Modeling, Detection, Correction and Recovery Techniques for Unified Dependable Design

Hiroto Yasuura
Kyushu University
Analysis of Dependability in VLSI Design

- Analysis techniques of dependability for hierarchical VLSI system design are required.
- Error in the lower level should be modeled in different signals in the upper level.
- In most conventional researches, error models are supposed independently and dependability is analyzed in each level.
- We developed a tool chain, which can analyze dependability from the results in the lower level for soft error of digital VLSI.
  - Techniques for transfer error model from analyzed results in the lower level
  - Reduction algorithms of the computation time of analysis against the rapid increase of the number of transistors in the upper levels
- We also developed other technologies for dependable VLSI design
  - Resilient design by Canary FFs
  - Trade-off between design for testability and security
Tool Chain for Soft Error Tolerance

- System Design (>10^8 Transistors)
- RTL Design (~10^8 Tr.)
- Logic Design (~10^6 Tr.)
- Circuit Design (~10^2 Tr.)
- Layout Design
- Atomic Level
- System Level Simulator
- Cache Simulator
- NBTI Mitigation
- Logic/RTL Analyzer
- Memory Management
- Canary FF Replacement
- Logic Synthesis
- Device Simulator
- SRIM
- PHITS (JAPAN Atomic Energy Agency)
- Fageeha

Consistent with design hierarchy of digital VLSI
Error Models and Analysis Tools

Analysis Tool for Logic Cells

Error Probabilities of Combinational Cells

Error Probabilities of Flip-Flop Cells

Analysis Tool for Combinational Circuits

Continuous Time

Logic/RTL Analysis

Error Probabilities of Flip-Flops

Discrete Time

Probabilistic Analysis Tool

Analysis Tool by Fault Simulation

Error Probability

Analysis of combinational circuits is separated from sequential circuit analysis.

2 approaches for RT level analysis
Analysis Tool for Logic Cells

Naïve tool chain

Environmental conditions

Device model

Transport calculation of secondary ion induced by neutron collision

Calculation of charge distribution model

Pulse $V(t)$ calculation with mixed-mode device/circuit simulator

Relation between pulse generation probability and pulse width

Problem

Device simulation must be iterated for a lot of times to achieve highly accurate estimation
\[ \Rightarrow \text{impractically long run-time} \]

Proposed Method

Approximation of noise electric current model using the amount of collected charge

Pulse $V(t)$ calculation with circuit simulator

A few seconds

The proposed method makes several thousands of samples to be analyzable in just a day!

Dependable VLSI International Symposium (Dec. 1, 2012)  Team Yasuura

5
Analysis Tool for Logic/RTL Circuits

Problem
- Naïve method requires more than 1 week computation for a circuit with 100,000 gates
- Several Tens of thousands of samples are needed to achieve highly accurate estimation

Proposed Method
- Timing-aware fault simulation that ignores logical masking
- Logic-aware fault simulation that considers all the gates at a time
- Fault sequential simulation considering multiple faults
- Computation in several hours for a circuit with 100,000 gates, guaranteeing the upper bound of error rate

SER : Soft Error Rate
(SEU : Single Event Upset
SET : Single Event Transient)

Team Yasuura
Timing-Aware Fault Simulation

- Analysis of propagation of a pulse caused at a gate to flip flops and/or primary output terminals considering its pulse width and timing
- Calculate the upper bound of the pulse width and estimate probability of which the pulse is latched onto the flip flops
  - Consider the variation of pulse width on the propagation paths
  - Ignore effects of logical masking and re-convergence

Calculation of pulse propagation

Input patterns \((\alpha, \beta, \gamma)\)

<table>
<thead>
<tr>
<th>Input patterns ((\alpha, \beta, \gamma))</th>
<th>Outputs</th>
<th>Normal value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0,0,0)</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>(0,0,1)</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>(0,1,0)</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>(0,1,1)</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>(1,0,0)</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>(1,0,1)</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>(1,1,0)</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>(1,1,1)</td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

Pulse (time span in which the value is different from normal value)

Calculation of pulse propagation \((*,*,*)\)
Computation Time of Accelerated Fault Simulation

The length of inputs

Computation time (seconds)

Circuits/ # of Gates

- b19/191,775
- b18/ 95,029
- b17/ 49,489
- b22/ 39,679
- b21/ 27,246
- b20/ 26,754
- b15/ 11,419

(5 hours)
Memory System Simulator

**Analysis Flow**

- Soft Error Rate of Memory Cells
- Program
- Input for Program
- HW Architecture

Calculate Temporal and Spatial Utilization of Memory Cells

Error Rate of Inputs of CPU (Instructions and Data)

**Evaluation**

Inputs of CPU

- Soft errors counted on if(a,1)
- Soft errors counted on if(a,2)
- Soft errors counted on if(a,3)
- Soft errors which do not affect the computer system

Not Inputs of CPU

**Results**

Vulnerability vs. Cache size [kB]

- L1 cache (I)
- L1 cache (D)
- Main mem (I)
- Main mem (D)
- Runtime

Dependable VLSI International Symposium (Dec. 1, 2012)
Example: Highly Dependable Design of Embedded Systems Using a Scratch Pad Memory

- No error correction in Cache Memory
- Error correction circuits in Scratch Pad Memory (SPM)

Improve dependability by allocation of program and data in SPM
## Trade-off Analysis between SPM Size and Vulnerability (Inverse of Error Rate)

<table>
<thead>
<tr>
<th>SPM size [KB]</th>
<th>Normalized vulnerability</th>
<th>Normalized execution time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.2</td>
<td>1.2</td>
</tr>
<tr>
<td>16384</td>
<td>0.8</td>
<td>1.0</td>
</tr>
<tr>
<td>32768</td>
<td>0.6</td>
<td>1.2</td>
</tr>
<tr>
<td>49152</td>
<td>0.4</td>
<td>1.4</td>
</tr>
<tr>
<td>65536</td>
<td>0.2</td>
<td>1.6</td>
</tr>
</tbody>
</table>

### Vulnerability (cache size: 16 KB)

### Execution time (cache size: 16 KB)

- **Execution program**
  - bi
  - ba
  - su
  - qs
  - st
  - ff
  - Mean

---

Dependable VLSI International Symposium (Dec. 1, 2012)  
Team Yasuura
Summary

1. Concept of Design Flow and Tool Chain for Dependable VLSI
   - Unified tool chain from physical phenomena to system level for estimation of dependability compatible with hierarchical design flow
   - Three practical problems: Soft Error, Timing Error and Security

2. Soft Error Caused by Neutron
   - Analysis tools for logic cells, combinational circuits and sequential circuits (logic and register transfer levels)
   - Memory architecture simulator and trade-off analysis

3. Timing Errors by Fluctuation of Manufacturing
   - Introduction of Canary Flip Flops for reducing design margin
   - Improvement of static noise margin of SRAM

4. Design for Security
   - Trade-off analysis between testability and security