Reliability Improvement in Reconfigurable FPGAs

B. Chagun Basha

Jeudis de la Comm
22 May 2014
Overview

FPGA Fabrics

- Logic Resources
  - Combinational
  - Sequential

- Configuration Bitstream
  - SRAM

Logic Resources

- Combinational
- Sequential

Configuration Bitstream

- SRAM
Single Event Effects (SEE) occur when a high-energy particle passes through the active region of a semiconductor, triggering non-destructive effects such as upset, multiple-bit upset, or analog transients; or destructive effects such as latch-up, gate rupture, and burnout.
Designing with Dependability

Single Event Effect (SEE)

Recoverable Effects (Soft)
- Single Event Transient (SET)
- Single Event Upset (SEU)
  - Single Bit Upset (SBU)
  - Multi Bit Upset (MBU)
- Single Event Latch-ups (SEL)

Non-Recoverable Effects (Hard)
- Single Event Burn-out (SEB)
- Single Event Gate Rapture (SEGR)
- Single Event Functional Interrupt (SEFI)
Designing with Dependability

Logic Resources

- Mux/LuT
- D-FF

Configuration Bitstream

- SRAM

- Combinational as well as Sequential functional Implementation.

![Diagram of Mux/LuT and D-FF](image)
Reliability Improvement

Configuration Bitstream Protection

Single Event Upset (SEU)
  - Single Bit Upset (SBU)
  - Multi Bit Upset (MBU)

Single Event Upset (SEU) Solution

Built-in 3D-Hamming Multiple Error Correcting Scheme.

Logic Resources Protection

Transistor Fault Model:
- Stuck faults (1, 0, close, open)
- Bridging between signal lines (i/p, o/p)
- Shorts in Source-Drain (SD), Gate-Source (GS), & Gate-Drain (GD)

Logic Resources Protection Solution

This is what the presentation talks about.
Built-in 3D-Hamming EC Scheme

- Built-in protection
- Less parity memory overhead
- Improved Multiple bit error correcting capability.

**Reference**: Chagun Basha, Sébastien Pillement and Stainslaw Piestrak, ‘Built-in 3-Dimensional Hamming Multiple-Error Correcting Scheme to Mitigate Radiation Effects in SRAM-Based FPGAs’, ARC’14 April 2014, Portugal.

**Optimization of Parity Memory Overhead**

- **Built-in protection**
- **Less parity memory overhead**
- **Improved Multiple bit error correcting capability.**
Logic Resource Protection

- Passive Redundancy Techniques
  - Fault Masking

- Active Redundancy Techniques
  - Detection, Localization, Recovery

- Hybrid Redundancy Techniques
  - Fault Masking/Detection + Reconfiguration (Dy-Pr)

- Duplication with Comparison (DWC)
  - Pair and a spare
  - Watch dog – Lock step
  - Standby sparing
  - Concurrent detection

Are they focus really on Logic Fault Models?
A circuit is **totally self-checking** if it is both **fault secure** and **self-testing**.

- Totally self-checking circuits are very desirable for highly reliable system design.
- Such circuits have significant advantages, such as:

  1. Transient faults as well as permanent faults are detected.
  2. Faults are immediately detected upon occurrence; this prevents propagation of corrupt data within the system.

Self Checking Circuits deals with **“Logic Faults”**
Block Diagram of Proposed Self-Checking Logic Block

Two-Rail Checker Circuit

Self-Checking Multiplexer

D-FF Pair

Error Evaluator

Logic Comparator

Error_Comb

Data_Comb

Error_Seq

Data_Seq

- Belongs to CLE

- Belongs to EDCA
Basic Building Block

- Error Detection and Correction Analysis
- Configuration Bitstream for User Logic
- Configuration Bitstream for Routing Resources
- (Re) Configurable Logic Element
- Basic Block
- EDCA
- CLE
In [4], a new Two-Rail Checker (TRC) circuit is presented, to detect the presence of logic faults such as bridging and stuck-on.

Inspired by the TRC presented in [4], a new Fault Tolerant Multiplexer is proposed in [5] & [6].

1. Two-Rail Checker

Fig 2 shows how to check a 4 input multiplexer by adding two inverters (M17M18 and M19M20) to its output(out).

2. Self-Checking Multiplexer design [5]
Proposed Self-Checking Multiplexerer

Different Threshold Voltage
Inv (Q17/Q18) – V_{T_1}
Inv (Q19/Q20) – V_{T_2}

<table>
<thead>
<tr>
<th>E1</th>
<th>E2</th>
<th>Err_comb</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Fault Free</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Fault</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Fault</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Fault Free</td>
</tr>
</tbody>
</table>
ARDyT© FPGA Architecture

EDCA – Error Detection and Correction Analysis.
CLE – Configurable Logic Element.

*H.F.D.C – Hierarchical Fault Detection Chain
Hardware Overhead Comparison

<table>
<thead>
<tr>
<th>Logic Cell</th>
<th>Proposed Scheme (With Unused Internal FF)</th>
<th>Dual Modulo Redundancy (DMR)</th>
<th>Triple Modulo Redundancy (TMR)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-Input</td>
<td>186 %</td>
<td>226 %</td>
<td>356 %</td>
</tr>
<tr>
<td>8-Input</td>
<td>150 %</td>
<td>215 %</td>
<td>332 %</td>
</tr>
</tbody>
</table>

Table 2: Hardware overhead comparison of fault tolerant logic cell architectures
References

1. Study of an architecture for fault tolerant Field Programmable Gate Arrays


5. S. Pontarelli, G.C. Cardarilli, A. Leandri, M Ottavi, M Re. A. Salsano, A Self-checking Cell Logic Block for Fault Tolerant FPGAs, Department of Electronic Engineering University of Rome “Tor Vergata”, Italy, 2002.
