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

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

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)

\[ \sigma_{seu} = \frac{\#errors}{fluence} \]

\( \sigma \) (cm²/bit) vs. LET (MeV·cm²/mg)
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
(SET)s can develop in combinatorial logic

SETs can vary in pulse width ($T_{\text{pulse}}$) and amplitude.

\[
f_c = \frac{1}{2\pi RC}
\]

Each capacitance has its own $f_c$

\[
Q_{\text{crit}} = C_{\text{node}} V_{\text{node}}
\]

\[
Q_{\text{coll}} > Q_{\text{crit}}
\]
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

**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

Glitch = Transient

\[ P_{SET} \]

Metal layers not susceptible

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

\[ P(f_s)_{SET\rightarrow SEU} \]

\[ t_p = \frac{1}{f_s} \]

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

\[
P(f_s)_{\text{SET} \rightarrow \text{SEU}} \propto \frac{T(f_s)_{\text{pulse}} \times P(f_s)_{\text{SETgen}} \times P(f_s)_{\text{SETprop}} \times P_{\text{DFFEn}}}{2 \times \frac{1}{f_s}}
\]
Summary: Most Significant Factors of System Error Probability $P(f_s)_{\text{error}}$

- **Configuration**
  - SRAM Based FPGAs
  - $P_{\text{Configuration}}$

- **DFFs**
  - STATIC SEU
  - Dynamic SET $\rightarrow$ SEU
  - $P_{DFFSEU}$
  - $P(f_s)_{\text{SET} \rightarrow \text{SEU}}$

- **SEFIs**
  - Clocks & Resets
  - Inaccessible control circuitry
  - $P_{\text{SEFI}}$

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

Combinatorial logic: C-CELL

Sequential logic R-CELL

Super Cluster

Combinatorial logic C-CELL

Sequential logic R-CELL
General Xilinx Virtex 4 FPGA Architecture: SRAM Based Configuration

Configuration Logic Block: CLB

Lookup Table (LUT)
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:

\[
P(f_s)_{error} \propto P_{Configuration} + P_{DFFSEU} + P(f_s)_{SET\rightarrow SEU} + P_{SEFI}\]
Example: TMR Mitigation Schemes will use Majority Voting

\[
\text{Majority Voter} = I_1 \land I_2 + I_0 \land I_2 + I_0 \land I_1
\]

<table>
<thead>
<tr>
<th>I₀</th>
<th>I₁</th>
<th>I₂</th>
<th>Majority Voter</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

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)
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

$$P(\text{fs})_{\text{error}} \propto P_{\text{OFFSEU}} + P(\text{fs})_{\text{SET\rightarrow SEU}} + P_{\text{SEFI}}$$
RTAX Example: Probability of Error Reduction

**LTMR**

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

\[
P(\delta_s)_{\text{error}} \propto P_{\text{Configuration}} + P_{\text{DEFSEU}} + P(\delta_s)_{\text{SET} \rightarrow \text{SEU}} + P_{\text{SEFI}}
\]

\[\approx 0\]
Example…Upper-Bound Error Prediction for LTMR + hardened Global Routes RHBD

\[ P(f_s)_{error} \propto P(f_s)_{SET \rightarrow SEU} \]

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

\[ P(f_s)_{SET \rightarrow SEU} : \]

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

Source: NASA Goddard
Upper-Bound Error Prediction Actel RHBD Anti-fuse FPGA

With embedded LTMR Mitigation + Hardened Clocks:

\[ P(f_s)_{\text{error}} \propto P(f_s)_{\text{SET} \rightarrow \text{SEU}} \]

\[
\frac{dE}{dt} < \frac{dE_{\text{bit}}(f_s)}{dt} \ast (\# \text{UsedDFFs}) < 6 \times 10^{-8} \left( \frac{\text{Errors}}{\text{bit} - \text{day}} \right) \ast 5 \times 10^4 \left( \frac{\text{bits}}{\text{design}} \right)
\]

\[
\frac{dE}{dt} < 3 \times 10^{-3} \left( \frac{\text{Errors}}{\text{design} - \text{day}} \right) \text{Years}
\]

50,000 DFFs
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 $\rightarrow$ Complexity

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

$P(f_s)_{\text{error}} \propto P_{\text{Configuration}} + P_{\text{SET\rightarrow SEU}} + P_{\text{SEFI}}$
GTMR Proves To be A Great Mitigation Strategy… BUT…

- Tripling 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
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

<table>
<thead>
<tr>
<th>FPGA</th>
<th>LTMR</th>
<th>DTMR</th>
<th>GTMR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antifuse</td>
<td>Green</td>
<td>Red</td>
<td>Red</td>
</tr>
<tr>
<td>SRAM</td>
<td>Red</td>
<td>Red</td>
<td>Green</td>
</tr>
<tr>
<td>Flash</td>
<td>Green</td>
<td>Red</td>
<td>Red</td>
</tr>
</tbody>
</table>

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