RASSP Design For Testability (DFT) Appnote

Abstract

The RASSP program has significantly benefitted with the development of the DFT Methodology and its application to the Benchmark efforts. RASSP Design For Testability ensures that hardware and software for a signal processing system is testable during all RASSP life cycle phases and that the system complies with all customer supplied requirements. DFT also generates significant economic benefits which reduce the overall life cycle cost of a product. Based upon a singular test philosophy of the DFT methodology the life cycle testability support concept .

Purpose

This note describes the role of DFT in the RASSP methodology and summarizes the life cycle singular test philosophy, procedures, and tools developed during the RASSP program. Technological and economic benefits produced by application of DFT to signal processing systems is also described and illustrated by examination of two benchmark projects.

Roadmap

 Dennis Basara

RASSP Design For Testability Application Note

1.0 Executive Summary

1.1 The Design For Test (DFT) Methodology - Complete Life Cycle Testability Support

The RASSP program has significantly benefitted with the development of the DFT Methodology and its application to the Benchmark efforts. RASSP Design For Testability ensures that hardware and software for a signal processing system is testable during all RASSP life cycle phases and that the system complies with all customer supplied requirements. DFT also generates significant economic benefits which reduce the overall life cycle cost of a produject. Based upon a **singular test philosophy** of the DFT methodology tThe life cycle testability support concept based upon a singular test philosophy of the DFT methodology is illustrated below in Figure 1-1 and includes along with a listing of technical objectives and benefits of the DFT methodology.

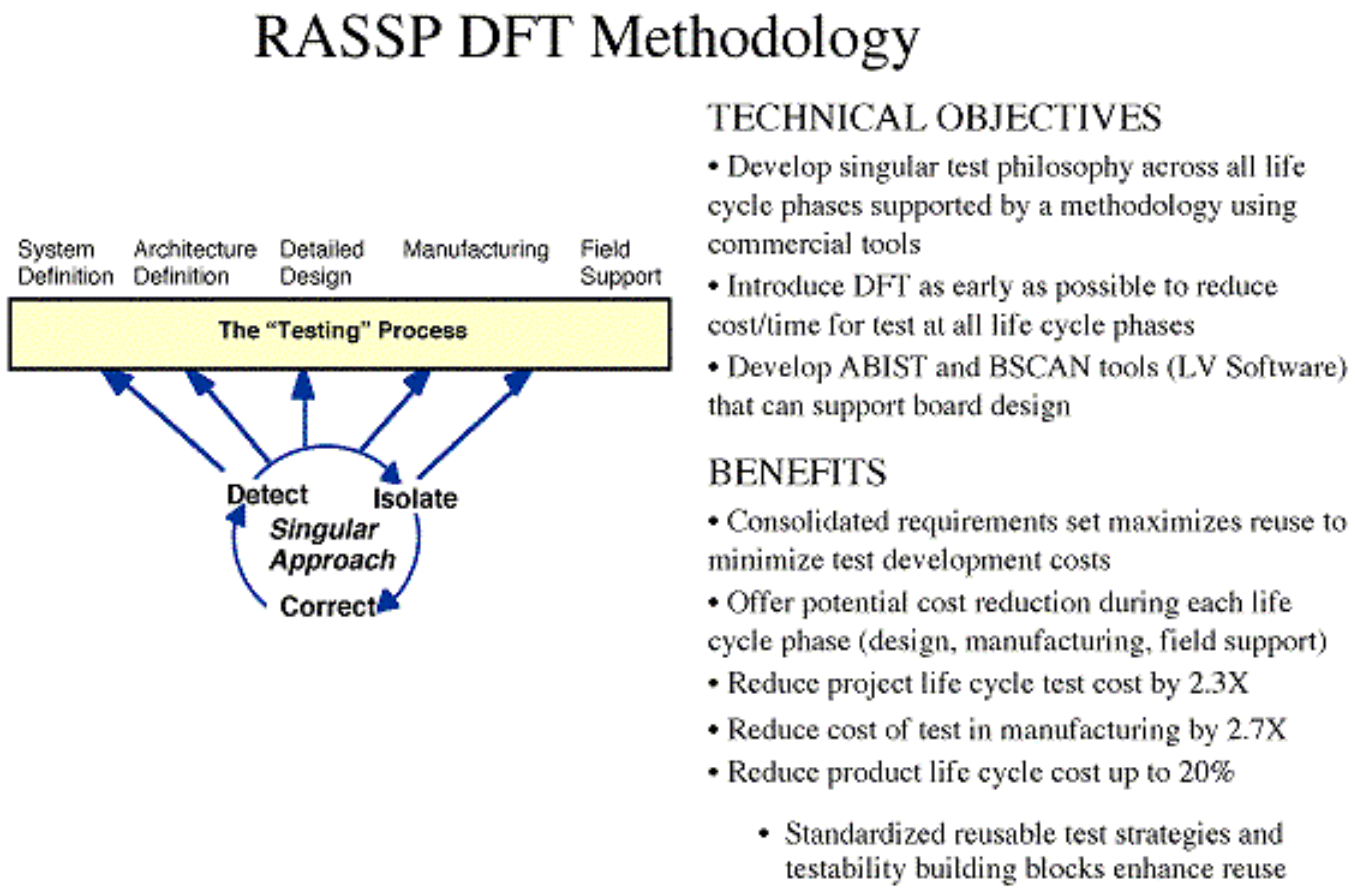


Figure 1 - 1: RASSP Uses a Common DFT Methodology Across All Project Phases.

1.2 The DFT Application Process - Tools and Design Element Reuse

Application of tThe DFT methodology is initiated by generating a set of **consolidated test requirements** fromby a team of experts with design, manufacturing, and field deployment experience. These requirements are processed to yield **Test Strategy Diagrams** (TSDs) that supportand a singular test philosophy which applies (to the extent possible) a

common test strategy to all life RASSP life cycle project phases. The TSD is an EXCEL spreadsheet based utility that manages (distributes) requirement allocation to all packaging levels and monitors multilevel conformance to these distributed requirements. In parallel with the TSD developmentSimultaneously, a complete supporting test architecture (TA) shown in Figure 1-2 is developed which specifies:

1. testbenches for design simulation,
2. details of board level DFT and BIST features, and
3. detailed specification of test equipment and test program sets (TPSs) is developed. This is followed by the developmentFinally of, detailed test plans and the procedures which implement these plans are developed.

DFT insertion begins during system design. It, is developed during simulation., and isThen it is installed and then validated in a prototype. This leads toIt then supportings manufacturing test, and eventually becominges a major part of (or the sole source of) the field supportability for a delivered system.

1.3 Summary of DFT Contribution to RASSP

RASSP DFT has developed a methodology that provides for the insertion of to insert DFT enabling features into signal processing system designs and contributess significantly toward the RASSP 4X time and cost improvement goals. In addition to demonstrable testability improvements, DFT offers;

1. the potential for up to a 20% reduction in life cycle cost,
2. a pre-DFT to post-DFT test cost ratio of 2.3, and
3. a post-DFT to pre-DFT reduction factor for manufacturing test cost of 2.7X.

Next Up Previous Contents
Next: 2 Introduction **Up:** Appnotes Index **Previous:** Appnote DFT Index

RASSP Design For Testability Application Note

2.0 Introduction

This Application Note is intended for system engineers and design engineers developing signal processing systems. The system engineer is presented with a methodology for incorporating DFT into a project to ensure compliance to customer requirements across the complete RASSP project life cycle. The design engineer is shown how to develop a complete testing approach in a verified design which translates directly to a product.

The RASSP DFT insertion methodology is demonstrated by application to all phases of one of the RASSP Benchmark 3 benchmark projects (Benchmark 3). Using systems fabricated from today's high density packaging systems which compress functionality at the expense of accessibility, the BM3 application illustrates how DFT inserts **full life cycle (i.e., system design to field deployment) testability**. Also examined are tinto systems fabricated using current high density packaging systems which compress functionality at the expense of accessibility. The economic benefits of DFT are also examined across the project life cycle and potential for life cycle cost (LCC) reduction is examined.

The system engineer applies the DFT methodology to **develop unambiguous, quantifiableed requirements. and to Aassistance is also provided in the process of architecture trade off and selection**. Testability insertion begins as consolidated requirements are developed by an integrated expert team representing design, manufacturing and field engineering project phases. The collaborative approach ensures that no single project phase is emphasized at the cost of another and that the end system will be optimized for testability throughout its life cycle. CReceived customer requirements are examined to ensure that they are consistent, realizable and valid. **Test means** for requirements verification are examined and a **singular test philosophy (STP)** is developed. Tthat maximizes the effectiveness of the selected DFT approach over the product life cycle is maximized by by using tests common to all life cycle phases is developed.

The design engineer **verifies** the operation and effectiveness of selected DFT features along with the functional system simulation. Component selection accompanies simulation to ensure that test circuitry can be integrated into the system hardware and software. IEEE 1149.1- and BIST (Built In Self Test) - enabled components and a complementary physical test bus architecture are added to board layouts. Test vectors are generated to be applied to DFT elements added to a design. Board level BIST technology is added to extend test coverage. A full set of test procedures is developed to meet the needs of development, manufacturing and deployment testing of the hardware.

RASSP DFT is supported with an extensive toolset and reusable templates supporting DFT during all the RASSP life cycle phases. Consolidated requirements are template based to ensure completeness. Test Strategy Diagrams (TSDs) implemented as EXCEL spreadsheets derived from the consolidated requirements distribute the requirements to lower packaging levels and support enforcement of requirements. A complete test architecture developed in parallel with the TSDs supplies a full set of test plans and procedures which measure compliance to requirements. A complete set of VHDL-based simulation and test vector generation tools supports development of JTAG--based tests and installation of board level BIST which maximizes hardware testability during manufacturing and field deployment. Any required supplementary ATE (Automatic Test Equipment) testing is also supported by definition of critical test points not covered by JTAG or BIST testing.

Page Status: in-review, January 1998 *Dennis Basara*

RASSP Design For Testability Application Note

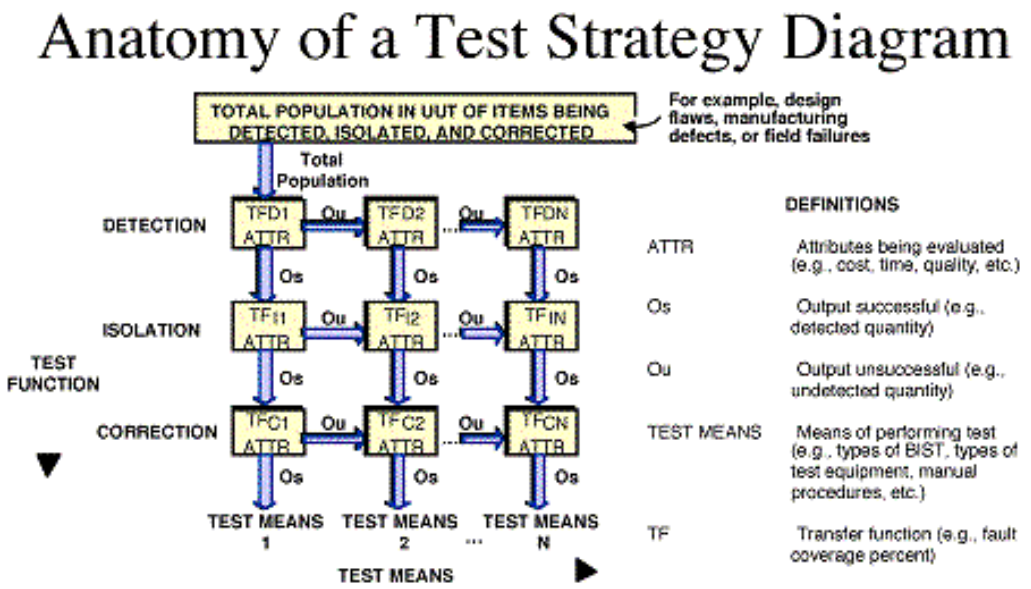
3.0 Technology Description

RASSP Design For Test technology has inserted testability into all phases of the RASSP methodology (**RASSP Methodology Version 2.0**) - from system design to field support. This insertion has been accomplished using a combination of existing and modified commercial simulation, CAD, and specialty DFT tools. along with Sseveral of the DFT specialty tools were developed during the RASSP program.

During early project phases, DFT influences the functional analysis and alternative architecture alternative evaluation and functional analysis viathrough the enabling technology of **functional dependency modeling** (DM) which examines testability of a system at the functional block diagram level. DM technology defines faults and tests for each system function and permits a parametric analysis of the consequences of limited measurement capability of each of the system functions. The designer uses "What if" analysis to rate the testability performance of alternative architectures and different proposed physical implementations of system functions.

A technology **developed** during the RASSP program is **template based requirements analysis**. Templates ensure uniform, unambiguous expression of requirements by an integrated product development team (IPDT) with expertise in the areas of design, manufacturing, and field support. **Consolidated test requirements** used throughout a project life cycle contain both customer and derived requirements added to ensure quality and producibility in the end product. The system designer is forced to examine each requirement from the viewpoint of eventual quantitative measurement to establish that each requirement is consistent, realizable and valid.

After requirements are defined, a **singular test philosophy (STP)** applicable to all life cycle phases is designed and expressed in a series of **test strategy diagrams (TSDs). As can be seen in Figure 3-1, the TSDs** which allocate detection, isolation, and correction requirements across RASSP methodology phases and packaging levels for specific fault classes such as shorts, opens, and single stuck-at faults. TSDs are linked across RASSP phases and **upward and downward** in the packaging level dimension to implement both requirements distribution and compliance monitoring. Designers at any level interact with the appropriate TSD to confirm or deny the ability of a specific system element (e.g., board) to satisfy its allocated requirements. Contents of a typical TSD are shown below. TSDs can contain prediction, verification, or measured values. One TSD is used for each **fault class** such as **opens, shorts, and stuck-at faults**. Quantitative detection, isolation, and correction values are allocated across a series of test means and the total coverages for the fault class being controlled are calculated and displayed. Measurement and verification TSDs for a specific fault class are compared to the related prediction TSD to monitor requirements compliance. As described in the Application Example below, the TSD is implemented as a multisheet Microsoft EXCEL workbook. This format supports reuse by simple modification of TSDs developed on earlier projects to current projects.



Faults are resolved by applying allocated test means to the processes of detection, isolation and correction. Each test means operates on a specific fault class to reduce the number of undetected, unisolated, and uncorrected faults. For each "box" shown, a fault population enters at the top or from the left. Successfully processed faults exit at the bottom and proceed to the next stage of the detection, isolation, correction cycle. Unsuccessfully handled faults exit to the right to be processed at the same stage.

Figure 3 - 1: Form and Contents of a Test Strategy Diagram.

As shown in Figure 3-1 and discussed in the Executive Summary, definition of the **test strategy** is accompanied by the following items:

1. definition of a supporting **test architecture**, including a testbench architecture to support design phase testing,
2. a testability architecture to define DFT/BIST features, and
3. a tester architecture to definedefne the physical implementation of the test strategy.

The test strategy and architecture result in **test plans** from which a set of **test procedures** areis developed. RASSP developed the framework to formalize the means needed to implement thisa test strategy.

During design, DFT features such as **JTAG strings** and **FPGA-based board level BIST** are simulated on a VHDL testbench and their effectiveness is evaluated. Specialty tools designed to create and manage these features supply VHDL code used in the simulations and the test vectors required for testing during design and after when the design has been reduced to hardware. A technology driver in **testability analysis** and **test feature installation** for RASSP DFT was a technology driver in testability analysis and test feature installation and resulted in contributing refinements to commercial test tools (VICTORY, Parallel Port Tester) and development of several new BIST insertion tools (LogicVision). Simulation efforts result in verification of the proposed test approach and generate**test program sets (TPS)** which will be used on the actual hardware. Satisfaction of **prediction** TSD-assigned hardware requirements to system components is verified and the path to actual performance **measurement** in hardware is established via reuse of the simulation TPS. In addition to Reuse enhancingenhance thes likelihood of first pass success for hardware test, and, in addition,reuse results in significant hardware-based test development time and cost economies.

Once simulation is completed, hardware prototypes are produced and physical testing begins. U using Test Pprocedures defined in the test architecture definition phase and now verified by the simulation exercises, physical testing is performed. Actual measurements during manufacturing and acceptance testing substantiate the satisfaction of all requirements, and the system is ready for field deployment where it is subject to the pre-established maintenance philosophy defined in the field phase measurement TSDs.

Next Up Previous Contents

RASSP Design For Testability Application Note

4.0 Technology Description

4.1 Problem Definition and Approach to Solution

The major demonstrabledemonstration target for application of RASSP DFT methodology to a signal processor system was the **AN/UYS-2A functional element (FE)** as defined in the Benchmark 3 (BM3) project. The FE consisteds of one controller (called FPCTL) and two bus-oriented processor PC boards (called FPCAP) of moderate complexity. Components primarily included processors, memory devices, and FPGAs. Under the RASSP program, several BM3 FEs were fabricated for delivery and insertion into active AN/UYS-2-A systems for testing in accordanceing witho a short schedule. For this reason, DFT was applied in a non-interfering "shadow" mode in order to examine the addition of DFT to the FEs of BM3 without the risk of imposing new technology into thea time-critical development path. The full suite of DFT techniques was applied to the FE boards, mainly to the controller (FPCTL), as the hardware and supporting software were developed. The benefits of DFT inclusion were demonstrated during simulation exercises. Application of DFT technology in the serial manner described in the **RASSP Design For Testability (DFT) Methodology Version 1.0** document resulted in a sound test strategy which provides could enhancements fore future versions of the FE hardware. DFT application to the BM3 FE **also advanced the state-of-the-art of PC board test coverage** to memories and other components with regular structures which had not previously been testable by on-board means.

4.2 Stepwise Application of Methodology

Application of DFT methodology can be described in a sequence of six basic steps deployed throuighout the project life cycle.

4.2.1. Dependency Modeling to Support Architecture Selection

Functional Dependency Modeling (DM) is a tool that allows DFT analysis to begin during the earliest stages of a project. DM consists of developing a model of the dependency relationships among system functions to allow effects of failure of any function to be propagated to all succeeding functions. Failure modes for each function are defined and the DM tool identifies consequences of designer-initiated assertions of failures on succeeding functions and on the entire system. DM tools implement a form of propositional logic to model tests, failure modes and the interdependencies of these entities. For RASSP, it is first applied during concept definition and its use continues through architecture investigation and selection phases. For BM3, DM was used to locate ambiguous (indistinguishable) faults on both the FPCAP and FPCTL boards and to identify consequences of potentially inaccessible circuit nodes. In so doing, DM was used to support **selection of a preferred architecture** from the testability viewpoint. Two tools, WSTA by Naval Undersea Warfare Center (NUWC) and STAT by Detex Systems Inc., were used for parametric "what if" analyses of functional block diagrams (FBD) ofor both types of boards used in the BM3 FE. Functional block diagrams are pictorial representations of a system's functional elements and their interrelationships. They are constructed by system engineering during concept development. FBDs indicate how each system function depends upon the others of the complete set. Certain functions, for example, will execute successfully if related previous functions execute properly. Conversely, failurs introduced early into a functional chain typically result in observations of failures in execution of one or more subsequent functions.

The most significant testability information supplied by DM was :

- a recommended binary test strategy tree - an **ordered sequence of tests alternatives** selected from a binary tree structure based upon a PASS/FAIL result for each test as it is executed
- recommendations for an optimized BIT (Built In Test) - a series of tests which leads to the **quickest verification of a working system** - this test series follows the PASS route from application of the first test through to the establishment of a **no-fault** condition
- identification of **test ambiguity groups** - listings of components (and their **aspects**, or failure modes) which could be responsible for indication of failure by a given test. An ambiguity group larger than unity implies that a failure cannot be isolated to a single component. Ambiguity group data can signal existence of potential design problems early in the system design phases.
- recommendations for **addition of tests** specified as **excluded** by the analyst during "what if" tradeoff analysis. Testpoint access can be controlled during DM analysis to identify a test sequence which supports a minimal

cost/time strategy. When excess restriction of test point access is imposed, testability becomes impaired. Test addition recommendations are indicators of this condition.

Dependency modeling is particularly useful in gaining insight into system testability very early in the design process, even **at the concept phase, as soon as the functionality of the system has been defined.** Figure 4-1 illustrates the concept of a Functional Block Diagram. and iIt also can continue to make significant contributions to the system development through architecture selection. A special feature of DM is that the model remains completely adaptable to the level of system information available at the time of the testability examination. More than one modeling exercise is likely as the system design continues to evolve.

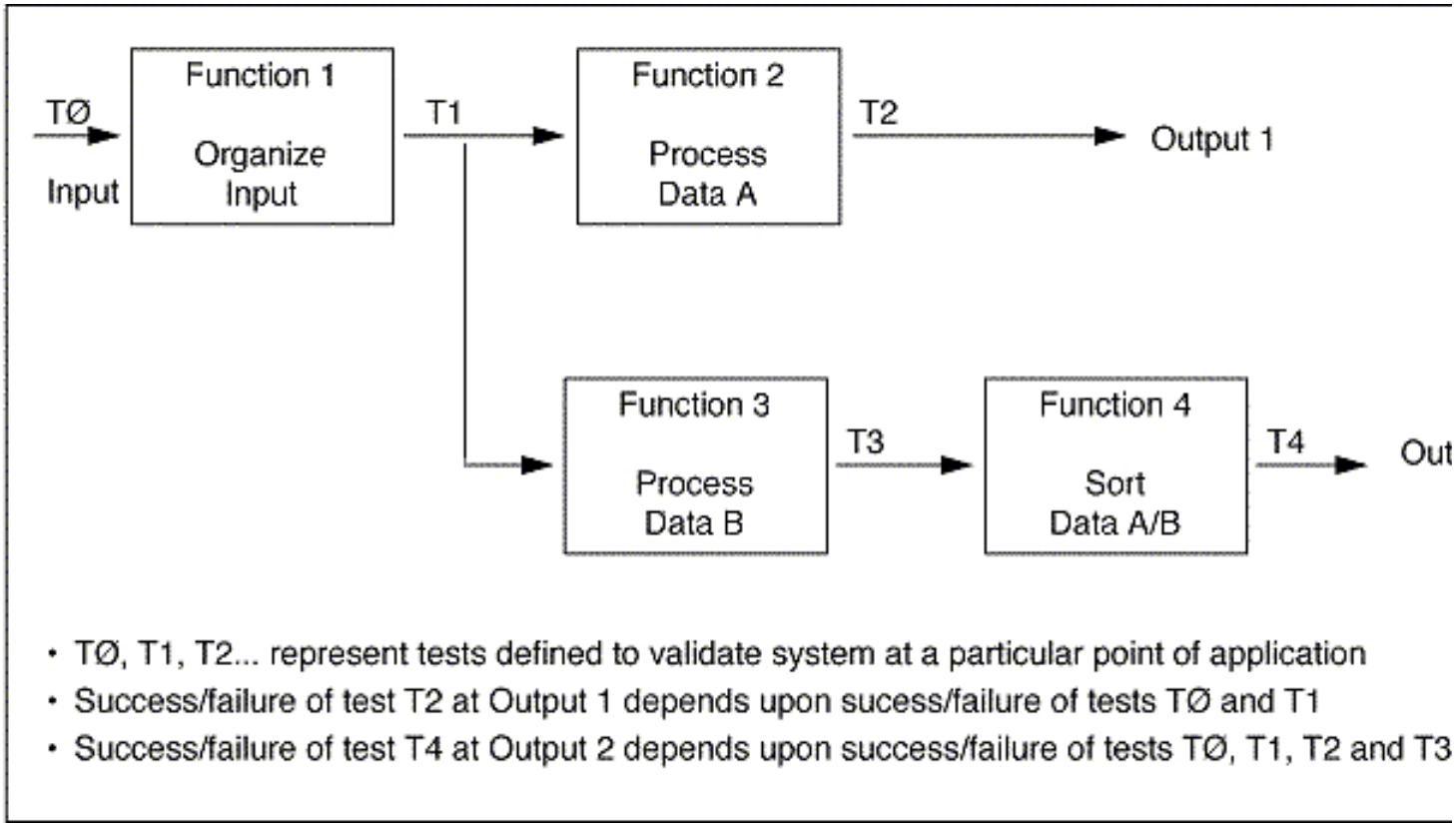


Figure 4 - 1: Example of Functional Dependency Model.

During concept development, time- and cost -independent dependency modeling can provide a definition of ambiguity groups resulting from imposition of test point restrictions and can give a very early indication of testability problems. As the system becomes better defined, test time and test cost can be factored into a dependency model if the designer can supply estimated or actual values for component mean time between failure (MTBF) and mean time to repair (MTTR). It is also possible for the user to prioritize known component function criticality. The DM recommended test strategy will be biased by the relative values of these three factors.

For candidate architectures, during architecture tradeoffs and selection, quantitative trade-offs based on:

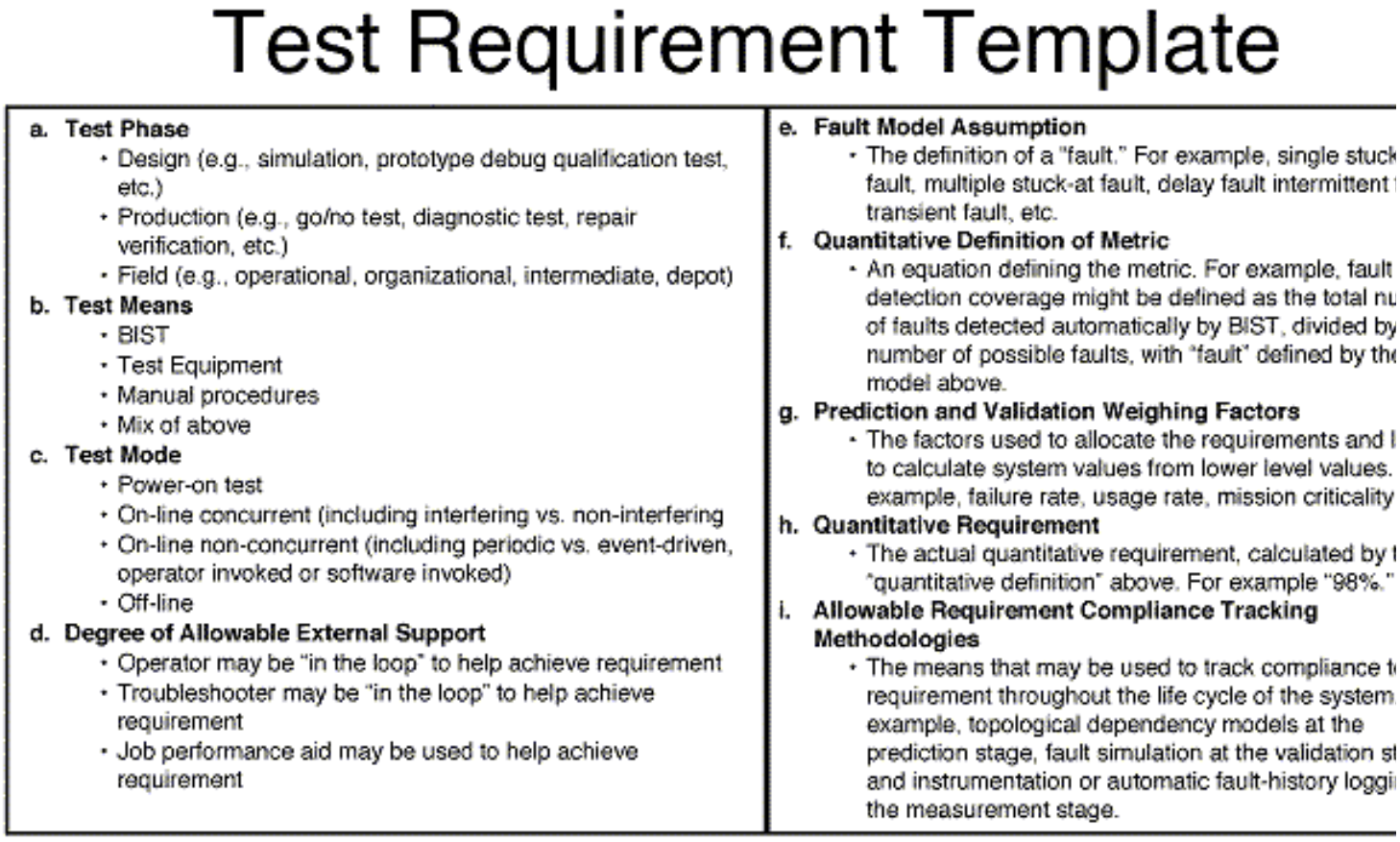
1. the number and size of ambiguity groups;,
2. extent of BIT;,
3. test strategy complexity; and
4. cost and repair time can be used to quantify criteria for candidate architectures being compared.

Dependency models are useful in identifying opportunities or requirements for adding specific testability features. Models can be exercised by denying access to selected test points associated with functions in the model. Access restriction corresponds to simulating a system with real physical accessibility limitations. For BM3, the Myrinet interfaces and associated RAM were, in fact, practically restricted because of bus speed and special low (1.6V) voltage requirements. No standard test equipment would likely be able to evaluate the status of these components at the board level. Dependency analyses quickly identified the ambiguity groups created by the restrictions and made this information available to the designers. The tool thus provided the designer an **early alert to testability problems**. DM also revealed that isolated FPCTL or FPCAP boards presented challenges to testability individually but when connected together, the combination

possessed alternative access paths which could evaluate operation of difficult to test components. DM can thus detect a potential problem and may also suggest possible solutions.

4.2.2. Collection and Specification of Requirements

A multi-step procedure was developed to achieve a **consolidated requirements set** having significant commonality across design, manufacturing, and field deployment project phases. First, candidate requirements were selected by a **team** consisting of engineers and test support personnel with design, manufacturing, and field test expertise. Each individual represented his/her own area to ensure that **explicit customer requirements** would be met in each of the life cycle phases and to ensure that any **derived requirements** (not explicitly stated by the customer), needed to build quality and, reliable equipment would be added to the formal requirements list. Requirement **templates** designed to unambiguously quantify requirements and the means of their measurements were used to ensure sufficiency and consistency of representation of the requirement information. These populated templates were the primary output from the requirements generation process. The content of a requirements template is shown in Figure. 4-2.



4 - 2: Template Used For Quantitative Unambiguous Requirement Specification.

Collected requirements weare grouped into two classes :

1. **conformance** requirements, measured with a go/no-go test, typical of an acceptance test, and
2. **TSD-controlled** requirements, to be distributed hierarchically through all RASSP phases and to all packaging levels via a hierarchy of test strategy diagrams described below in step 4.0 of the Methodology application.

4.2.3. Requirements Consolidation and Test Strategy Selection - Singular Test Philosophy Development

Requirements consolidation consists of collecting all customer defined requirements and integrating them with a supplementary set of RASSP process-related and fundamental fabrication quality requirements. This combination ensures that all customer needs will beare satisfied, that advantages associated with use of the RASSP DFT methodology will be realized and that quality equipment will be manufactured and delivered to the field. The steps summarized in Table 4-1 were followed to generate the set of consolidated requirements.

Table 4 - 1: Summary Procedure For Requirements Consolidation

1.	Create a set of requirement templates for the 3 project phases: design, manufacturing, and field support
2.	Check the inter-phase inherited anomalies for consistency and completeness: design flaws escaping to manufacturing , design flaws escaping to field, and manufacturing faults escaping to field
3.	Within each project phase, look for opportunities to merge the identified flaws/faults within the specific phases. Consider **test means** and **available support** as first level bases of comparison. Document differences, if any, in **quantitative requirements** for requirements which are merged.
4.	Scan through all test phases for common test means at **detection** and **isolation** coverage levels. Do **not** consider the inherited faults in this comparison. Formulate a preliminary core Singular Test Philosophy (STP). Also do not consider correction coverage since, in most cases, the test means will be manual procedures only.
5.	Consolidate the merged requirements formed in Step 3 across the 3 test phases with the intent of applying the preliminary STP to **all** phases, if possible. Several possibilities exist:
	• the STP applies to the 3 sets of preliminary merged requirements
	• the STP can be modified by addition, deletion, or reordering during negotiations among representatives of design, manufacturing, and field
	• individual templates can be modified to force conformance to the STP
	• individual templates can be tagged as not compatible with the preliminary STP
6.	Examine applicability of the STP to the inherited requirements. Negotiate an agreement on the STP by any of the means identified in Step 5.
7.	Perform final consolidation by negotiation among the 3 phases:
	• choose a final STP
	• add phase-specific supplemental items to the STP as required
	• modify individual templates to conform to the STP as required

8.	Document the STP along with any exceptions and attach the documentation to the Consolidated Requirements Document.
9.	Sign off the Consolidated Requirements Document.

A **test strategy** is a specification for an ordered application of a series of test means which will lead to defined levels of coverage for detection, isolation, and correction of specified fault classes. Coverage specifications are quantified and measurement techniques are specified in detail in the requirements templates. RASSP DFT seeks a test strategy called a **singular test philosophy (STP)** which is a strategy that can be applied during design, manufacturing, and field deployment. It is a desired goal because it allows reuse of test stimuli developed during simulation in manufacturing and in the field. The need for specialty tests is minimized and significant costs associated with development of phase specific test program sets (TPS) are eliminated or greatly reduced.

For BM3, common sets of test means to cover the phase-specific faults for each project phase were relatively easy to extract. A pattern of BIST followed by Boundary Scan test followed by less automated testing means was developed in each case. Negotiation sessions among the phase experts then examined the test means series extracted from the design, manufacturing and field environments with the intent of discovering possible application of a test means series **across** all phases. Examination again showed that a preferred arrangement existed - BIST, Boundary Scan (BS) Automated Test Equipment (ATE), non-BS ATE, and manual test. In some cases this basic series needed to be supplemented by means such as inspection or simulation. As will be discussed in more detail in the steps presented in the following example, an STP did exist for BM3.An STP did exist for BM3.

An example of using the individual steps identified in Table 1 and applied to BM3 will now be discussed. The cost-benefit analysis of applying these steps to a project can be found later in this Application Note in Section 4-4, DFT Economic Analysis.

Step 1.
A set of requirements templates was completed for the design, manufacturing, and field deployment project phases. The purpose of using templates to specify requirements was to unequivocally define requirements quantitatively and to define metrics which would establish whether conformance could be established for a given requirement.

An annotated template establishing the requirement for "Detection coverage for design flaws" along with annotations which describe the individual entries is shown below. It is typical of the templates for all project phases.

Requirement Name : Detection coverage for design flaws
Realizable _____ Consistent _____ Valid _____

a. Test Phase

- Design Phase selection by definition.

b. Test Means

- Simulation
- BIST
- BS-based ATE
- Non-BS-based ATE
- Manual procedures

These are the selected ordered set of test means available for detecting design flaws. Simulation is the primary method at the modeling level. Other means apply to the design phase up to and including production of the first prototype hardware. As will be seen in Step 7 where test requirements are finally consolidatedlater, the last four test last means belong to the STP. Simulation is a supplemental test means appropriate to the design phase. These test means are assigned independently by the expert representing the particular project phase, in this case **Design.**

c. Test Mode

- Off-line

  The design phase is a non-operating test.

d. Degree of Allowable External Support

- Unlimited

  No limitation is made since any available technician and/or engineering support will be made available to support the design effort.

e. Fault Model Assumption

- Design flaws : partitioning, interface, timing, model, algorithm

  The listed set of design flaws are typical of a board design project.

f. Quantitative Definition of Metric

- Total number of design flaws detected, by divided by the total number of design flaws

  This ratio defines the effectiveness of the design process.

g. Prediction and Validation Weighting Factors

- Flaw distribution profiles

  Previous experience with board design processes establishes a likely distribution of similar flaws on future projects. These factors represent the capability of a specific companyCompany with a specific type of project.

h. Quantitative Requirement

- 99%

  This number must approach 100% for a reasonable design process. Some tolerance must also be allotted to reflect the potential for an imperfect process.

i. Allowable Requirement Compliance Tracking Methodologies

- P - analysis - rate of flaw discovery
- V - simulation (testbench, functional, mission)
- M - data collection and statistics

  Prediction of design flaws is based on analysis of the circuit function and the typical rate of discovery of flaws for a circuit of that type. Fewer flaws per unit time will be uncovered as the design approaches its completion.

  Simulation is the primary design analysis tool. Lack of design flaw effects during simulation establishes the capability of a design. The rate of discovery during simulation can be compared with the predicted values for a consistency check.

  The ultimate means of compliance tracking is actual measurement of performance of a design. A design performing as expected indicates the absence of design flaws.

Step 2.
Templates showing coverage for inherited anomalies were added to ensure that noany design flaw escapes (non-detections) from design to manufacturing and field phases or manufacturing faults from the manufacturing to field phase were accounted for. This coverage is added to reflect such possibilities and their effects upon satisfaction of the requirements. The possibility of inherited anomalies generally reduces the levels of detection, isolation, and correction which can be achieved by any real physical system.

Step 3.
An effort was made by each of the project-phase-specific representatives to examine the extent to which flaws/faults (titles of templates) in their respective areas could be "merged" so that the number of templates wais minimizreduced. This was

accomplished by searching the requirements templates for commonality among test means, support requirements, test mode, and quantitative requirements for the baseline set of flaws/faults established by that expert. **This within-phase flaw/fault consolidation was performed independently for each of the phases of design, manufacturing, and field support.** As flaw/fault consolidations occurred, the fault model entry (template entry e.) would bere modified to reflect the extension of coverage. **and Iindividual quantitative requirements were explicitly specified as needed** to document the post-merged results and preserve the established quantitative requirement values. Consolidation never **eliminates** important established requirements but may combine them. For BM3, an example of a fault merge is the template for "Detection coverage for manufacturing faults - Bridging, Open, and Stuck-At".

Step 4.
Following the within-phase flaw/fault consolidation, the test means for the identified coverages for **detection** and **isolation** within each test phase were examined for commonality with the intent of establishing a candidate local (within-phase) singular test philosophy (STP). For BM3 the test means subseries : BIST, Boundary Scan ATE, and standard ATE wereas appropriate. BIST and Boundary Scan-based The first two testing means weare very cost effective from the life cycle point of view and theseir tests weare executed in the shortest times. These tests are introduced during early simulations and, once developed, accompany the product through the manufacturing and field deployment phases. The need for specialized manufacturing or field testing is reduced or eliminated.

ANeeded additional coverage needed in any phase for a specific flaw/fault could be provided by supplementing the core test means set with flaw-/fault-specific test means. Inherited flaws/faults were not considered in the search for an STP to prevent forced commonalities from having an excessive biasing influence (feedforward or feedback) on any candidate STP. Correction is not included as a comparison category, since correction is almost always a manual process.

At the end of this step, the requirements have been **merged by test means within their respective life cycle phase**s. Three phase-specific candidate STPs exist.

Step 5.
Examination of the sets of test means applicable to the individual phases will exhibit the likelihood of obtaining a true STP **across all phases** at this point. Commonality will lead to a merging of the sets of test means across the phases. Lack of commonality will indicate that a singular test philosophy may not exist. In most cases, the extent of an STP derived at this point will be a core STP which may need supplementationg within individual life cycle phases. Results of the attempt to generate an STP may be any of the following cases :

- An STP can be formulated across all life cycle phases.
- A baseline for an STP can be established and applied across all phases if additions, deletions or reorderings of test means can be accomplished.
- Individual templates can be arbitrarily modified to conform to the STP.
- Individual templates can be tagged as not compatible with the preliminary STP.

The first two cases represent situations where an STP can be readily defined. **For BM3, the second case applied**. In all but a few instances, the original assigned ordering of in-phase test means in the templates was very similar and it was relatively easy to reorder the individual test means sequence for a specific nonconforming flaw/fault to bring their final ordering into conformance with the majority of the test sequences. Thus, the basis for a core STP was found.

In a few cases, the core set of test means was supplemented with appropriate means. In particular, INSPECTION was added as the first test means for manufacturing to consider the reality that a manual or assisted inspection should precede execution of BIST or any subsequent means when physical faults such as shorts or opens are the faults of interest. SIMULATION was added as a supplemental test means for design.

Step 6.
The STP was next applied to the inherited anomalies. For BM3, no significant problems were encountered and the STP was readily applied to the anomalies. Some reordering was required in a manner similar to that used in the across-phase consolidation.

For BM3, little effort was required in this area.

Step 7.
Final consolidation was achieved by taking the cross-phase STP and adding the supplemental test means as required.

For **system** test, the test means arrangement is : SIMULATION, BIST, BSCAN ATE, NON-BSCAN ATE, MANUAL

For **manufacturing** test, the test means arrangement is : INSPECTION, BIST, BSCAN ATE, NON-BSCAN ATE,

MANUAL

For **field** test, appropriate test means sets are :

- At Organizational level:
  - For inherited flaws/faults : BIST, MANUAL, MISSION OP

MISSION OP is added as a supplemental test means and corresponds to observation of the continued operational status of the equipment during mission operation.

- For all field faults : ONLINE CONCURRENT BIST, OFFLINE NONCONCURRENT BIST, OFFLINE BIST, MANUAL

In this case, the basic sequence is BIST, MANUAL to conform to the requirement that the equipment execute BIST during field deployment.

- At Depot level :
  - BIST, BSCAN ATE, NON-BSCAN ATE, MANUAL

This arrangement is essentially the same as that for the manufacturing test since field Depot test corresponds to return to the manufacturing source for evaluation/repair.

The **core STP** is evident in all cases :
BIST, BSCAN ATE, NON-BSCAN ATE, MANUAL

Steps 8. and 9.
The STP was documented for BM3 by generation of a **Consolidated Requirements document.**

4.2.4. Generation of TSDs and Test Architecture

The separate template sets, each containing the STP and any supplemental test means, were used to generate **test strategy diagrams (TSDs)** in their respective phases. The TSD is a two dimensional (see Figure. 3-1) array displaying fault coverage for detection, isolation, and correction for each of the test means used. Specific fault coverage for any test means is called the transfer function of the test and is the ratio of the number of successfully handled faults to the total number of input faults. Test costs and test times are also entered into the TSD as test attributes. Individual TSD arrays can contain requirements, prediction, verification, or measurement values, or differences between the cells of requirements and one of the other TSDs. A **prediction TSD worksheet** contains requirements, their predicted distributions, and the prediction minus requirement values in a specific RASSP project phase. The differences yield a quantitative statement of conformance to requirements. An EXCEL worksheet version of a requirements TSD for the FE at the subsystem level is shown in Figure. 4-3. At the architecture selection and definition phases, two coupled TSD arrays on the worksheet will contain experience-based predictions and differences between the requirements and predicted values. As the project advances, and simulations develop and measurements made on hardware become available, TSDs contained on other worksheets of the EXCEL workbook are created to measure simulation and hardware conformity to requirements.

EXCEL TSD Example

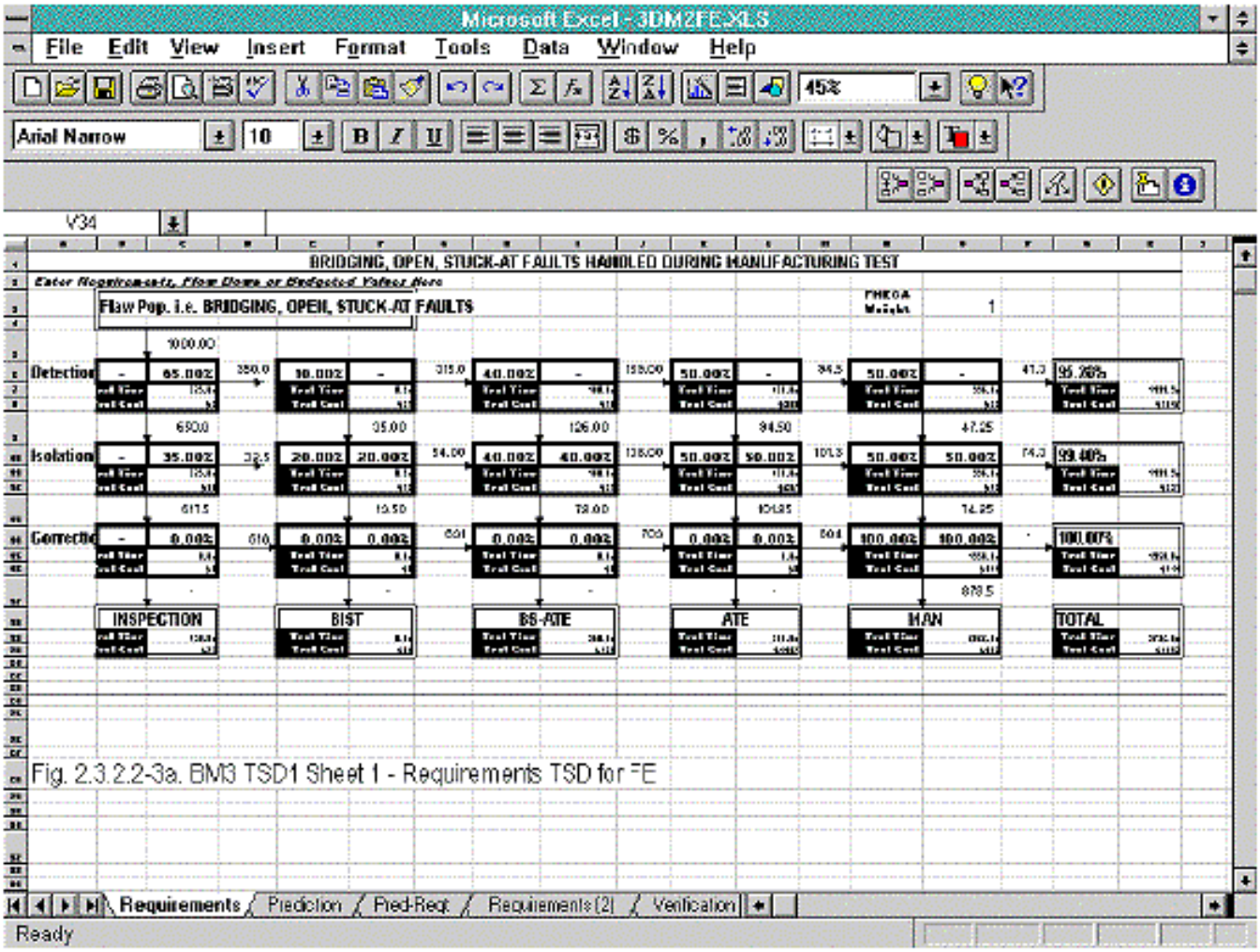


Fig. 2.3.2.2-3a. BM3 TSD1 Sheet 1 - Requirements TSD for FE

Figure 4 - 3: TSD for open, short, and bridging faults for manufacturing test of BM3 FE.

For BM3, requirement/prediction TSDs for selected flaws and faults during design, manufacturing, and field deployment were produced for the FE as a **subsystem** and for the FE divided into its three **PC board** (one FPCTL, two FPCAP) constituents. Other TSDs could have been generated, but the handling of these fault classes were well understood, so and standard fault management control techniques would bere applied. **TSDs are used to their best advantage to specify/monitor coverages for faults not fully understood.**

4.2.4.1. TSD Transfer Function Values

TSD Transfer Function Values A transfer function is a quantitative specification of detection, isolation, or correction coverage by a specific test means. Specific parameter values assigned to test means in the BM3 prediction TSDs were based upon historical experiences with similar projects. As an example, consider the case of short circuit faults:

> INSPECTION - detection 65%, isolation 95%, correction 0% - based on the assumption of a multilayer board and experience with inspection of similar boards. An estimate of the number of visible traces vs. the total number of traces is a good starting point for detection. The 95% isolation value is selected since detection implies that isolation has been accomplished. The 0% correction allocation is assigned here since correction is not a normal function of an inspector.
>
> BIST - detection 10%, isolation 20%, correction 0% - Values for this test means are low since the FPCTL board contains only 2 components (of 15) with an incompletely understood BIST capability. A reasonable assumption for is to assign the BIST values based upon the percentagefraction of circuitry which contains BIST. Correction

receives a 0% value since BIST will not change conditions causing a short fault.

BOUNDARY-SCAN ATE - detection 40%, isolation 40 %, correction 0% - BS-ATE is assigned a relatively high value for detection since BS-ATE can detect short faults as well as faulty components in the interconnection paths. The actual values are chosen by examination of the interconnection routing and an estimate of the percentagefraction of the board circuitry accessible to BS-ATE testing. Isolation is given the same value as detection since BS-ATE testing is intended to locate faults by proper choice of test vectors. BS-ATE does not perform correction, thus the 0% value for this transfer function.

NON-BS-ATE - detection 50%, isolation 50%, correction 50% - Non-BS ATE is testing on an external tester. The detection level possibleility here is related to the number of traces/components which are accessible via a test connection interface such as bed-of-nails. Isolation for non-BS ATE testing is assumed to be theat same as for detection. Correction is given a value of 0% for this test means for FPCTL.

MANUAL TEST - detection 50%, isolation 50%, correction 100% - Detection and isolation are assigned estimated values based upon the nature of the testing anticipated. Values assigned reflect the ability of a "typical" manual test to identify and locate a specific fault causing the problem reported by a previous test means. Manual test is expected to be used mainly as a path to **correction**. It is not intended to be a primary means for detection of faults because the **time consumed by a manual test as a primary test means for detection and isolation is long and such a test is costly**. Correction is assigned a value of 100% - otherwise the board under test is declared non-functional.

Similar lines of reasoning were used to assign other transfer functions used in TSDs. The resulting values were **assumption-based**. Their use will allowed reasonable allocation of expected values for detection, isolation, and correction events. The **assumptions used can be modified** to conform to a particular companyCompany's capability to execute the testing required. For example, if a company does not have non-BS ATE, it cannot execute such a test and either the available means must perform more effectively or other additional test means must be made available to replace the "missing" capability.

4.2.4.2 TSD Test Time/Test Cost Attribute Values

TSD Test Time/Test Cost Attribute Values Test time entries for TSD attributes are based primarily upon estimates of the time required for test means with generic characteristics to complete a specified test. BIST, for example, is typically completed in microseconds or milliseconds, and manual tests typically take tens of minutes to complete. Known values are used if available but this situation rarely occurs during a test design phase. The exact number of microseconds or minutes is not critical. The order of magnitude will categorize the time and cost for a test means. Most test cost/time analysies will thus be **assumption-based** and will use results reported by experienced test experts familiar with similar tests to generate cost/time attributes.

Test cost entries for TSD attributes come from estimate of the **fixed** and **variable** costs associated with testing. Fixed costs are based on the cost of test equipment needed to test all boards produced, fixturing costs and nonrecurring labor cost for test development. Variable test costs are associated with technician labor required to perform tests. The amount of labor required is directly related to the test time discussed in the previous paragraph.

4.2.4.3. TSD Implementation

The TSD sets for BM3 were constructed as **EXCEL workbooks** designed to efficiently implement data sharing among all the two dimensional spreadsheets in the set. Each workbook consists of a set of worksheets which details the four classes of TSDs for one particular flaw or fault.

Three worksheets form a set capable of controlling/monitoring of a specific flaw/fault. A separate workbook is assigned to each flaw and fault for which requirements have been established. Individual worksheets are assigned to requirements, predictions, verification, and measurement related to the flaw/fault. For any one complete workbook, the number of worksheets will be up to 12. At system design levels, fewer sSheets will be needed since verifications and/or measurements may not apply.

During the system concept and design development phases of a project, the requirements and prediction worksheets are populated. Verification and measurement worksheets will be populated as the project proceeds into the actual design and fabrication phases.

Figure 4-4 identifies the contents on the worksheets for one TSD class (requirements, prediction, verifications, or measurements) in one EXCEL workbook. For the sake of analysis, a **Rrequirements TSD set** will be discussed.

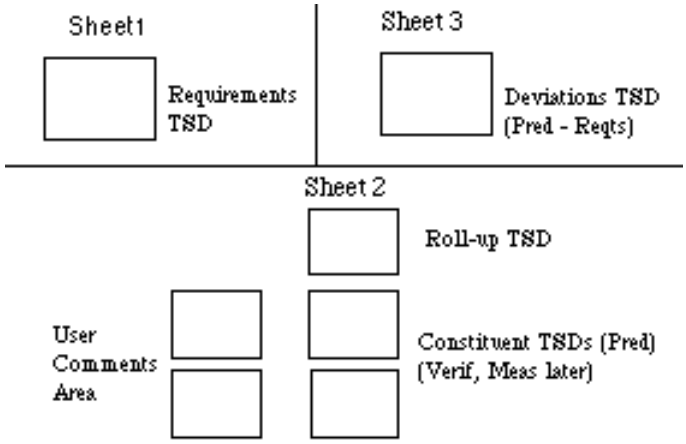


Figure 4 - 4: Worksheets for an EXCEL Test Strategy Diagram set.

Sheet 1 contains a flaw/fault specific TSD which states the **requirements** for testing related to that flaw/fault at some level of packaging. On this sheet, the user will enters the transfer functions, cost and time allocations appropriate to the flaw/fault being processed.

Sheet 2 contains a Rroll-up **requirements** TSD for a specific flaw/fault and the allocation of these requirements among the contributing lower packaging levels. For example, a TSD describing **subsystem** requirements will have **board** components. Downward allocation ensures that each lower level packaging level is examined to ensure that it will satisfy requirements which must be met in order that the requirements of the higher assembly be satisfied. The roll-up is the topmost section of Sheet 2. The roll-up TSD is automatically created from summing the contributions from the individual lower level TSDs.

TSDs at different packaging levels are coupled by EXCEL intersheet and inter-workbook communication. Thus the roll-uprollup TSD from a lower level TSD can function as a complete Sheet 1 entity at some higher packaging level.

Sheet 3 contains the differences between the **requirements** stated on Sheet 1 and the **roll-up section** of Sheet 2. Sheet 3 entries are **automatically generated** by intersheet subtraction. During system design phases, reported differences between requirements and predictions could suggest that there could be a problem with the requirements.

Similar sets of TSDs for the same flaw/faultcan be constructed to specify reporting means for **verifications and measurements**. The TSD set thus serves as the common connection link among requirements, predictions, verifications, and measurements and specifies values for detection, isolation, and correction for each of these categories.

4.2.5. Test Pprocedure Development

Test procedures are defined once the physical test setup has been established and a test plan has been designed. Both the plan and procedures are outputs of **test architecture** (TA) design and TSD construction. The TA is constructed simultaneously with the TSDs since any evolving test strategy needs an infrastructure which can support the tests called for in the TSD. In particular, all testbenches must be built using test features which will be physically realizable. BIST and JTAG test elements become embedded into testbenches and the extent to which these features are used impacts the entire test environment. Testability features themselves are tested during simulation. BIST and Boundary Scan testing are desirable but they are only supportable by designs containing BIST circuitry or JTAG-compliant components.

Also, any testing requires access to the physical testers required to perform the tests. If expensive ATE is currently not available, either it must be acquired or the test strategy calling for its use must be modified. For any company, a one-time analysis of facility test equipment in terms of expected project needs should be performed to ensure that the test architecture required to support test plans is appropriate. The analysis for BM3 generated a list of test means available to Lockheed Martin to evaluate different potential equipment anomalies. Tests selected for the STP were chosenselected from this list based upon availability of specific test means.

When all the details of test architecture are resolved, a test plan which orders application of test means in accordance with the TSDs is generated. For the BM3 exercise, BIST and Boundary Scan test were used as primary test means even though the FPCTL board was not built using a majority of BIST or JTAG components. (Customer requirements demanded use of these test means.) BIST and Boundary Scan test were augmented with the use of a wide range of test means at the disposal of the LM facility. Test procedures were developed in template form to provide test personnel with a well defined test sequence and a statement of all required test data.

4.2.6. Board Level Application of DFT Design Tools

DFT installation is supported by several tools which add DFT features to ato a design and evaluatesexaluate their effectiveness of the addition. Most of these were VHDL-based, which is in line with the use of VHDL as the basic RASSP modeling language.

A primary tool wais the VHDL simulator which produceds models and testbenches to add and evaluate the effects of DFT features. QuickVHDL and IKOS simulators were used to model BM3 hardware and testbenches. The former is a utility system and the latter can generate code which can execute on a general purpose processor or can be applied at high speed on a hardware accelerator.

Other tools, mostly VHDL-based, are used to install particular DFT features into a design as required or to perform testability analyses.- Automatic Test Pattern Generators (ATPG) such as VICTORY by Teradyne and board level BIST tools by LogicVision are examples which were used with BM3. The purpose of their application in the Shadow Program was to prove that the capability for installation of DFT features existed and was supported bywith a toolset that made DFT practical.

Another category of DFT tools which falls into the support category andwhich is usable even after the design process has been completed-. The Parallel Port Tester (PPT) by Teradyne, which is used to test hardware during both manufacturing and field support phases isphases.is an example. This tool uses a portable PC to apply test vectors, generated during the design phase to equipment in operational environments.

DFT design tools will be used as needed on a RASSP project. The nature of the specific hardware/software design will determine which design tools are applied. Some examples pertinent to BM3 arwill now be presented in the following paragraphs.

4.2.6.1. VHDL Component Modeling

One significant VHDL-based test component evaluation was conducted for a Texas Instruments Test Bus Controller (TBC) type ti8990 which is used to distribute JTAG test bus signals to multiple boards at the system level. In the modeling exercise, the TBC was used to activate a PseudoRandom Pattern Generator/Parallel Signature Analyzer (PRPG/PSA) pair to trigger a signature analysis test of a single logic block to be tested. Models (Smart Models) used for the TBC and the supporting ti8245 buffer PRPG/PSA were obtained from Synopsys Logic Modeling Group. A VHDL testbench was written to configure and control the TI Test Bus Controller device to simulate operation of a TBC under control of a host processor. The TBC communicated with a microprocessor and initiated a PRPG to supply codes to the logic block under test. The outputs of this logic block are passed to the parallel signature analyzer. The PRPG code sequence has a specific "correct" signature defined by the logic block which forms the basis for an evaluation of the logic block function. A correctly functioning logic block will always result in the generation of the proper signature.

TBC control of multiple circuit blocks was also simulated in a testbench to simulate testing of multiple boards from a central station. The FPCTL board of BM3 can use this mode for testing 2 FPCAP boards, and can, in turn, be tested by a higher level controller. The VHDL system model used a TBC to control three simple virtual boards. Virtual board 1 was simulated as a chain of PRPG, logic block, and PSA as described above. Two other boards were simulated by PRPG/PSA pairs. The testbench for this case sent parallel commands to the TBC which serialized the command in compliance with JTAG protocol and configured one ti8245 as a 16-bit PRPG and its receiving counterpart ti8245 into a PSA. The PRPG then supplies test vectors to the logic and associated PSA. Under control of the testbench, when the random pattern completes, the signature acquired in the PSA is shifted over the JTAG bus to the TBC which converts the serial message to a parallel data word and makes it available for further processing. Virtual boards 2 and 3 were sequentially configured by the testbench via the TBC as PRPGs and returned patterns via the JTAG bus to the TBC to verify how a microprocessor would use the TBC to control and access data from multiple boards.

The VHDL testbenches and model code used for the TBC analyses remain as legacy items for future designs.

4.2.6.2 Testability Analysis and Interconnection Testing

Once a preliminary design has been reduced to hardware, a testability analysis can be performed to determine the extent of external test support required during manufacturing and field deployment. For JTAG test strings, the tools VICTORY and VTM_TOP can perform accessibility analyses and generate test vectors which can be applied to the JTAG-accessible components. VTM_TOP is a program which takes a Mentor Graphics schematic and controlscontrolls passage through the steps required to execute VICTORY, a program which performs testability analyses and generates test vectors in serial vector format (SVF). The serial vectors are applied to the components in the serial JTAG string to implement an interconnection test. VTM_TOP generates a VICTORY data base from a Mentor data base. In addition to the serial test

vectors, a significant output of VTM_TOP is an accessibility analysis which reports inaccessible, partially accessible, and fully accessible nets. This report is useful in suggesting test points which must be made available to ATE testing because the nets are not JTAG-accessible. Extra pads can be added to a board design to support ATE testing.

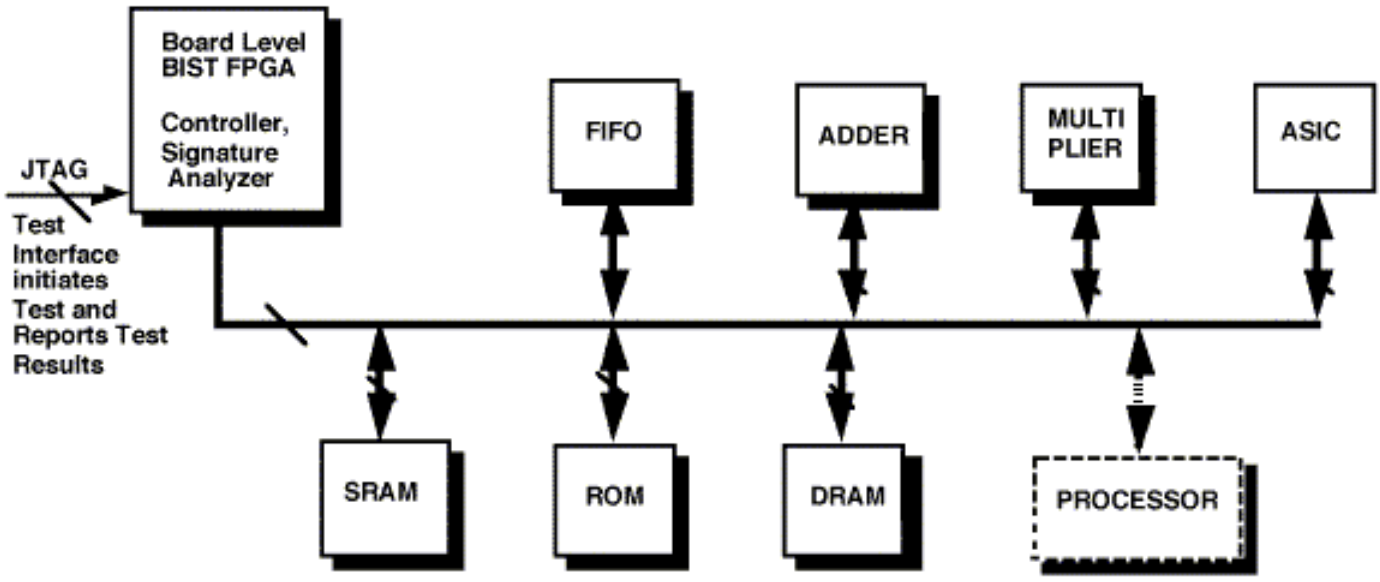
As part of its normal activity, VICTORY generates a Test access port (TAP) integrity test, which verifies that the JTAG connection path for the compliant board components can support a JTAG-based interconnection test. Part of this procedure is a verification of the boundary scan description language (BSDL) files for all components. These files which describe the JTAG features of each component in a VHDL-like language are obtained from component manufacturers and are critical to a successful JTAG interconnect test for the components.

Both VTM_TOP and VICTORY were applied to the BM3 FPCTL board, even though the JTAG string was very limited. All features of the tools were tested. The primary result was a demonstration of the accessibility analysis. Outputs from these tools will be significantly more important as the number of JTAG components on a board increase. In the limiting case of 100% JTAG component compliance, the SVF file will be capable of executing a full interconnection test for a board. Both VTM TOP and VICTORY are general purpose tools which offer considerable manufacturing test support and should be considered for use on any RASSP project.

A support tool which works with the SVF file generated by VICTORY is the Teradyne parallel port tester (PPT), a tool which accepts an SVF file and applies the JTAG interconnect test to a board under test via the parallel port of a PC. The PC-based test allows the interconnect testing to be executed during manufacturing and field support using only a PC attached to the board via a 5 wire JTAG test bus cable. In cases where the JTAG-compliant component complement is relatively high, the PC test can replace more elaborate equipment and provide testability which is not normally possible. The testing will take longer because of the serial vectors, but the test accessibility feature is a good trade. For BM3, PPT installed on a portable PC was used to evaluate the use of the 8245 buffer/PRPG/PSA device using VICTORY-generated SVF vectors. PPT was also used to test the DATA I/O board from the RASSP Benchmark 2 project.

4.2.6.3. FPGA-based Board Level ABIST

A VHDL model and testbench used with BIST tools showed feasibility of introducing FPGA-based memory ABIST using the FPCTL board as an example. The memory ABIST application used a modified version of the LogicVision ICRAMBIST tool, designed to implement on-chip memory BIST for ICs, to demonstrate the concept of using a reconfigurable FPGA to perform **complete, at-speed** testing of external memory (RAM and similar devices with regular structure not JTAG-accessible) arrays. Figure. 4-5 illustrates the concept. An FPGA is configured as a component-specific memory tester activated via a JTAG test bus interface also built into the FPGA. BIST is triggered and the PASS/FAIL result is returned via command received over the JTAG bus inas a standalone test or as another "component" in a JTAG string. The memory ABIST test requires that each component to be tested must be separately accessible on the bus. This is the case for most boards which use Output Enable (OE) signals to select components to be addressed by a processor.



FPGA-based board level BIST provides controllability and observability of Non-compliant, JTAG, regular-structure CO

Figure 4 - 5: FPGA-based board level BIST.

If memory components can be tested independently, the need for a software-based processor-controlled test that requires much of the board to be functional before memory testing can begin, is eliminated. A simulation (complete with testbench) of an FPGA which addressed and wriote/read data to each memory cell of a 256K x 4 static RAM to validate its storage ability was successfully executed. This is a promising new technique which can extend test coverage to complete arrays of components which have traditionally not been independently testable. Another desirable feature of FPGA-based board level BIST isalso has the desirable feature that the same testing is available for both manufacturing and field testing. LogicVision has announced a commercial product (memBIST XT) which is capable of installing an FPGA-based memory test capability at board level.

LogicVision also created other board ABIST tools during their participation in the RASSP program which were used to perform "what if" analysies on the BM3 controller board. The tool was used to investigate the fault coverage improvement resulting from replacement byof JTAG compliant components on the board in place of those actually used and test coverage effects of strategic JTAG buffer insertion. The BM3 controller board was a good example for analysis but a poor candidate for improvement since only 2 of 15 components had boundary test capability and 2 others had BIST capability. No significant improved coverage could be achieved for the hardware already constructed to meet project timing constraints. Future versions of the hardware could benefit from the analysis, however, by substitution of JTAG-compliant components, where possible, for the current components and the addition of JTAG buffers.

4.3. Technical Summary of Application Example

The shadow DFT application effort for BM3 resulted in a full implementation of the RASSP DFT methodology. It demonstrated the procedures involved, exercised the full range of tools available, and produced design recommendations which could improved the testability of the hardware produced. The work produced a legacy of DFT-related templates usable on future projects. Notable among those items produced are: the requirement template and the EXCEL TSD files; a method of inserting FPGA-based board level BIST for memory structures; and the tools to accomplish the insertion.

4.3.1. Tool Application Summary

The nature of the DFT problem and its wide scope of applicability dictate that most of the tools bare **separately** applied during the appropriate project phases. Dependency modeling, for example, supports early system design and architecture comparison and selection as an isolated tool application. DM can be used at lower packaging levels but is not typically employed there in the RASSP DFT methodology. DM outputs, however, can serve as starting points or checkpoint references for other processes such as test strategy generation. An "intermediate" level of tool interaction is represented by the filling in of **TSDs** from the **consolidated requirements**. The outputs of the requirement templates are used directly to supply the ordering of test means and fault coverage parameters for the TSDs. Maximum tool interaction occurs in the area of simulation and DFT feature addition such as that provided by the LogicVision suite of board level DFT and BIST tools. VHDL is a common thread between simulation and these tools. Other supporting tools such as VICTORY, an interconnect testability analyzer, also support the simulation phase by generating inputs such as test vectors for given arrangements of components.

4.3.2 . Reuse Element Listing

The Experience with BM3's shadow program effort provided many reusable procedures and templates which apply to many signal processor system designs. This reusability results from the similarities inherent in such systems such as PC board or MCM construction and bus oriented architectures. The following list identifies reuse elements.

- Requirements consolidation procedure and templates
- EXCEL spreadsheet TSD templates
- Analysis of Company-specific testing capabilities
- VHDL models for JTAG components
- Tool validation examples
- Methodologies for economic analyses

4.3.3 . Lessons Learned

The BM3 experience also yielded a significant number of "lessons learned" which provide direction to future applications of the RASSP DFT methodology. The following recommendations will reduce the DFT insertion effort and aid in the realization of the potential technical and economic improvements DFT offers.

- Requirements consolidation is the foundation of RASSP DFT insertion. Spare no efforts in generating an acceptable set of consolidated requirements.

- Templates save considerable time and eliminate sources of error sources and omissions.
- Functional dependency modeling should be used as early as possible.
- Maximize the roles of BIST and Boundary Scan devices as test means.
- Maintain detailed records to serve as future reuse elements.
- Develop an extensive VHDL model library to support a wide range of signal processor designs. Include JTAG and reconfigurable FPGAs as parts in that library to support DFT insertion.
- FPGA-based board level BIST will considerably enhance board testability whenever bus connected memory arrays are used.

4.4. DFT Economic Analysis

RASSP DFT efforts primarily addressed technology. However, since economics can never be disregarded, consideration was given to the effects of the DFT methodology on project cost from several viewpoints.

- First, economics was treated from a generally by examining the contribution of DFT to the life cycle cost (LCC) of the BM3 project.
- Second, economics was examined from a project point of view to deterexamined the influence of DFT on each of the project life cycle phases.
- Third, an economic analysis sought to demonstrate how the Singular Test Philosophy with priority given to BIST/BScan test, contributed to cost savings by placing emphasis on the earliest application of lower cost means - specifically BIST and Boundary Scan as opposed to ATE.

The general conclusion was that DFT inclusion can result in savings in each of the project phases, with DFT-based cost savings increasing as the project life cycle moves from beginning to completion.

The methodology used for determining LCC reduction effects associated with RASSP DFT was to define an assumption-based model derived from experience and outputs from existing DFT application studies. For example, an economic analysis specifically targeted to BM3 quantitatively compared detailed estimated costs for manufacturing quantities of boards tested with BIST/BS and ATE test means. Test costs for **each** of the project phases **before (using conventional test methods) and after application of the RASSP DFT methodology** weare computed. This approach presents a methodology which can be adapted to particular circumstances by changing the assumptions appropriately. For BM3, analysis began with assumption of a model for the distribution of LCC :

- Design Phase 10%
- Manufacturing Phase 35%
- Field Support Phase 55%

This distribution model reflects historical costs incurred for a "typical1" project requiring fabrication of a large (1000s) number of boards. A three step before/after/compare procedure was applied (see BM3 Shadow Report Summary for basis of assumptions) to estimate the effects of testing on a system with this LCC distribution.

Step 1. Estimate pre-DFT contributions of testing to LCC.

- Design phase : Assume 30% of design costs are dedicated to test development. This represents 3% of LCC.
- Manufacturing phase : Assume that 10% of LCC results from manufacturing testing.
- Field support phase : Assume that 3% of LCC is TPS re-engineeringreengineering cost and that 1.4% of LCC is consumed by testing of spares at the time of their manufacturing.

When these quantities are summed, the testing portion of LCC attributable to testing is 17.4%.

Step 2. Impose RASSP DFT methodology and reevaluate effects of testing on LCC.

- Design phase : Assume 20% additional effort added to test portion of design to install BIST and other DFT recommendations. **This increases LCC to 3.6% from 3%.**
- Manufacturing phase : Assume that reuse of design TPS elements and introduction of a BIST/BS architecture to the maximum extent reduces manufacturing costs by 62.5%. **The pre-DFT LCC of 10% is reduced to 3.75% from 10%.**
- Field support phase : Assume the 3% LCC cost for TPS re-engineeringreengineering and the 1.4% cost of spares testing can be reduced to 10% of the pre-DFT values by reuse of TPS elements developed during the design phase. **Thus, the post-DFT LCC consumed in field support is 0.44%.**

When Tthese quantities are summed, the portion of LCC attributablest-DFT cost tof testing is approximately 7.8% of LCC cost.

Step 3. Draw conclusions.

Several significant conclusions can be drawn by a before and after comparison of test costs in steps 1 and 2:

- pre-DFT LCC from test / post-DFT LCC from test = 17.4%/7.8% = 2.3X
- DFT can reduce test costs associated with manufacturing by 2.7X
- Potential absolute reduction in LCC from direct DFT introduction equals 17.4% - 7.8% = 9.6%. Indirect costs resulting from lower spares requirements can further reduce LCC by an additional 6%. Cost reductions for repair labor could save another 5% LCC.
- Total potential absolute reduction of LCC > 20% (approximate).

To verify that DFT offers savings throughout the product life cycle **Potential DFT savings at each project phase** were also examined to show that DFT offers savings throughout the product life cycle. A summary of the application of DFT to each project phase and thealong with DFT-related potential cost differences is summarized in Table 2.

Table 4 - 2: Phase-Specific DFT Economics

W/O RASSP DFT	W/RASSP DFT	Cost Difference Estimate
Design Phase		
System Design	HW/SW Codesign	+10%
Functional Design	VP Design/Test Analysis	+10%
Prototype Fab	VP Testbench Design	-10%
Prototype Test	VP Simulation Execute	-10%
HW/SW Integration	(Part of Codesign Effort)	-
System Debug	Functional Simulation	-20%
Prototype Mods	VP Model/Testbench Mods	-20%
Retest	Rerun Simulation	-10%
The potential savings (sum of the cost difference estimates) during design resulting from effective DFT installation can be up to 50 %.		
Manufacturing Phase		
Inspection	Inspection	-
Manufacturing Proto Fab	Manufacturing Proto Fab	-

ATE Setup	BIST/BScan Setup	-200%
Generate Final TPS	Reuse Design TPS	-100%
TPS Validate	Reuse Design TPS	-
Board Test	Board Test	-50%
Retest	Retest	-50%
The potential savings during manufacturing can be up to 400% if effective DFT is built into the product. The main source of savings is in reuse of the TPS generated as part of the design effort. As expected, the manufacturing phase benefits considerably from DFT.		
Field Support Phase		
TPS Re-engineeringReengineering	BIST/BScan Reuse	-1000%
Spares Test (Depot)	Spares Test (Local)	-100%
Repair/ATE-based Retest	Repair/BIST/BScan Retest	-50%
The potential differences for field support costs with and without DFT are very large(1150%) due to the known high cost of Test Program Set Re-engineering.		

4.5. Application Example Conclusions/Summary

Application of the RASSP DFT methodology to BM3 has demonstrated that **DFT can beis a significant technical and economic contributor** to any signal processor design project. It resultdevelops in well defined requirements which lead to production of equipment which is economical to manufacture and later supportable in the field, both of which lead to customer satisfaction. RASSP DFT asserts a positive influence during all RASSP project phases - an influence that will be enhanced by **reusing** experience acquired on previous projects - which result in phase-specific technical improvements and economic savings. The DFT methodology supplies functional dependency modeling which aids in **architecture selection** and the, definition of a suite of tests that are supportable with existing test facilities., It andlso provides for the creation of designs and the test plalns which support testing of very high density packaging approaches which are becoming **untestable** by conventional approaches. The RASSP DFT methodology has created a complete - across all phases - solution to testability for RASSP projects that results in improved hardware/software systems and more effective use of the funds required to build therse systems. The RASSP DFT effort work has contributed advancements to design and simulation tools advances and has developed **new techniques** which ensure testability of current and future systems.

Next Up Previous Contents

Next: 5 References **Up:** Appnotes Index **Previous:**3 Technology Description

Page Status: in-review, January 1998 *Dennis Basara*

RASSP Design For Testability Application Note

5.0 References

RASSP Methodology Document, Version 2.0, Volume 1, Lockheed Martin Advanced Technology Laboratories, October 1995. [METHODOLOGY_95]

RASSP Design for Testability (DFT) Methodology document, Version 1.0, Lockheed Martin Advanced Technology Laboratories, September 1995. [METHODOLOGY_DFT95]

Evans, J. S., and R. M. Sedmak, " A Hierarchical, DFT Methodology for RASSP ", Journal of VLSI Signal Processing 15, 1997. [EVANS_97]

Evans, J. S., P. McHugh, and R. M. Sedmak, " Integration of DFT into RASSP", Proceedings 2nd Annual RASSP Conference, Arlington, Va. July 1995, pp 217-222. [EVANS_95]

Evans, J. S., S. Sharma, R. J. Tarzaiski, and R. M. Sedmak, "Tutorial on RASSP Design-For-Testability", Beta Site Tutorial Powerpoint presentation, Lockheed Martin Advanced Technology Laboratories, Camden, NJ, March 1996.

[TUT_INTRO_96] - Introduction to the problem of testing across design, manufacturing, and field deployment. Describes DFT solution, RASSP DFT goals, and the integration of DFT into RASSP.

[TUT_DFTMETH_96] - Tutorial steps for application of DFT Methodology, including test strategy diagrams (TSDs). Part of a training session for prospective users of the Methodology.

[TUT_TESTARCH_96] - Turtorial describing the RASSP overall test architecture with components testbench architecture, testability architecture, and tester architecture. The relationship of these components to the test strategy diagram is examined. The complete spectrum of testing from requirements specification to development of test plans and procedures is presented.

[TUT_METHEX_96] - Detailed application of the RASSP Methodology to the Benchmark 3 systems as part of a non-interfering shadow program. Develops TSDs, a singular test philosophy, and describes a test means evalustion. Also describes the role of dependency modeling in high level system analysis.

Tarzaiski, R. J., and S. Sharma, "Design For Test Methodology Applied Across Design and Support Cycles", RASSP Review Powerpoint presentation, Lockheed Martin Advanced Technology Laboratories, Camden, NJ, November 1996. [TARZAISKI_R96] - A complete description of the results of the LM DFT methododlogy presented at the Program Review of November, 1996. The presentation was repeatedly given during peration of a display booth where all RASSP achievements were highlighted.

Test Strategy Diagram Example (Subsystem Level), Excel 4.0 workbook, November 1996. [3dm2fe.xls] - An EXCEL implementation of the subsystem level test strategy diagram for the Benchmark 3 Functional Element consisting of one FPCTL board and two FPCAP boards. Requirements, predictions, and differences between these quantities are entered into cells for later distribution to lower level TSDs.

Test Strategy Diagram Example (Board Level), Excel 4.0 workbook, November 1996. [3dm2bds.xls] - An EXCEL implementation of the board level test strategy diagram for the benchmark 3 boards. Requirements, predictions, and differences are described. The TSD automates the interrelationships among the established sequence derived from the singular test philosophy. Test means for the boards and their capabilities of error detection, isolation, and correction are described as cell entries and summaries for higher level application of these resulat is automatically coupled to the next higher level TSD.

Page Status: in-review, January 1998 *Dennis Basara*