Test and Diagnosis Pattern Generation for Dynamic Bridging Faults and Transition Delay Faults

Cheng-Hung Wu, Saint James Lee and Kuen-Jong Lee
National Cheng Kung University
Tainan, Taiwan

ASPDAC 2016
Outline

• Introduction
• Background
  – Dynamic Bridging Faults (DBFs)
  – Dominance
  – DBF Detection
• Diagnosis Pattern Generation
  – Inverse Dynamic Bridging Faults (IDBFs)
  – Fault Pair Classification
  – Test and Diagnosis Pattern Generation Flow
• Experimental Results
• Conclusions
Introduction

- Accurate diagnosis needs to identify both the location and the type of fault.
- Both the transition delay fault (TDF) and the dynamic bridging fault (DBF) have similar transition delay faulty effect.

- We need a diagnosis method to distinguish these two similar fault models.
Dynamic Bridging Faults

- Each DBF involves two wire nodes; one whose delay is affected is called the **victim** and the other is called the **aggressor** of the DBF.

- There are two types of DBFs including bridge slow-to-rise (bsr) and bridge slow-to-fall (bsf).
Fault Pairs

- Due to similar faulty effect, a fault pair is defined as a pair composed of a DBF and the corresponding TDF which located on the DBF’s victim.

![Diagram of fault pairs](image)

- TDF Slow-to-Rise (0-1) transition delay occurs
- DBF Slow-to-Rise (0-1)
- Transition delay occurs
- TDF? DBF?
Dominance

• (1) Active a transition on target wire and propagate to some observable outputs.
  (2) Set a constraint on the other target wire.

To test DBF need to satisfy both (1) and (2), but to test TDF only need to satisfy (1), therefore $TDF \rightarrow DBF$.

• In order to distinguish a DBF from its corresponding TDF, a pattern must detect the TDF without detecting the DBF.
DBF Detection

- To definitely detect a DBF, the aggressor must be set to a constant value that is opposite to the final value of the victim during the application of the test pattern.

Condition 1
Definitely detected

Condition 2
Probably detected

Condition 3
Definitely not detected
Summary of Fault Behaviors

1. Conventional tests targeting TDFs may not detect some DBFs.

2. If a test for a TDF also detects a DBF dominated by the TDF, the test cannot distinguish the two faults.

3. The only way to distinguish a DBF and its corresponding TDF is to inactivate the DBF while detecting the TDF.

4. To detect the TDF without detecting the DBF, the aggressor of DBF must hold the opposite value of victim’s first-cycle value.
Outline

• Introduction
• Background
  – Dynamic Bridging Faults (DBFs)
  – Dominance
  – DBF Detection
• Diagnosis Pattern Generation
  – Inverse Dynamic Bridging Faults (IDBFs)
  – Fault Pair Classification
  – Test and Diagnosis Pattern Generation Flow
• Experimental Results
• Conclusions
Inverse Dynamic Bridging Faults (IDBFs)

- Our method uses the inverter to set the opposite constant value on the aggressor by building a new DBF with a new aggressor.

- The DBF would not be detected when the corresponding IDBF is detected.
Diagnosis Method

1. Insert the inverter pairs for all wires at once, which does not affect original circuit’s function.

2. This method targets all inverse DBF at a time in the ATPG stage so as to achieve dynamic pattern compaction.
Fault Pairs Classification

- **Group 1:**
  \[ \text{DBF} = \text{Undetectable} \]  
  (Distinguished)

- **Group 2:**
  \[ \text{DBF} = \text{Detected or Aborted} \]  
  \[ \text{IDBF} = \text{Detected} \]  
  (Distinguished)

- **Group 3:**
  \[ \text{DBF} = \text{Detected} \]  
  \[ \text{IDBF} = \text{Undetectable} \]  
  (Equivalent)

- **Group 4:**
  the remaining cases  
  (Unidentified)

\[
\text{DR\%} = \frac{\#\text{Group 1} + \#\text{Group 2} + \#\text{Group 3}}{\#\text{All fault pairs under consideration}}
\]
Test and Diagnosis Flow - Overview

Part 1
- Run TDF ATPG
- Generate DBFs from testable TDFs and form fault pairs
- Modify circuit and get IDBFs
- Run ATPG for all IDBFs
- Run Fault Simulation for all TDFs using TP_TDF and DP
- Use TP_TDF
- Use TP_TDBF

Part 2
- Run ATPG for all IDBF_UD
- Classify all fault pairs into 4 groups by ATPG results
- Run Fault Simulation for all DBFs using TP_TDF and DP
- Run ATPG for all DBF_UD
- TP_TDF + DP + New_TP_TDBF

Part 3
- Run ATPG for all IDBF_UD
- Classify all fault pairs into 4 groups by ATPG results
- Run Fault Simulation for all DBFs using TP_TDF and DP
- Run ATPG for all DBF_UD
- TP_TDF + DP + New_TP_TDBF

Part 4
- Run ATPG for all IDBF_UD
- Classify all fault pairs into 4 groups by ATPG results
- Run Fault Simulation for all TDFs using TP_TDBF and DP
- Run ATPG for all TDF_UD
- TP_TDBF + DP + New_TP_TDBF
Due to the dominance relation, we can drop all undetectable TDFs.

- Build a DBF list based on detectable TDFs.

- Build fault pairs according to the DBF list and IDBF list.

- Form a IDBF for each fault pair.

### Flow Diagram

1. **Run TDF ATPG**
   - TP_{TDF}

2. **Generate DBFs from testable TDFs and form fault pairs**
   - DBF List

3. **Modify circuit and get IDBFs**
   - IDBF List

4. **Run ATPG for all DBFs**
   - TP_{DBF}
The test patterns for IDBF fault simulation can use either $TP_{TDF}$ or $TP_{DBF}$.

- The selection between $TP_{TDF}$ and $TP_{DBF}$ will cause the difference of total number of test patterns and diagnosis patterns since the sequence of ATPG processes are different.

- Distinguish fault pairs by the test patterns using IDBF’s fault simulation process.
The patterns to detect IDBF<sub>UD</sub> are diagnosis patterns (DP).

Classify all fault pairs into 4 groups, thus the diagnosis resolution is obtained.

Generate test patterns for the rest of TDFs or DBFs so as to achieve both TDF testing and DBF testing.
Test and Diagnosis Flow - Part 4

- The difference between part 3 and part 4 is the execution sequence of ATPG.

- TDF → IDBF → DBF on part 3
  DBF → IDBF → TDF on part 4
Outline

• Introduction

• Background
  – Dynamic Bridging Faults (DBFs)
  – Dominance
  – DBF Detection

• Diagnosis Pattern Generation
  – Inverse Dynamic Bridging Faults (IDBFs)
  – Fault Pair Classification
  – Test and Diagnosis Pattern Generation Flow

• Experimental Results

• Conclusions
Definition of Distance

We define the distance between an aggressor and a victim of a fault pair as the number of gates on the shortest path from the aggressor to the victim with both forward paths and backward directions considered.

\[
\begin{align*}
\text{D} & = 1 \\
\text{D} & = 2 \\
\text{D} & = 2 \\
\text{D} & = 3 \\
\text{D} & = 4
\end{align*}
\]
## Experimental Results

### Results of Fault Lists Generated by Distance ≤ 4.

<table>
<thead>
<tr>
<th>CKT</th>
<th>#TDF</th>
<th>Distance</th>
<th>#FP</th>
<th>Number (Percentage) of each Group</th>
<th>DR%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Group 1</td>
<td>Group 2</td>
</tr>
<tr>
<td>≤ 1</td>
<td>24922</td>
<td></td>
<td></td>
<td>2501</td>
<td>13326</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(10.04%)</td>
<td>(53.47%)</td>
</tr>
<tr>
<td>≤ 2</td>
<td>13228</td>
<td></td>
<td></td>
<td>19290</td>
<td>90968</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(14.59%)</td>
<td>(68.80%)</td>
</tr>
<tr>
<td>≤ 3</td>
<td>41297</td>
<td></td>
<td></td>
<td>40148</td>
<td>327297</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(9.72%)</td>
<td>(79.25%)</td>
</tr>
<tr>
<td>≤ 4</td>
<td>100472</td>
<td></td>
<td></td>
<td>75459</td>
<td>844773</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(7.51%)</td>
<td>(84.08%)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CKT</th>
<th>#TDF</th>
<th>Distance</th>
<th>#FP</th>
<th>Number (Percentage) of each Group</th>
<th>DR%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Group 1</td>
<td>Group 2</td>
</tr>
<tr>
<td>≤ 1</td>
<td>27426</td>
<td></td>
<td></td>
<td>384</td>
<td>18154</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(1.40%)</td>
<td>(66.19%)</td>
</tr>
<tr>
<td>≤ 2</td>
<td>1458620</td>
<td></td>
<td></td>
<td>49658</td>
<td>1337149</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(3.40%)</td>
<td>(91.67%)</td>
</tr>
<tr>
<td>≤ 3</td>
<td>4553456</td>
<td></td>
<td></td>
<td>79310</td>
<td>4255628</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(1.74%)</td>
<td>(93.46%)</td>
</tr>
<tr>
<td>≤ 4</td>
<td>14965716</td>
<td></td>
<td></td>
<td>43166</td>
<td>14141303</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(2.89%)</td>
<td>(94.49%)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CKT</th>
<th>#TDF</th>
<th>Distance</th>
<th>#FP</th>
<th>Number (Percentage) of each Group</th>
<th>DR%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Group 1</td>
<td>Group 2</td>
</tr>
<tr>
<td>≤ 1</td>
<td>127543</td>
<td></td>
<td></td>
<td>7192</td>
<td>83829</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(5.64%)</td>
<td>(65.73%)</td>
</tr>
<tr>
<td>≤ 2</td>
<td>708823</td>
<td></td>
<td></td>
<td>64976</td>
<td>568857</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(9.17%)</td>
<td>(80.25%)</td>
</tr>
<tr>
<td>≤ 3</td>
<td>2285025</td>
<td></td>
<td></td>
<td>120570</td>
<td>2021045</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(5.28%)</td>
<td>(88.45%)</td>
</tr>
<tr>
<td>≤ 4</td>
<td>5298061</td>
<td></td>
<td></td>
<td>179665</td>
<td>4917819</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(3.39%)</td>
<td>(92.82%)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CKT</th>
<th>#TDF</th>
<th>Distance</th>
<th>#FP</th>
<th>Number (Percentage) of each Group</th>
<th>DR%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Group 1</td>
<td>Group 2</td>
</tr>
<tr>
<td>≤ 1</td>
<td>116192</td>
<td></td>
<td></td>
<td>8717</td>
<td>62085</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(7.50%)</td>
<td>(53.43%)</td>
</tr>
<tr>
<td>≤ 2</td>
<td>567966</td>
<td></td>
<td></td>
<td>99059</td>
<td>3616013</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(17.44%)</td>
<td>(63.67%)</td>
</tr>
<tr>
<td>≤ 3</td>
<td>2346913</td>
<td></td>
<td></td>
<td>239687</td>
<td>1792810</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(10.21%)</td>
<td>(76.39%)</td>
</tr>
<tr>
<td>≤ 4</td>
<td>8771968</td>
<td></td>
<td></td>
<td>826337</td>
<td>7212141</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(9.42%)</td>
<td>(82.22%)</td>
</tr>
</tbody>
</table>

Average of ≤ 4  | 5.80% | 88.40% | 5.75% | 0.04% | 99.96% |

• Our diagnosis method can be applied on **88.40%** of fault pairs.

• **99.96%** of fault pairs can be identified as distinguished or equivalent.

- Group 1 = Distinguished
- Group 2 = Distinguished
- Group 3 = Equivalent
- Group 4 = Unidentified

---

NCKUEE

VLSI TEST LAB
## Experimental Results

**Numbers of Test Patterns and Diagnosis Patterns for Fault Lists Determined by Distance.**

<table>
<thead>
<tr>
<th>CKT</th>
<th>#TDF</th>
<th>Distance</th>
<th>#FP</th>
<th>DATPG after using TP&lt;sub&gt;TDF&lt;/sub&gt;</th>
<th>DATPG after using TP&lt;sub&gt;DBF&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>#TP&lt;sub&gt;TDF&lt;/sub&gt;</td>
<td>#D&lt;sub&gt;TDF&lt;/sub&gt;</td>
</tr>
<tr>
<td>s15850</td>
<td>16104</td>
<td>≤ 1</td>
<td>24922</td>
<td>1598</td>
<td>229</td>
</tr>
<tr>
<td></td>
<td></td>
<td>≤ 2</td>
<td>132228</td>
<td>8306</td>
<td>507</td>
</tr>
<tr>
<td></td>
<td></td>
<td>≤ 3</td>
<td>412973</td>
<td>30366</td>
<td>816</td>
</tr>
<tr>
<td></td>
<td></td>
<td>≤ 4</td>
<td>1004722</td>
<td>74691</td>
<td>1494</td>
</tr>
<tr>
<td>s35932</td>
<td>37530</td>
<td>≤ 1</td>
<td>27426</td>
<td>2063</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td></td>
<td>≤ 2</td>
<td>1458620</td>
<td>49559</td>
<td>64</td>
</tr>
<tr>
<td></td>
<td></td>
<td>≤ 3</td>
<td>4553456</td>
<td>262854</td>
<td>223</td>
</tr>
<tr>
<td></td>
<td></td>
<td>≤ 4</td>
<td>14965716</td>
<td>627867</td>
<td>255</td>
</tr>
<tr>
<td>s38417</td>
<td>43757</td>
<td>≤ 1</td>
<td>127543</td>
<td>9594</td>
<td>325</td>
</tr>
<tr>
<td></td>
<td></td>
<td>≤ 2</td>
<td>708823</td>
<td>54303</td>
<td>801</td>
</tr>
<tr>
<td></td>
<td></td>
<td>≤ 3</td>
<td>2285025</td>
<td>188071</td>
<td>1606</td>
</tr>
<tr>
<td></td>
<td></td>
<td>≤ 4</td>
<td>5298061</td>
<td>454406</td>
<td>2428</td>
</tr>
<tr>
<td>s38584</td>
<td>52007</td>
<td>≤ 1</td>
<td>116192</td>
<td>6050</td>
<td>377</td>
</tr>
<tr>
<td></td>
<td></td>
<td>≤ 2</td>
<td>567966</td>
<td>28480</td>
<td>919</td>
</tr>
<tr>
<td></td>
<td></td>
<td>≤ 3</td>
<td>2346913</td>
<td>145725</td>
<td>2342</td>
</tr>
<tr>
<td></td>
<td></td>
<td>≤ 4</td>
<td>8771968</td>
<td>381876</td>
<td>2426</td>
</tr>
</tbody>
</table>
Conclusions

• Based on the dominance relation between TDFs and DBFs, we propose a novel circuit modified method to generate very compact diagnosis patterns to distinguish TDFs and DBFs.

• The results show that very high diagnosis resolution can be achieved by our proposed flow. (99.96% for fault pairs where DBF’s aggressor and victim are near to each other, 99.99% for fault pairs that are randomly selected).