Test Cost Reduction

LG Electronics
Lee, Yong
• Introduction

• Key factors for test cost reduction in DFT
  – Test vector volume
  – Low cost ATE
  – Test time
  – Reuse a large block

• Test cost reduction using DFT methods
  – On chip clocking
  – Scan compression
  – Scan balancing using OCC
  – At-speed test using SDD (Small Delay Detect)
  – Power-Aware DFT
  – MBIST

• Future works
• **Un-modeled signal integrity faults**
  – Need for accurate at-speed test
  – High Cost ATE
  – Huge test patterns are required for un-modeled faults
Introduction

- Test Cost
  - Reduction the cost of wafer test & final test
  - To increase test coverage for large circuits
  - To facilitate multi-site testing
  - To use a low cost ATE
• Very large block reuse
  – Many case HDL codes are modified for the test
  – Scan stitching & MBIST are inserted in top level
  – Hierarchical DFT flow is required
Key factors for test cost reduction

- **Test pattern volume**
  - ATE memory capacitance
  - Increasing test time
  - Multi DUT test

- **Low cost ATE**
  - Test cost saving per sec.
  - Load board cost

- **Test time**
  - Test cost saving each device
  - A number of test vectors are required

- **Reuse a large block**
  - Time-to-market
Top block diagram within DFT
Scan compression

- Key benefits
  - 10-100x test time and test volume reduction to lower test costs
  - Same high test coverage and ease-of-use as standard scan
  - No impact on design timing
  - No impact on design physical implementation
  - Very low area impact
  - Tightly integrated with low-power design flows
  - Enables higher test quality for designs at 130-nm and below
Scan balancing Method

- Calculate the block power for scan stitching in shift mode
- Compose of scan compression logics for each blocks
- Connect PLL & OCC & DFT block
## Scan STIL Format

### Scan Shift Frequency: 25Mhz

<table>
<thead>
<tr>
<th>Timing</th>
<th>50Mhz_clk</th>
<th>20Mhz_clk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period ‘50ns’</td>
<td>P { ‘0ns’ D; ‘25ns’ U; ‘50ns’ ;D}}</td>
<td>P { ‘0ns’ D; ‘25ns’ U; ‘50ns’ ;D}}</td>
</tr>
<tr>
<td>Pattern 0</td>
<td>Chain1 {101010101...01010101010}</td>
<td>Chain2 {11011101...00000000000}</td>
</tr>
<tr>
<td>Pattern 1</td>
<td>Chain1 {001010100...01111101010}</td>
<td>Chain2 {111010100...01111101010}</td>
</tr>
</tbody>
</table>

### Scan Shift Frequency: 50Mhz

<table>
<thead>
<tr>
<th>Timing</th>
<th>50Mhz_clk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period ‘25ns’</td>
<td>P { ‘0ns’ D; ‘14ns’ U; ‘24ns’ ;D}}</td>
</tr>
<tr>
<td>25Mhz_clk</td>
<td>01XZ { ‘0ns’ }</td>
</tr>
<tr>
<td>Pattern 0</td>
<td>Chain1 {101010101...01010101010...01010101010}</td>
</tr>
<tr>
<td>Pattern 1</td>
<td>Chain1 {001010100...01111101010...01111101010...01111101010}</td>
</tr>
</tbody>
</table>

| Pattern 1 | Chain1 {001010100...01111101010...01111101010...01111101010} | Chain2 {11111001100110000...0011110000110011001100} |

| Pattern 1 | Chain1 {001010100...01111101010...01111101010...01111101010} | Chain2 {11111001100110000...0011110000110011001100} |

| Pattern 1 | Chain1 {001010100...01111101010...01111101010...01111101010} | Chain2 {11111001100110000...0011110000110011001100} |

| Pattern 1 | Chain1 {001010100...01111101010...01111101010...01111101010} | Chain2 {11111001100110000...0011110000110011001100} |

| Pattern 1 | Chain1 {001010100...01111101010...01111101010...01111101010} | Chain2 {11111001100110000...0011110000110011001100} |

| Pattern 1 | Chain1 {001010100...01111101010...01111101010...01111101010} | Chain2 {11111001100110000...0011110000110011001100} |

| Pattern 1 | Chain1 {001010100...01111101010...01111101010...01111101010} | Chain2 {11111001100110000...0011110000110011001100} |

| Pattern 1 | Chain1 {001010100...01111101010...01111101010...01111101010} | Chain2 {11111001100110000...0011110000110011001100} |

| Pattern 1 | Chain1 {001010100...01111101010...01111101010...01111101010} | Chain2 {11111001100110000...0011110000110011001100} |

| Pattern 1 | Chain1 {001010100...01111101010...01111101010...01111101010} | Chain2 {11111001100110000...0011110000110011001100} |

| Pattern 1 | Chain1 {001010100...01111101010...01111101010...01111101010} | Chain2 {11111001100110000...0011110000110011001100} |

| Pattern 1 | Chain1 {001010100...01111101010...01111101010...01111101010} | Chain2 {11111001100110000...0011110000110011001100} |

| Pattern 1 | Chain1 {001010100...01111101010...01111101010...01111101010} | Chain2 {11111001100110000...0011110000110011001100} |
### Scan compression results

<table>
<thead>
<tr>
<th>DUT</th>
<th>F/F (number)</th>
<th>scan chain</th>
<th>Gate count</th>
<th>Maximum Frequency / Test Clock Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>LGxxxx</td>
<td>268,925</td>
<td>16/160</td>
<td>17,143,112</td>
<td>400Mhz / 25Mhz</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Normal Scan</th>
<th>Scan compression</th>
<th>Scan balance Using scan compression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scan chain /scan chain number</td>
<td>16chains 15100 FF</td>
<td>160chains 1510 FF</td>
<td>80chains (50Mhz) : 2012 FF 80chains (25Mhz) : 1006 FF</td>
</tr>
<tr>
<td>Test pattern volume</td>
<td>5,000 patterns * 15,100 length = 75,500,000</td>
<td>5000 patterns * 1510 length = 7,550,000</td>
<td>5000 patterns * 2012 length = 10,060,000</td>
</tr>
<tr>
<td>Test coverage</td>
<td>Stuck-at fault : 99.4% Transition fault : 87.42%</td>
<td>Stuck-at fault : 99.4% Transition fault : 87.42%</td>
<td>Stuck-at fault : 99.4% Transition fault : 87.39%</td>
</tr>
<tr>
<td>Test time</td>
<td>3.41 sec</td>
<td>0.377 sec</td>
<td>0.252 sec</td>
</tr>
<tr>
<td>Hardware overhead</td>
<td>2.1%</td>
<td>2.4%</td>
<td>2.7%</td>
</tr>
</tbody>
</table>
At-speed Test using SDD

- Small Delay Defect (SDD) target faults and non-target faults
  - Uses slack-based test generation for defined SDD target faults
  - Uses regular transition fault test generation for all others in fault list
- Slack-aware tests detect small-delay defects
  - ATPG selects observation path with lowest slack
  - Slack data from timing sign-off tool
Power-aware DFT

- Power-aware DFT using AND gate
  - Logic driven by the functional output does not toggle during scan shift
  - Adds an AND gate at the functional output of a scan flop per user’s specification
- Benefit
  - Reduces power dissipation during shift
Controllers configured and assigned to memories based on several criteria / constraints including:

- Clock domains
- Physical clustering
- Test time
- Power draw
MBIST detail connection

Test time of Algorithms : $20N + 2 \times \text{retention time}$
(determined by retention time in the memory library file)

Fault model list : Stuck-at faults * Transition faults *
Unlinked dynamic coupling faults * Address decoder faults *
Read/Write logic faults * Parametric faults
( cycle time, write recovery time, data retention )
Destructive read faults * Single-port bitline coupling fault
## Fault Summary Report (Wafer test)

### Table

<table>
<thead>
<tr>
<th></th>
<th>O/S</th>
<th>P/S</th>
<th>IIL</th>
<th>IPU</th>
<th>IIL</th>
<th>IIP</th>
<th>MBIST</th>
<th>SCAN</th>
<th>SC_TR</th>
<th>SC_PATH</th>
<th>SC_BR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Per input</td>
<td>0.044%</td>
<td>0.270%</td>
<td>0.019%</td>
<td>0.000%</td>
<td>0.016%</td>
<td>0.262%</td>
<td>0.022%</td>
<td>0.840%</td>
<td>1.065%</td>
<td>0.222%</td>
<td>0.001%</td>
</tr>
<tr>
<td>Per fail</td>
<td>1.602%</td>
<td>9.766%</td>
<td>0.689%</td>
<td>0.000%</td>
<td>0.576%</td>
<td>9.496%</td>
<td>0.811%</td>
<td>30.427%</td>
<td>38.571%</td>
<td>8.031%</td>
<td>0.020%</td>
</tr>
</tbody>
</table>

* path delay pattern is covered 10ppm
* Bridging pattern is covered 5~6ppm
* TSMC 65nm LP process

### Diagram

- Total faults 90% (excluded AIP, DDR interface)
- Undetected fault: 10% (AIP, DDR interface)
- Stuck-at 35%
- MBIST 27.1%
  - Included PLL (using ETCreat in Logicvision)
- Transition 7.2%
- Path delay 0.02%
- Bridging 0.012%
- IIL, IPU, SIDD, AIDD, O/S, P/S 20.7%
- Pattern numbers: 4000 (using adaptive scan in synopsys)
Future Works

- **Analog / Mixed-signal / RF**
  - All-digital DFT methods
  - Structurally-based DFT methods
  - RF wrappers for SoCs

- **MPU / PE / DSP / Memory**
  - Delay BIST methods
  - Alternative digital BIST methods using IDDx, thermal test, etc.
  - System-wide digital BIST architectures

- **SoC / SIP solutions**
  - Test generation tools and integrated DFT methods
  - Hierarchical DFT methods
  - Simulation models and tools for predicting DFT impact on system performance
  - Statistical methods for coverage monitoring
  - DFT / ATE interface standard definitions