How To Well Control Power Consumption of BIST in Low Power Design Flow

TP Hsieh
Senior Technical Manager/iSTART
Outline

1. iSTART Corp. and Its’ Mission
2. iSTART BIST Tool Introduction
3. Memory Grouping Architecture
4. Professional Memory Testing Algorithm
5. Low Power Design Methodology of iSTART
6. Summary
1. iSTART Corp. and Its’ Mission
iSTART Corp.

- Founded in 2010
- A professional Built-In Self Testing (BIST) company
- Has 34 memory testing patents in US, Taiwan and China
- Exceed 30 customers adopt iSTART BIST solution
- Customers across Taiwan, China, Korea, Europe
- Customers involve IC design service, IC design company, Foundry ...etc.
- Products have been adopted in TCON, Fingerprint, Automotive, AI ...etc.
2. iSTART BIST Tool Introduction
Main Lines of Business

Products

START
SRAM Test/Repair Tool
Tool-based

HEART
SRAM Test/Repair IP
IP-based

Customized design

EFlash/MTP Test/Repair IP
IP-based
START Flow

- Design files
- BFL file
- BII file

START Flow Diagram:

1. Auto Identification
2. Auto Classification
3. Auto Clock Tracing
4. Auto Grouping
5. Auto Generation & Insertion
6. System Integrating

Branches:

- Inserted Design With BIST/BISR
- Synthesis/ Formal Scripts
- BIST/BISR Verilog RTL
- Faulty Memory
- Test Benches
- Integrated Design with BISR
- Test Bench

- Synthesis
  - BIST/ BISR Netlist
  - Simulation
  - Simulation
BFL GUI Interface

➢ Plan chip hierarchy arch.
➢ Set clock source
➢ Set memories grouping
➢ Specify faulty model for testing
➢ Specify testing algorithm and pattern
➢ Assign diagnosis output information
BII GUI Interface

◆ Integrate all MBIST/BISR controllers
◆ Set only one IEEE 1149.1 interface to communicate with ATE
◆ Define the ordering of MBIST/BISR controllers
◆ Define TAP hierarchy
◆ Hookup IEEE 1149.1 pins
3. Memory Grouping Architecture
Patented Architecture of BIST and BISR

**C = Controller**
The controller is in charge of function selection, test pattern selection and TRA control.

**S = Sequencer**
The Sequencer is in charge of hardware sharing and keep all of test patterns can be executed, ordering, sequencing address at the same time.

**T = Test Pattern Generator**
The test pattern generator is in charge of At-Speed execution, instruction execution, comparator.

**TRA = Testing Redundancy Analyzer**
The testing redundancy analyzer is in charge of spare redundancy memory control.
Hardware sharing architecture of BIST

◆ Benefit of hardware sharing

■ The proportion of MBIST area will drop rapidly when the number of memory is more than 4.

■ The proportion of MBIST does not increase with the number of memory.

<table>
<thead>
<tr>
<th>Mem</th>
<th>1 area</th>
<th>%</th>
<th>Mem 4 area</th>
<th>%</th>
<th>Mem 20 area</th>
<th>%</th>
<th>Mem 30 area</th>
<th>%</th>
<th>Mem 150 area</th>
<th>%</th>
<th>Mem 200 area</th>
<th>%</th>
<th>Mem 300 area</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controller</td>
<td>920</td>
<td>21.33</td>
<td>1008.72</td>
<td>7.57</td>
<td>1299</td>
<td>2.12</td>
<td>1443.96</td>
<td>1.58</td>
<td>3356</td>
<td>0.75</td>
<td>4241</td>
<td>0.71</td>
<td>5817</td>
<td>0.65</td>
</tr>
<tr>
<td>Sequencer</td>
<td>439</td>
<td>10.18</td>
<td>487</td>
<td>3.66</td>
<td>745.56</td>
<td>1.22</td>
<td>884</td>
<td>0.97</td>
<td>2554</td>
<td>0.57</td>
<td>3757.68</td>
<td>0.63</td>
<td>5109.12</td>
<td>0.57</td>
</tr>
<tr>
<td>TPG</td>
<td>2955</td>
<td>68.50</td>
<td>11827.88</td>
<td>88.77</td>
<td>59188.92</td>
<td>96.66</td>
<td>88850.04</td>
<td>97.45</td>
<td>444176</td>
<td>98.69</td>
<td>592245.2</td>
<td>98.67</td>
<td>888806.4</td>
<td>98.79</td>
</tr>
<tr>
<td>Total</td>
<td>4314</td>
<td>100%</td>
<td>13323.6</td>
<td>100%</td>
<td>61233.48</td>
<td>100%</td>
<td>91178</td>
<td>100%</td>
<td>450086</td>
<td>100%</td>
<td>600243.9</td>
<td>100%</td>
<td>899732.5</td>
<td>100%</td>
</tr>
<tr>
<td>SRAM area</td>
<td>70409</td>
<td>6.13%</td>
<td>281636</td>
<td>4.73%</td>
<td>1408180</td>
<td>4.35%</td>
<td>2112270</td>
<td>4.32%</td>
<td>10561350</td>
<td>4.26%</td>
<td>14081800</td>
<td>4.26%</td>
<td>21122700</td>
<td>4.26%</td>
</tr>
</tbody>
</table>
4. Professional Memory Testing Algorithm
## Suggested testing algorithm for Emerging Market

<table>
<thead>
<tr>
<th>Application</th>
<th>Technology node</th>
<th>Suggested algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>AI (Video)</td>
<td>40nm, 28nm, 16nm</td>
<td>March C+, March 17N, March 33N</td>
</tr>
<tr>
<td>AI (Audio)</td>
<td>55nm, 40nm, 28nm</td>
<td>March C+, March C, March 17N</td>
</tr>
<tr>
<td>AI (Blockchain)</td>
<td>28nm, 16nm</td>
<td>March C+, March 17N, March 33N</td>
</tr>
<tr>
<td>Security</td>
<td>55nm, 40nm, 28nm</td>
<td>MOVI, March C+, March C, Non-March BM*, March Mdsn1**</td>
</tr>
<tr>
<td>Wireless (WiFi or BLE)</td>
<td>55nm, 40nm, 28nm</td>
<td>MOVI, March C+, March C, Non-March BM*, March Mdsn1**</td>
</tr>
<tr>
<td>Audio</td>
<td>55nm, 40nm</td>
<td>MOVI, March C+, March C, Non-March BM*, March Mdsn1**</td>
</tr>
<tr>
<td>TCON</td>
<td>55nm, 40nm, 28nm</td>
<td>MOVI, March C+, March 17N</td>
</tr>
<tr>
<td>Fingerprint recognition</td>
<td>180nm, 130nm, 110nm, 55nm</td>
<td>MOVI, March C+, March C, Non-March BM*</td>
</tr>
</tbody>
</table>

*: Non-March BM is for testing embedded memory with write enable bit  
**: March Mdsn1 is for testing retention memory for low power SoC
## Suggested testing algorithm for Programming type

<table>
<thead>
<tr>
<th>Programming Type</th>
<th>Suggested algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Most of instructions are related to Load/Store</td>
<td>March C+, March 17N, March 33N</td>
</tr>
<tr>
<td>Most of instructions are related to Branch</td>
<td>MOVI, March C+, March C, March C-</td>
</tr>
<tr>
<td>Most of instructions are related to VLIW and SIMD</td>
<td>March C+, March 17N, March 33N</td>
</tr>
<tr>
<td>Most of instructions are related to ALU</td>
<td>MOVI, March C+, Non-March BM*, March Mdsn1**</td>
</tr>
</tbody>
</table>

* Non-March BM is for testing embedded memory with write enable bit
** March Mdsn1 is for testing retention memory for low power SoC
## Suggested testing algorithm for Automotive

<table>
<thead>
<tr>
<th>Suggested algorithm*</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>March CW</td>
<td>The March CW algorithm is a word-oriented test algorithm for embedded memory. It has highest fault coverage than other algorithms.</td>
</tr>
<tr>
<td>March 17N</td>
<td>The March 17N algorithm is a March-based RAM diagnosis algorithm which not only locates faulty cells but also identifies their types.</td>
</tr>
<tr>
<td>March 33N</td>
<td>The March 33N minimal March test algorithm is introduced for detection of all two-operation single-cell dynamic faults.</td>
</tr>
<tr>
<td>March SSSc</td>
<td>The March SSSc is one of the popular algorithms. It can detects continue read operation.</td>
</tr>
<tr>
<td>Non-March BM</td>
<td>The Non-March BM is used to detect bit/group write enable faults and datapath shorts.</td>
</tr>
</tbody>
</table>

*: The memory defect of Automotive related applications come from multi-port memory, continue write/ read operations and burst write/ read operation.
## Suggested testing algorithm for Automotive

<table>
<thead>
<tr>
<th>Suggested algorithm*</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>March Weak WL @2P</td>
<td>The March Weak WL @2P for two port memories in advance manufacturing technologies</td>
</tr>
<tr>
<td>March C+ @2P</td>
<td>The March C+ @2P algorithm detects normally memory fault in two ports memory</td>
</tr>
<tr>
<td>March A2PF-M</td>
<td>The March A2PF-M algorithm for dual port memories in advance manufacturing technologies</td>
</tr>
<tr>
<td>March C+ @DP</td>
<td>The March C+ @DP algorithm detects normally memory fault in dual port memory</td>
</tr>
</tbody>
</table>

*: The memory defect of Automotive related applications come from multi-port memory, continue write/ read operations and burst write/ read operation.
5. Low Power Design Methodology of iSTART
Generic BIST/BISR Hierarchy

- BIST/BISR controller hook up with top level
- It may waste power even BIST or BISR do not work
- To reduce power consumption, BIST/BISR need to be hooked up OFF domain but top level, but how to implement it?
BIST Hierarchy Specific Architecture

Top_module_name
Top_hierarchy
BIST Hierarchy Specific Architecture Case 1

Top_module_name = A

Top_hierarchy = A u B1 u C1

Diagram showing:
- Top_module_name: A
- Top_hierarchy: A u B1 u C1
- Blocks: INTEG, RP CTR, BISR, BIST CTR

Diagram layout:
- A
  - B
    - C
      - INTEG
      - RP CTR
      - BISR
      - BIST CTR
BIST Hierarchy Specific Architecture Case 2

B1
- BIST CTR
- INTEG
- RP CTR
- BISR

B2
- RP CTR
- BIST CTR

B3
- RP CTR

Top_module_name = B1
Top_hierarch = B1

Top_module_name = B2
Top_hierarch = B2

Top_module_name = B3
Top_hierarch = B3
6. Summary
What can iSTART do for you?

➢ Provides professional BIST/BISR tools
➢ User friendly BIST/BISR tools
➢ Provides cost-effective tools
➢ Apply all advance technology designs
➢ Increase chip yield rate
➢ Provides best solution for low power design
Thank You