Recovering from Transient Failures in FPGA Applications

Robert Wei-Je Huang
Stanford CRC
May 1, 2000

Outline
- Introduction
- FPGA Models
- Recovery Schemes
- Summary

Introduction
- Advantage of FPGAs
  - Efficient Use of Hardware
  - Avoid Faulty Parts by Reconfiguration
- Fault Tolerance Issues in FPGAs
  - Concurrent Error Detection (CED)
  - Diagnosis
  - Repair
  - Distinguish Transient from Permanent Faults

FPGA Architecture Model
- Configurable Logic Block (CLB)
  - Look-Up Table (LUT)
  - MUX
  - Flip-Flops and Latches
- Interconnect
  - Connection Box (CB)
  - Switch Box (SB)
- I/O Block (IOB)
- On-Chip RAM Block

CLB and Interconnect Array
Recovering Transient Failures on FPGA Applications

Top-level Architecture

- Example: Xilinx Virtex Series [Xilinx 99]

<table>
<thead>
<tr>
<th>DLL</th>
<th>CLB and Interconnect Array</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left IOBs</td>
<td>Right IOBs</td>
</tr>
<tr>
<td>Left Block SelectRAM</td>
<td>Right Block SelectRAM</td>
</tr>
<tr>
<td>DLL</td>
<td>Bottom IOBs</td>
</tr>
</tbody>
</table>

Configuring FPGA

- Configuration Interface
  - 8-bit Parallel SelectMAP Port
  - Boundary Scan (JTAG) Port
- Configuration Frames

Configuration Bits for Virtex

- 0.5 Million to 6 Million Bits
  - Up to 3 Million Gates
- CLB Columns (0.4 Million to 5.2 Million)
  - 18 * (# CLB_rows + 2) Bits Per Frame
- LUT and SelectRAM (6 %)
- Flip-Flops & Latches (0.4 %)
- Interconnect (93.6 %)
- Block RAMs (0.1 Million to 0.9 Million)
  - User Configurable Memory (70 %)
  - Interconnect (30 %)

Configuration Time for Virtex

- Interface
  - 8-bit SelectMAP Port
  - 50 MHz Clock Rate
- Full Configuration
  - 1.25 ms to 20 ms
- Single CLB Configuration Frame
  - 0.8 µs to 3 µs

Transient Fault Models

- Configuration Bit Flips
  - SEU Caused by Radioactive Particles
  - Up to 1 Upset Per Hour on Low Earth Orbital Path Applications [Carmichael 99]
  - Equivalent to a Change in Functionality
- Other Temporary Failures
  - Transient Stuck-At Faults on Signal Wires

Outline

- Introduction
- FPGA Models
- Recovery Schemes
  - Retry
  - Configuration Data Recovery
- Summary
Retry - Where to Restart?
• From the Beginning of the Task
  • Long Latency
  • Difficult for Stream-based Applications
• From the Previous PassedCheckpoint
  • Need to Rebuild Machine States
  • Need Scan Chains in FPGA Applications
  • Storage Overhead
    ◆ Dependent on CED Schemes

FSM for Retry Control
• Start from the Previous Checkpoint

Example: LZ Compressor [Huang 00]

Retry Example: LZ Compressor
• Machine States
  • Dictionary Entries
    ◆ Previous Input Source Data
    ◆ No Need For an Extra Scan Chain
  • Storage Overhead
    • \((N + 2 \cdot L_{\text{max}})\) Entries
    • 16% CLB Overhead for \(N = 512\), \(L_{\text{max}} = 63\)
  • Length Counter
    ◆ Simply Reset to 0

Weakness of Retry
• Cannot Recover Configuration Bit Flips in FPGAs
  • Soft Errors, Not Permanent
  • Waste if Disable the Faulty CLB
• Solution
  • Configuration Data Readback
Recovering Transient Failures on FPGA Applications

Configuration Data Recovery

- Architecture

```
Reconfigurable
FPGA 1
uController
Memory
EPROM
```
```
Reconfigurable
FPGA 2
uController
Memory
EPROM
```

Issues

- Long Latency for Complete Scrubbing
- Not Used as a Stand-Alone Scheme
- Readback Not Affecting FPGA Operation
- Extra Storage of Configuration Bits
- Use ECC for Each Frame
- Memory Coherence Problem [Xilinx 99]

```
Before Readback
LUT R2C3.S1 04
RAM R3C3.S1 01
C3

Before Writeback
LUT R2C3.S1 04
RAM R3C3.S1 14
C3

After Writeback
```

Memory Coherence Strategies

- Method 1

```
Normal
Stall
Writeback
```

```
Col: Column Under Check
WR: Memory Write Operation
Normal operation stalled in shaded states
```

Memory Coherence Strategies

- Method 2: Use Dirty Flag
  - FSM Corresponding to Each Frame

```
Readback
Normal
Dirty
Writeback
```

Overhead Comparison

- Assume Random Access Model
- Assume 1 Faulty Frame Present
- Parameters
  - $rd$ = No. of Cycles for Readback and EC
  - $wb$ = No. of Cycles for Writeback
  - $col$ = No. of Columns in FPGA
  - $mw$ = Memory Writes Per Cycle
  - $pr$ = Prob. of Columns with RAM
- Percentage of Stalls for Normal Operations ??

```
Method 1: 36 CLB columns, readout = writeback = 30 cycles
```

```
Method 1: 36 CLB columns, readout = writeback = 30 cycles
```

Overhead Comparison

- For Different Prob. of Columns with RAM

```
Method 1: 36 CLB columns, readout = writeback = 30 cycles
```

```
Method 1: 36 CLB columns, readout = writeback = 30 cycles
```

```
Method 1: 36 CLB columns, readout = writeback = 30 cycles
```

```
Method 1: 36 CLB columns, readout = writeback = 30 cycles
```
Overhead Comparison

For Different Prob. of Columns with RAM

For Different FPGA Sizes

For Different Readback / Writeback Latencies

Other Possible Schemes

- Duplicated Memory Modules
  - Separate Columns
  - Disable Unsynchronized Module
  - Synchronize When System Idle
  - Only Stall When Writeback Needed
  - Optimal Efficiency
  - Good for Multiple Faulty Frames in 1 Column
  - Bad for Multiple Faults in Different Columns
Summary

- FPGA Transient Fault Models
- Retry
  - Transient Faults w/o Affecting Config. Data
- Need Scan Chain to Rebuild States
- Configuration Data Recovery
  - Not a Stand-alone Scheme
  - Memory Coherence Issue
  - Dirty Flag Strategy

Reference