Design For Test Methodology Applied Across Design and Support Cycles

RASSP Lockheed Martin Program Review
November, 1996
Camden, New Jersey

Lockheed Martin Advanced Technology Laboratories
R. J. Tarzaiski (609) 338-4046 rtarzais@atl.lmco.com
S. Sharma (609) 338-4029 ssharma@atl.lmco.com
RASSP DFT Methodology

TECHNICAL OBJECTIVES

• Develop singular test philosophy across all life cycle phases supported by a methodology using commercial tools
• Introduce DFT as early as possible to reduce cost/time for test at all life cycle phases
• Develop ABIST and BSCAN tools (LV Software) that can support board design

BENEFITS

• Consolidated requirements set maximizes reuse to minimize test development costs
• Offer potential cost reduction during each life cycle phase (design, manufacturing, field support)
• Reduce project life cycle test cost by 2.3X
• Reduce cost of test in manufacturing by 2.7X
• Reduce product life cycle cost up to 20%
  – Standardized reusable test strategies and testability building blocks enhance reuse
**DFT Integration Into RASSP Methodology**

**Available DFT Methodology Outputs**

1. **Enterprise Workflow**
   - DFT Methodology
   - Beta Site Courses
   - Application Documentation

2. **DFT Integration**
   - Commercial Tool Modifications
   - Strategic Tool Combinations
   - Test Strategy Diagram

3. **Reuse Element Generation**
   - Models
   - Procedures
   - Templates

**TECHNICAL OBJECTIVES**

- Define opportunities and methods to introduce DFT into all life cycle phases of a DSP project
- Reduce project life cycle cost by up to 20% by DFT inclusion
- Support DFT introduction by providing training in methodology and specific tool usage
- Integrate commercial tools for DFT into project workflow definitions to obtain advantages of cost/time reductions promoted by DFT

**BENEFITS**

- RASSP DFT Methodology effects cost/time improvements for their products > 2X savings in test
- Integrated tools minimizes disjoint test/validation
**RASSP DFT Generates Library-Grade Reusable Test HW/SW Elements**

**TECHNICAL OBJECTIVES**
- Create DFT component models/procedures which are installable in project designs across design, manufacturing, and field phases
- Contribute to architectures selection process as Figure Of Merit entry
- Create reusable elements
  - Procedure Templates
  - VHDL Models
  - Test Procedures
  - Test Vector Sets

**BENEFITS**
- Eliminate need for redevelopment of design, manufacturing, and field elements which span a variety of projects
- Shorten installation time of DFT for a project
- Reduction of actual test time and cost for a project

---

**Diagram**
- Design
- Manufacturing
- Field
- Reuse Library
- Project Specification

- Consolidated Requirements
- Testbench Component Models
- Test Strategies & Plans
- HW/SW Test Procedures
- Test Vector Sets
- HW/SW Test Procedures
**RASSP DFT CAE/CAD Tools**

- **Participants and Products**
  - Logic Vision
    - ICRAM BIST Suite
    - memBIST-XT (10/96 release)
    - ABIST to be developed to support preeminent role of BIST and BSCAN (12/96)
  - Teradyne
    - VICTORY
    - Parallel Port Tester (PPT)
  - ASSET Intertech, Inc.
    - ASSET
  - STS/Lockheed Martin ATL
    - LM ATL developed TSD EXCEL spreadsheets (5/96)

- **DFT Support Tools**
  - DETEX Systems, Inc.
    - STAT
  - NUWC
    - WSTA
  - IKOS Systems
    - Voyager
DFT Design Impact

• Insertions/Utilizations
  —Methodology Publication
    – Elements of Methodology installed at GES, Moorestown, NJ, E&M, Orlando FL
  —Beta Site Training/Other Presentation Support
    – Lockheed Martin Companies, JAST Program Office, commercial companies (Chrysler, Delco) investigating insertion of DFT into their design/manufacturing processes
  —AM3 Program
    – RASSP DFT methodology accepted as baseline Enhanced Built In Test (EBIT) insertion approach
  —LM Engineering Process Improvement (EPI) Activity
    – RASSP DFT methodology recommended as Lockheed Martin corporate standard
  —CAD/CAE tool integration activity
    – Several commercial tools modified/enhanced to support DFT insertion
• Related Effort
  —VTEST Program
    - RASSP DFT activity supplies baseline testing methodology as a reference for application of virtual test technology
Expanded Definition of “Test”

- **System Definition**
- **Architecture Definition**
- **Detailed Design**
- **Manufacturing**
- **Field Support**

The “Testing” Process

- **Detect**
- **Isolate**
- **Correct**

**Singular Approach**

The Packaging Hierarchy

- **Detect**
- **Isolate**
- **Correct**

- **Chip**
- **MCM/Board**
- **System**

**Singular Test Approach to achieve 4X.**
Relationship of Test Requirements, Test Strategies and Test Architectures

Test Requirements

- Test Strategies (TSD)
  - Test Means
- Test Architecture (TA)
  - Testbench Architecture
  - Testability Architecture
  - Test Architecture
  - Testbenches
  - DFT & BIST Features
  - Test Access & Vectors
  - Test Plans
  - Test Procedures
Application Example
RASSP DFT Methodology Goals and DFT Solution

• DFT Methodology Goals
  • Hierarchical approach integrated with RASSP processes
  • Foster reuse of test and BIST/DFT elements
  • Guided, but not driven, by existing tools
  • Accommodate presence of COTS in a system

• DFT Solution
  • Singular test architecture for Design, Manufacturing, Field to reduce development effort
  • Predictable, Verifiable, Measurable process for DFT development
  • BIST/Boundary Scan test means priority
  • Hierarchical, independent test and maintenance architecture
**Test Requirement Template**

**a. Test Phase**
- Design (e.g., simulation, prototype debug qualification test, etc.)
- Production (e.g., go/no test, diagnostic test, repair verification, etc.)
- Field (e.g., operational, organizational, intermediate, depot)

**b. Test Means**
- BIST
- Test Equipment
- Manual procedures
- Mix of above

**c. Test Mode**
- Power-on test
- On-line concurrent (including interfering vs. non-interfering)
- On-line non-concurrent (including periodic vs. event-driven, operator invoked or software invoked)
- Off-line

**d. Degree of Allowable External Support**
- Operator may be “in the loop” to help achieve requirement
- Troubleshooter may be “in the loop” to help achieve requirement
- Job performance aid may be used to help achieve requirement

**e. Fault Model Assumption**
The definition of a “fault.” For example, single stuck-at fault, multiple stuck-at fault, delay fault, intermittent fault, transient fault, etc.

**f. Quantitative Definition of Metric**
An equation defining the metric. For example, fault detection coverage might be defined as the total number of faults detected automatically by BIST, divided by the number of possible faults, with “fault” defined by the fault model above.

**g. Prediction and Validation Weighing Factors**
The factors used to allocate the requirements and later, to calculate system values from lower level values. For example, failure rate, usage rate, mission criticality etc.

**h. Quantitative Requirement**
The actual quantitative requirement, calculated by the “quantitative definition” above. For example “98%.”

**i. Allowable Requirement Compliance Tracking Methodologies**
The means that may be used to track compliance to the requirement throughout the life cycle of the system. For example, topological dependency models at the prediction stage, fault simulation at the validation stage, and instrumentation or automatic fault-history logging at the measurement stage.
Anatomy of a Test Strategy Diagram

For example, design flaws, manufacturing defects, or field failures

**DEFINITIONS**

**ATTR** Attributes being evaluated (e.g., cost, time, quality, etc.)

**Os** Output successful (e.g., detected quantity)

**Ou** Output unsuccessful (e.g., undetected quantity)

**TEST MEANS** Means of performing test (e.g., types of BIST, types of test equipment, manual procedures, etc.)

**TF** Transfer function (e.g., fault coverage percent)

Key construct to bridge requirements to implementation, manufacturing and field support.
Role of TSD

• Document the **Singular Test Philosophy** for all phases of a project

• Establish a monitored test strategy which will result in complete satisfaction of the consolidated requirements

• Compare predictions, verifications, and measurements to requirements throughout program life cycle

• Flag significant deviations from requirements

• Inform designer what is expected from different levels of test

• Ensure that test process is validated and accepted by designers
Test Strategy Diagrams

- TSD describes effectiveness of test means for detection, isolation, and correction of flaws/faults
- TSD used in sets implemented as EXCEL workbook
- TSD0/TSD1 set typical contents on 3 worksheets
  - Requirements, Predicted values, and Differences between requirements and predictions
- TSD0/1 will evolve to TSDn as system is better defined
  - Addition of worksheets to support monitoring verification and measurements at lower architectural levels
- Downward and upward data flow provided by EXCEL intersheet communication process
Typical TSD as an EXCEL Workbook

Sheet 1
- Requirements TSD

Sheet 2
- Roll-up TSD
- User Comments Area

Sheet 3
- Deviations TSD (Pred - Reqs)
- Constituent TSDs (Pred) (Verif, Meas later)
**EXCEL TSD Example**

**Microsoft Excel - 3DM2FE.XLS**

**Bridging, Open, Stuck-at Faults Handled During Manufacturing Test**

**Enter Requirements, Flow Down or Budgeted Values Here**

<table>
<thead>
<tr>
<th>Detection</th>
<th>Flow Pop. i.e. BRIDGING, OPEN, STUCK-AT FAULTS</th>
<th>PMEGA Weights</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1000.00</td>
<td>1</td>
</tr>
<tr>
<td>350.0</td>
<td>10.00%</td>
<td></td>
</tr>
<tr>
<td>315.0</td>
<td>40.00%</td>
<td></td>
</tr>
<tr>
<td>183.0</td>
<td>50.00%</td>
<td></td>
</tr>
<tr>
<td>34.5</td>
<td>50.00%</td>
<td></td>
</tr>
<tr>
<td>47.3</td>
<td>50.00%</td>
<td></td>
</tr>
<tr>
<td>95.00%</td>
<td>35.00%</td>
<td></td>
</tr>
<tr>
<td>650.0</td>
<td>20.00%</td>
<td></td>
</tr>
<tr>
<td>54.00%</td>
<td>40.00%</td>
<td></td>
</tr>
<tr>
<td>108.00%</td>
<td>50.00%</td>
<td></td>
</tr>
<tr>
<td>50.00%</td>
<td>50.00%</td>
<td></td>
</tr>
<tr>
<td>74.3</td>
<td>50.00%</td>
<td></td>
</tr>
<tr>
<td>99.40%</td>
<td>35.00%</td>
<td></td>
</tr>
<tr>
<td>617.5</td>
<td>0.00%</td>
<td></td>
</tr>
<tr>
<td>631.0</td>
<td>0.00%</td>
<td></td>
</tr>
<tr>
<td>703.0</td>
<td>0.00%</td>
<td></td>
</tr>
<tr>
<td>804.0</td>
<td>100.00%</td>
<td></td>
</tr>
<tr>
<td>100.00%</td>
<td>35.00%</td>
<td></td>
</tr>
<tr>
<td>878.5</td>
<td>22.00%</td>
<td></td>
</tr>
<tr>
<td>INSPECTION</td>
<td>BIST</td>
<td></td>
</tr>
<tr>
<td>618.0</td>
<td>22.00%</td>
<td></td>
</tr>
<tr>
<td>BS-ATE</td>
<td>22.00%</td>
<td></td>
</tr>
<tr>
<td>ATE</td>
<td>22.00%</td>
<td></td>
</tr>
<tr>
<td>MAN</td>
<td>22.00%</td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>35.00%</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 2.3.2.2-3a. BM3 TSD1 Sheet 1 - Requirements TSD for FE
**TSD Hierarchy**

- **Subsystem Level, FD TSD1**
  - **Sys Reqs**
  - **Flow Down Req Allocation**

- **Board Level, Arch TSD2**
  - **FE Reqs**
  - **Roll-up**
  - **FE**
  - **Prediction**
  - **Flow Up (Deviations)**
  - **Ver - Req Pred - Req**

- **Roll-up**
  - **FPCAP1**
  - **FPCAP2**
  - **FPCTL**
  - **Diff**
  - **Pred - Req**
TSD Value Population

- Three quantitative measures used in BM3 TSD structure
  - Transfer Function for fault coverage
  - Cost Attribute
  - Time Attribute

- **Transfer function** values selected from experience base or specific knowledge of test means coverage
  - First assignments will be Predictions - subject to modification

- **Cost attribute** values assigned on basis of economic analysis of the proposed application of test means sequence
  - fixed (test equipment) and variable costs (labor)
  - may be based on previous experience

- **Time attribute** value assigned primarily from experience base
**Test Means Sequence**

- General Rule: Assign test means in order of "effectiveness" measured by highest flaw/fault coverage or shortest execution times

- Always minimize the number of faults passed on to expensive, time consuming tests (where possible)
  - Manual usually most expensive and slowest - reserve for last
  - For BM3, the preferred performance sequence is:
    - BIST, BS-ATE, ATE, MANUAL

- Transfer functions combine as:
  - Total Coverage = \((1 - TF1)(1 - TF2)(1 - TF3) \ldots (1 - TFn)\)

- Sequencing and magnitude variations trade-offs are a significant feature of the TSD
  - Time and economic effects become manifest
Preliminary Economic Models for Prediction

- Must be based upon a specific project (e.g., BM3) a Company testing model (available test means) and the TSD structure for the project
- For BM3 the following test means are available
  — Inspection
  — BIST
  — BS-ATE
  — ATE
  — MANUAL
  — SIMULATION
  — OPERATIONAL TEST
Primary Economic Assumptions

• Number of FEs to be produced - 500
  — Total number of boards - 1500
• Labor rate - $75.00 per hour
• Production Interval - 1 year
• ATE Investment for project - $500000
• Test Means are **always** Applied Where Specified in the TSDs - No Tests Skipped
• Some Overlap Exists Among Faults Detectable by Test Means - Efficiency Better than simple application implies
  — Use efficiency factor to estimate this effect
### Fixed Cost Model

<table>
<thead>
<tr>
<th>Test Means</th>
<th>Fixture Cost</th>
<th>Non-Recur Labor</th>
<th>Test Eq</th>
<th>Cost/Board</th>
</tr>
</thead>
<tbody>
<tr>
<td>INSPECTION</td>
<td>1000</td>
<td>3000 (40 hr)</td>
<td>0</td>
<td>2.60</td>
</tr>
<tr>
<td>BIST</td>
<td>1000</td>
<td>15000 (200 hr)</td>
<td>1000</td>
<td>11.33</td>
</tr>
<tr>
<td>BS-ATE</td>
<td>4000</td>
<td>30000 (400 hr)</td>
<td>20000</td>
<td>36.00</td>
</tr>
<tr>
<td>ATE</td>
<td>4000</td>
<td>30000 (400 hr)</td>
<td>500000</td>
<td>349.00</td>
</tr>
<tr>
<td>MANUAL</td>
<td>1000</td>
<td>6000 (80 hr)</td>
<td>5000</td>
<td>8.00</td>
</tr>
<tr>
<td>SIMULATION</td>
<td>0</td>
<td>30000 (400 hr)</td>
<td>0</td>
<td>20.00</td>
</tr>
</tbody>
</table>

OPERATIONAL TEST - Normal Field Operation - No cost

TOTAL FIXED COST / BOARD - $ 426.93
# Life Cycle Test Means Application Distribution (BM3)

<table>
<thead>
<tr>
<th>Phase</th>
<th>#TSDs</th>
<th>INSP</th>
<th>SIM</th>
<th>BIST</th>
<th>BS-ATE</th>
<th>ATE</th>
<th>MAN</th>
<th>OP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Man</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>Fld-Dep</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Fld-Org</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

**TOTAL TEST MEANS APPLICATIONS**

4 1 8 6 6 8 1
### Test Time Model

#### Test Time Performance per Board

<table>
<thead>
<tr>
<th>Test Means</th>
<th>Indiv Test Time</th>
<th>Alloc Factor</th>
<th>Effic Factor</th>
<th>Effective Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>INSPECTION</td>
<td>300 s</td>
<td>4</td>
<td>1/4</td>
<td>300 s</td>
</tr>
<tr>
<td>BIST</td>
<td>0.1 s</td>
<td>8</td>
<td>8/8</td>
<td>0.8 s</td>
</tr>
<tr>
<td>BS-ATE</td>
<td>120 s</td>
<td>6</td>
<td>3/6</td>
<td>360 s</td>
</tr>
<tr>
<td>ATE</td>
<td>300 s</td>
<td>6</td>
<td>3/6</td>
<td>900 s</td>
</tr>
<tr>
<td>MANUAL</td>
<td>300 s</td>
<td>8</td>
<td>4/8</td>
<td>1200 s</td>
</tr>
<tr>
<td>SIMULATION</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OP TEST</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Variable Cost Model

• Assumption - Variable cost is **test labor** time

<table>
<thead>
<tr>
<th>Test Means</th>
<th>Test Time</th>
<th>Test Labor Cost/Board</th>
</tr>
</thead>
<tbody>
<tr>
<td>INSPECTION</td>
<td>300 s</td>
<td>6.25</td>
</tr>
<tr>
<td>BIST</td>
<td>0.8 s</td>
<td>-</td>
</tr>
<tr>
<td>BS-ATE</td>
<td>360 s</td>
<td>7.50</td>
</tr>
<tr>
<td>ATE</td>
<td>900 s</td>
<td>18.75</td>
</tr>
<tr>
<td>MANUAL</td>
<td>1200 s</td>
<td>25.00</td>
</tr>
</tbody>
</table>

SIMULATION - Part of Design Engineering Effort

OPERATIONAL TEST - No additional cost for observation of results.

• Total Variable Cost/Board (All Project Phases) - 57.50

• Variable Costs lower than fixed costs in this case
Practical TSD Application

- Original plan for TSD was population of complete hierarchy
  — Still valid concept where required
- TSD is a strategy evaluator and planning tool
  — Allocates cost, time, and fault coverage to test means
  — Allows consideration of effects of means interchange and fault coverage reallocation
- Hierarchical in nature to allow its use at various levels of packaging as required
  — All entries to all levels of system phases and packaging levels may not be needed
- TSD usage extent is project and Company specific
DFT Payback Analysis

Economic Justification
Test Economics

• Economics *always* an issue
• Generally accepted that DFT insertion can reduce project life cycle cost
• Applicable historical data base for cost justification is currently (usually) limited
• Test economics is *project* and *Company* specific
• Assumption-based economic analysis offers an alternative
Project Economics

• Two aspects of project economics coexist
  — Project Phase economics
  — Project Life Cycle economics

• Each view seeks to minimize cost in its own domain

• Recognize that conflicts will exist when DFT insertion is considered for a project
  — Basis: Increasing cost of design to achieve life cycle cost reduction
DFT Payback Analysis Procedure

• Specify a cost distribution over life cycle model
• Define assumptions or use actual values to partition costs among design, manufacturing, and field support phases
• Obtain pre-DFT testing cost
• Obtain post-DFT testing cost - Impose DFT methodology and reevaluate cost of testing
• Draw Conclusions
Life Cycle Cost Model

- Assumed LCC distribution across phases
  - Design 10%
  - Manufacturing 35%
  - Field Support 55%

- Model based on historical costs of a project producing large number (1000s) of boards

- Not obvious but absolute values for the estimates are not critical

- Limited model is still useful as long as limitations are understood
Economic Analyses

- Test equipment and TPS needs are established by project requirements
  - Both are major cost items

- Two possibilities for cost reduction:
  - Eliminate TE and TPS
  - Include ATE in all project life cycle phases and reuse TPS throughout project life cycle

- Typical existing economics analyses compare boundary scan (BS) and non-boundary scan test scenarios
Step 1: Pre-DFT Cost Model

- Design Phase (Test Development)
  - 30% design cost for test design
  - 3% of LCC

- Manufacturing Phase
  - 30% manufacturing cost for test setup and execution
  - 10% LCC

- Field Support Phase (Hardware intensive project)
  - 3% LCC for TPS reengineering (conservative)
  - 1.4% LCC spares test

- 17.4% LCC attributed to test
Step 2. Post-DFT Cost Model

• Design Phase
  — 20 % additional design effort for DFT insertion
  — LCC *increase* from 3 % to 3.6 %

• Manufacturing Phase
  — Design phase test vector (TPS) reuse and BIST / BS architecture reduce manufacturing test component of LCC to 3.75 % from 10 % (2.7 X)

• Field Support Phase
  — TPS reuse *reduces* cost of TPS reengineering and spares testing from 4.4 % LCC to 0.44 % LCC (10 X)

• 7.8 % LCC attributed to test
Conclusions of Payback Analysis

- Pre-DFT test LCC / Post-DFT test LCC = 2.3 X
- Manufacturing test cost reduction factor = 2.7 X
- Potential LCC reduction from DFT insertion
  - > 20 %
    - Direct cost reduction 17.4 % (pre-DFT testing) - 7.8 % (post-DFT testing) = 9.6 % LCC
    - Indirect cost reduction (lower spare count) = 6 %
    - Repair labor cost savings 5 %

- 20 % LCC reduction is 1/6 (approx) of RASSP goal of 4 X
FPGA-Based Board Level BIST Technique
Background

• Design dilemma
  — Extensive use of COTS components can restrict use of available DFT approaches - tends to create “untestable” designs
  — Need for DFT does not diminish

• Dilemma can be resolved for boards with bus-based architecture used in many systems
  — Processor/memory board
  — Memory board
Role of FPGA-based Board BIST

- Serve as a test means which can be installed and verified during project design and used during manufacturing and field support project phases
- FPGA-based board BIST can replace external ATE as a test means during manufacturing
- Test COTS components in field environment
  — Moves complex COTS components from “get out of the way” category into “follow/lead” categories
- Connect to existing boundary scan interface in a system to test non-boundary scan components
Description of FPGA-based Board Level BIST Concept

• Attach an FPGA to a system bus
  — Use existing reprogrammable or added device
• Design and install tester circuitry matched to bus components and JTAG interface in FPGA
  — Design and circuitry synthesis generated by SW tools
• Trigger BIST and have FPGA execute test of all components attached to bus then report results of test
  — Use selection capability of components for isolation
A low-cost, reprogrammable, bus-oriented, FPGA-based, embedded/external Board BIST

FPGA-based board level BIST provides controllability and observability of Non-compliant, JTAG, regular-structure COTS
Enhanced COTS Testability via FPGA-Based Board BIST

• BIST is performed at-speed
• Processor not required on the board
• Same BIST is executable in manufacturing and field test project phases
• Can use existing on board reprogrammable FPGA, added on board FPGA, or external FPGA connected at time of test
• No special operator skills required
• JTAG 4/5 line TAP interface
• Simple PC interface suitable for field support
**PC Control of BIST via PPT (Parallel Port Tester)**

- PPT is SW (Teradyne) which controls a JTAG interface via hardware connection to the parallel port of a PC
- PC can apply any vector supplied in Serial Vector Format (SVF) to JTAG TAP
  - Can include board BIST FPGA control and standard JTAG component I/O
- Standard PC assumes role of test equipment capable of BIST and JTAG component control in manufacturing or field environments
  - Portable PC can be used in field environment
Implementation Approach

• Use BIST tool to generate VHDL code for JTAG test access port
• Create VHDL tester code matched to components to be tested
  — Address and data generation, read/write commands
  — Functional/test selector circuitry
• Use VHDL models of components to be tested in testbench with tester code
• Synthesize VHDL code into FPGA program to create tester in hardware
Process Flow of FPGA Based Board Level BIST
Reconfigurable Embedded/External Board BIST

1149.1 TAP

SRAM BIST Controller

SRAM Collar

SRAM
Concept Verification

• VHDL testbench execution of external test of 256K x 4 SRAM has been demonstrated
  — Complete SRAM tester circuitry
  — Complete functional/test selector circuitry
  — Complete JTAG TAP interface
  — True external SRAM operation
Reuse and Lessons Learned Catalogs
Reusable DFT Methodology Components

• Selected reuse component source areas
  — Methodology
  — Test plan definition
  — Modeling and tool evaluation
  — Test economic analyses
Methodology Reuse Components

- Basic methodology application procedure
- Requirements consolidation procedure
- Requirements template
- Complete Consolidated Requirements document
- TSD EXCEL spreadsheet templates
- Economic analysis procedure for TSD value population
Test Plan Related Reuse Components

• Board examination procedure illustrated by BM3 test plan
• TSD-based path to development of test procedure
• Company and project test means analysis procedure and specific applications of the means to generic components
• Test procedure template
Reuse Components From Modeling and Tool Evaluation

- VHDL models verified for JTAG 1149.1 Test Bus Controller and 8245-class buffers
- Tool validations and procedures
- Board level BIST concept for bus-oriented boards
Reusable Features of Test Economics

• Procedure for populating TSD spreadsheets with economic data
• Spreadsheets as hierarchical (multisheet and multiworkbook) templates
• Procedure for high level economic analysis
• Procedure for use of TSDs for economic optimization
DFT Methodology Application - Lessons Learned

• Use of procedures and components of DFT methodology can be optimized
  — Entire DFT methodology is requirements-based
  — Test plans verify satisfaction of requirements using an optimized ordered set of test means
  — Modeling and tool evaluation contribute to both specific projects and future DFT applications
  — Test economic analysis always is important
DFT Methodology Lessons Learned

• Correct consolidated requirements specification is the primary enabler of the DFT methodology
• Prediction TSDs are best constructed by experienced designers
• TSDs should be applied as needed - a full hierarchical set of TSDs is not needed for every project
• Use the TSD to explore testing of project areas which are not fully understood
• Use the TSD for test sequence optimization
Test Plans
Lessons Learned

• Test plans must verify satisfaction of all requirements - process and customer specified
• Maximize use of BIST and boundary scan test means to minimize cost and time
• A comprehensive critical analysis of a Company’s testing infrastructure can suggest (or force) test means selection for projects
• Keep good records of test results to ease burden of future test definition
Modeling and Tool Evaluation
Lessons Learned

- Accumulate VHDL models from any source and add to a reuse library *before* a project starts
- Verify all models
- Consider new tools for opportunities to provide simpler manufacturing/field testing
- Always examine possibilities for implementing board level BIST to maximize testability
Test Economics Lessons Learned

• For a product manufactured in quantity, the cost of DFT will be justified by test cost/time savings in manufacturing or in field support

• Maintain detailed economic records of every project to support economic analyses - validated cost bases are invaluable