Parallel Fault Backtracing for Calculation of Fault Coverage

R. Ubar, S. Devadze, J. Raik and A. Jutman

Tallinn University of Technology
Department of Computer Engineering
ESTONIA
Outline

- Introduction and Motivation
- Description of the proposed method
- Experimental results
- Conclusions
Motivation

- Fault simulation that used to build fault coverage table can take huge amount of time

- Fault simulation is widely used in digital circuit design flow:
  - Built-in Self Test
  - Fault diagnosis
  - Test pattern generation
  - ...

- Acceleration of fault simulation will speed-up all abovementioned tasks
Introduction: previous work

  Approximate fault simulation using critical path tracing

  Reducing number of fanout stems should be processed for fault simulation

  Dominator gate concept, early cut-off of fault evaluation

- F. Maamari, J. Rajski, “A Method of Fault Simulation Based on Stem Regions”, 1990, CAD
  Stem regions and exit lines, reduces fault simulation area

  Exact, linear-time critical path tracing
Introduction

Proposed fault analysis method:

- Uses single stuck-at fault model
- Intended for use with combinational circuits
- Works on higher abstraction level than gate-level
- Describes circuit using special class of BDDs (SSBDD)
- Based on Critical Path Tracing technique

Parallel computations for $N$ patterns ($N$ – width of computer word)
Extends critical path tracing beyond Fan-out Free Regions (FFRs)
Uses calculation of parallel Boolean derivatives
Circuit representation

Fan-out stems

SSBDD1

SSBDD2

SSBDD3

SSBDD4

SSBDD5

FFR

FFR

FFR

FFR

FFR

Y1

Y2

A

B

C

D

a

b

c

a

Y
Critical Path Tracing inside FFR

Fan-out free region (FFR)

Boolean derivative: $\frac{\partial y}{\partial x_1}$

If $\frac{\partial y}{\partial x_1} = 1$ – fault at $x_1$ is detected at output $y$

Using of special Structurally Synthesized Binary Decision Diagrams (SSBDD) we can rapidly calculate parallel Boolean derivatives (critical path tracing on SSBDD)
Extending Critical Path Tracing

Two consecutive Fan-out Free Regions

Sensitivity of $y$ to fault at $z_1$:

$$\frac{\partial y}{\partial z_1} = \left( \frac{\partial y}{\partial x_1} \right) \land \left( \frac{\partial x_1}{\partial z_1} \right)$$
Extending Critical Path Tracing (2)

Reconvergent fan-out

\[ y = F(x_1, \ldots, x_i, x_j, \ldots, x_n) \]
\[ x_1 = f_1(x, X_1) \]
\[ \ldots \]
\[ x_i = f_i(x, X_i) \]

\[ \frac{\partial y}{\partial x} = y \oplus F(x_1 \oplus (\frac{\partial x_1}{\partial x}), \ldots, (x_i \oplus (\frac{\partial x_i}{\partial x})), x_j, \ldots, x) \]

Where: \( \frac{\partial x_1}{\partial x} \) and \( \frac{\partial x_i}{\partial x} \) are Boolean derivatives were calculating during critical path tracing inside fan-out free regions \( f_1 \) and \( f_2 \)
Extending Critical Path Tracing (3)

Nested reconvergencies

\[ y = F_y(x, z, X_y) \]
\[ z = F_z(x, X_z) \]

\[ \frac{\partial y}{\partial x} = y \oplus F_y(x_1 \oplus (\frac{\partial x_1}{\partial x}), z \oplus (\frac{\partial F_z}{\partial x}, X_y) \]
Fault simulation algorithm (steps)

- **Topological pre-analysis**
  Constructs reconvergency and calculation models of the circuit

- **Parallel simulation**
  Calculates the values of all variables for given set of patterns

- **Fault simulation**
  Performs fault backtracking on the created calculation model
Topological model

Primary outputs:
A, B, C, D, E

Fan-out stems:
1, 2, 3, 4, 5, G

Internal fan-in gates:
H, G
Creating formulas / calculation steps

Simulation from node 2:

<table>
<thead>
<tr>
<th>Step</th>
<th>Action / Whole formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>3_1</td>
</tr>
<tr>
<td>2H</td>
<td>H_1</td>
</tr>
<tr>
<td>24</td>
<td>H_1 \land 4_1</td>
</tr>
<tr>
<td>2G</td>
<td>F_G(H_1 \land 4_1, 3_1)</td>
</tr>
<tr>
<td>2B</td>
<td>F_B(3_1, F_G)</td>
</tr>
<tr>
<td>2A</td>
<td>3_1 \land A_1</td>
</tr>
</tbody>
</table>

.......  

2A \lor 2B \lor 2C \lor 2D
Calculation model

Simulation from node 2:

<table>
<thead>
<tr>
<th>Step</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>2A:</td>
<td>$3_1 \land A_1$</td>
</tr>
<tr>
<td>2B:</td>
<td>$F_B(3_1, F_G)$</td>
</tr>
<tr>
<td>2G:</td>
<td>$F_G(H_1 \land 4_1, 3_1)$</td>
</tr>
<tr>
<td>24:</td>
<td>$H_1 \land 4_1$</td>
</tr>
<tr>
<td>2H:</td>
<td>$H_1$</td>
</tr>
<tr>
<td>23:</td>
<td>$3_1$</td>
</tr>
</tbody>
</table>

Calculated during Critical Path Tracing inside FFRs

Computed using calculation of parallel Boolean derivatives by formulas
Experimental results

- ISCAS’85/ISCAS’89 combinational benchmarks

Comparison with:

- State-of-the-art commercial tools from CAD vendors
- Exact Critical Path Tracing implementation by:
  

- Older version of the same algorithm (w/o topology optimization)

- Fault dropping mode was disabled

- 10000 patterns were simulated for each circuit
Scalability of the algorithm

![Graph showing scalability of the algorithm with Tool C1 and Proposed algorithm]

- **Y-axis:** time, s
- **X-axis:** circuits
- **Legend:**
  - Tool C1
  - Proposed algorithm
Conclusions / Future Work

- New fault simulation algorithm is proposed
- Simulation is performed for network of macros (instead of gates) with gate-level accuracy
- Macros are represented by Structurally Synthesized BDDs
- Topological analysis is used to speed-up simulation
- The speed of fault simulation outperforms several commercial tools

- Further optimization still possible
- Solution for fault dropping
Thank You for Your Attention!