BIST for FPGAs & CPLDs

- Overview of FPGAs & CPLDs
- Logic BIST Architectures
  - LUT-based FPGAs
  - PLA-based CPLDs
- Interconnect BIST Architectures
- Boundary Scan Access to BIST
  - User-Defined Scan Chain
  - Configuration Memory Readback
- On-Line BIST
- Partial Reconfiguration
- Embedded Processor Based BIST
- Benefits & Limitations
BIST for FPGAs

- FPGA:
  - an $N \times N$ array of identical programmable logic blocks
  - programmable interconnect network
  - programmable I/O cells
- SRAM-based FPGA:
  - programming = writing the configuration memory
- FPGA testing:
  - complex problem
    - must cover all modes of operation and failure modes
  - current manufacturing tests:
    - manually generated - create & exercise many application circuits
    - expensive fault simulations
    - device tests not applicable for board and system testing
BIST for FPGAs (cont)

- **Configure the FPGA to test itself** *(Abramovici & Stroud)*
  - FPGA’96, VTS’96, ITC’96, ITC’97, ITC’98, ITC’99, ITC’00

- Configure the FPGA logic resources as:
  - Programmable Logic Blocks under test (BUTs)
    - BUTs are repeatedly reconfigured for all modes of operations
  - TPGs – counters supply exhaustive test patterns to BUTs
  - Comparator-based ORAs - compare outputs of BUTs
    - All BUTs have identical configurations & receive the same patterns
    - No aliasing & easy to implement
  - **Swap TPG/ORA & BUT roles to ensure each PLB is tested**

- “Free lunch” - no area overhead or performance penalty
- Pseudo-exhaustive testing ensures high fault coverage
- Same tests applied at every level of testing (device to system)
- Can identify/locate defective programmable logic block(s)
BIST for FPGAs (cont)

- Independent of array size
- Complete test for all PLBs
- 2 test sessions to test all PLBs
- Advantages:
  - No area overhead
  - No performance penalty
  - Applicable at all testing levels
  - Function-independent
  - Complete logic test
  - Maximal diagnostic resolution
- Cost:
  - Download time for test sessions
  - Memory to store test configurations
BIST for CPLDs

*Cypress* 37x and 37K series CPLDs

- Incorporates BIST circuitry to test PLA portion of LB
- Test patterns: all 0s, walk 1 thru 0s, all 1s, walk 0 thru 1s
  - but this does not detect all stuck-at & bridging faults
  - must include walking two 1s thru 0s & two 0s thru 1s
    - the complete set is included in 39K manufacturing tests

<table>
<thead>
<tr>
<th>AND-plane</th>
<th>OR-plane</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0 0 0 0 0 0 0</td>
<td>1 0 0 0</td>
</tr>
<tr>
<td>0 1 0 0 0 0 0 0</td>
<td>0 1 0 0</td>
</tr>
<tr>
<td>0 0 1 0 0 0 0 0</td>
<td>0 0 1 0</td>
</tr>
<tr>
<td>0 0 0 1 0 0 0 0</td>
<td>0 0 0 1</td>
</tr>
<tr>
<td>0 0 0 0 1 0 0 0</td>
<td>1 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 1 0 0</td>
<td>0 1 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 1 0</td>
<td>0 0 1 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 1</td>
<td>0 0 0 1</td>
</tr>
</tbody>
</table>

Test Session 1

<table>
<thead>
<tr>
<th>AND-plane</th>
<th>OR-plane</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 1 1 1 0 1 0</td>
<td>0 1 1 1</td>
</tr>
<tr>
<td>1 0 1 1 1 1 1 1</td>
<td>1 0 1 1</td>
</tr>
<tr>
<td>1 1 0 1 1 1 1 1</td>
<td>1 1 0 1</td>
</tr>
<tr>
<td>1 1 0 1 1 1 1 1</td>
<td>1 1 1 0</td>
</tr>
<tr>
<td>1 1 1 0 1 1 1 1</td>
<td>0 1 1 1</td>
</tr>
<tr>
<td>1 1 1 0 1 1 1 1</td>
<td>1 0 1 1</td>
</tr>
<tr>
<td>1 1 1 1 0 1 1 1</td>
<td>1 1 0 1</td>
</tr>
<tr>
<td>1 1 1 1 1 1 1 0</td>
<td>1 1 1 0</td>
</tr>
</tbody>
</table>

Test Session 2

Note: AND-plane 1=bit, 0=bit-bar
OR-plane 1=active, 0=inactive

C. Stroud 11/06
DFT for CPLDs

*Cypress* Delta 39K

- Incorporates full scan design in macrocells
  - Boundary Scan interface to configuration memory & scan chain allows almost all manufacturing testing via BS interface
    - reduces ATE I/O pin requirements
  - Manufacturing tests uses scan chain to test:
    - programmable logic tests
    - programmable interconnect tests
  - RAMs routed to primary I/O for standard RAM tests
  - **Problem:** no scan chain included for FIFO flag outputs
- BIST developed for RAMs (like BIST for FPGAs)
  - To be used for manufacturing burn-in tests