compatible with the RTAXS LET. However, the overall ProASIC3-LTMR cross sections are higher than the RTAXS cross sections.

Unconventional \( \sigma_{SEU} \) reduction was observed as the number of combinatorial logic blocks were increased for both devices (No-TMR ProASIC3 and LTMR RTAXS). However, it has been shown that the cause for the \( \sigma_{SEU} \) reduction in both devices and when it occurs are due to completely different conditions. The REAG FPGA model proved to be a useful tool for validating \( \sigma_{SEU} \) reduction and for providing trend rationale.

Due to the level of detail required, \( \sigma_{SEU} \) is still being evaluated and will be presented in the future once analysis is completed.

FPGA SEU models have been incorporated into the SEU testing methodology developed by NASA REAG. The methodology covers test preparation, test execution, and extensive data analysis. The approach has proven to be a successful, technology-independent means to facilitate device evaluation and comparison studies.

### VIII. Conclusion

The NASA REAG FPGA SEU testing methodology has been applied to Actel RTAXS and ProASIC3 FPGA devices. Because the ProASIC3 does not include embedded mitigation, mitigation strategies have been inserted into ProASIC3 test structures. Each design with and without mitigation has been evaluated to determine the effectiveness of the various mitigation strategies. Only LTMR data is presented in this manuscript.

During the development and test phases, high level REAG-FPGA-SEU models assisted with DUT design creation and were used as points of reference during testing. Post-radiation, SEU test results were analyzed and applied to the original SEU probability models to develop more precise top-down models. The refined FPGA SEU models have proven to reliably reflect the \( \sigma_{SEU} \) data, mitigation strategy, and synchronous design component effects (DFFs and combinatorial logic).

### References

[1] M. Berg “Trading Application Specific Integrated Circuit (ASIC) and Field Programmable Gate Array (FPGA) Considerations for System Insertion”, NSREC Short Course, Quebec City, CN, July 2009


