

Integration of DFT into RASSP

John Evans
Lockheed Martin Advanced Technology Laboratories
jevans@atl.ge.com

Richard M. Sedmak
Self-Test Services
rsedmak@omni.voicenet.com

Patrick McHugh
Army Research Laboratory
MCHUGH@monmouth-etd11.army.mil

Abstract

This paper describes the integration of Design-For-Testability (DFT) processes, tools, and methodologies into the RASSP System, which consists of the Design Environment and Enterprise System. The blueprint for the DFT developments is the DFT Methodology, which is highly automated, hierarchical, and spans the entire life cycle, contributing significantly to the RASSP goals of 4x improvement in cycle time, design quality, and life cycle costs. Key concepts of the DFT methodology are covered. A preferred testability architecture that encompasses embedded test resources (BIST), external test resources (ATE), and testbenches for design verification is described. Integration of the DFT methodology and testability architecture into the RASSP system is covered. The paper concludes with a discussion of the contribution of DFT to meeting the RASSP goals.

Key words: BIST, design-for-test, rapid prototyping, reuse, RASSP, signal processors, testability architecture, test metrics, test strategy diagram

Introduction and Review of the RASSP Program Goals

The primary goal in the RASSP program is to provide an improvement of at least a factor of four in the time required to conceptualize, design/ upgrade, and field signal processor designs, with similar improvements in design quality and life cycle cost. To achieve this goal the Lockheed Martin Advanced Technology Laboratories (ATL) team has developed an overall design methodology. The overall

RASSP methodology emphasizes the following approaches:

- Concurrent engineering
- First-pass success
- Reuse designs and information
- Providing tools for automation and high-level synthesis

The Design-For-Testability (DFT) methodology emphasizes these approaches also. In addition, the DFT methodology allows for the hierarchical development of DFT and supports feedback of in-process test data from production and field for incorporation into subsequent model years and new designs. Furthermore, it is not driven by existing CAD tools but does use knowledge of tools to assess the practicality of implementing the methodology.

Overview of the RASSP DFT Methodology

The RASSP DFT Methodology enables designers to create systems that can be cost-effectively tested throughout their life cycles. Designs that adhere to this methodology are made testable on the basis of various DFT and built-in-self-test (BIST) techniques. The methodology, as shown in Figure 1, covers all aspects of test and diagnosis at the system, board, MCM, and chip levels — including test requirements capture; test strategy development; DFT and BIST architecture development; DFT and BIST design and insertion; test pattern generation; test pattern evaluation; and test application and control. The designer is provided a process for introducing testability requirements and constraints early in the design cycle and for addressing DFT and BIST issues hierarchically at the chip, multichip module (MCM), board,

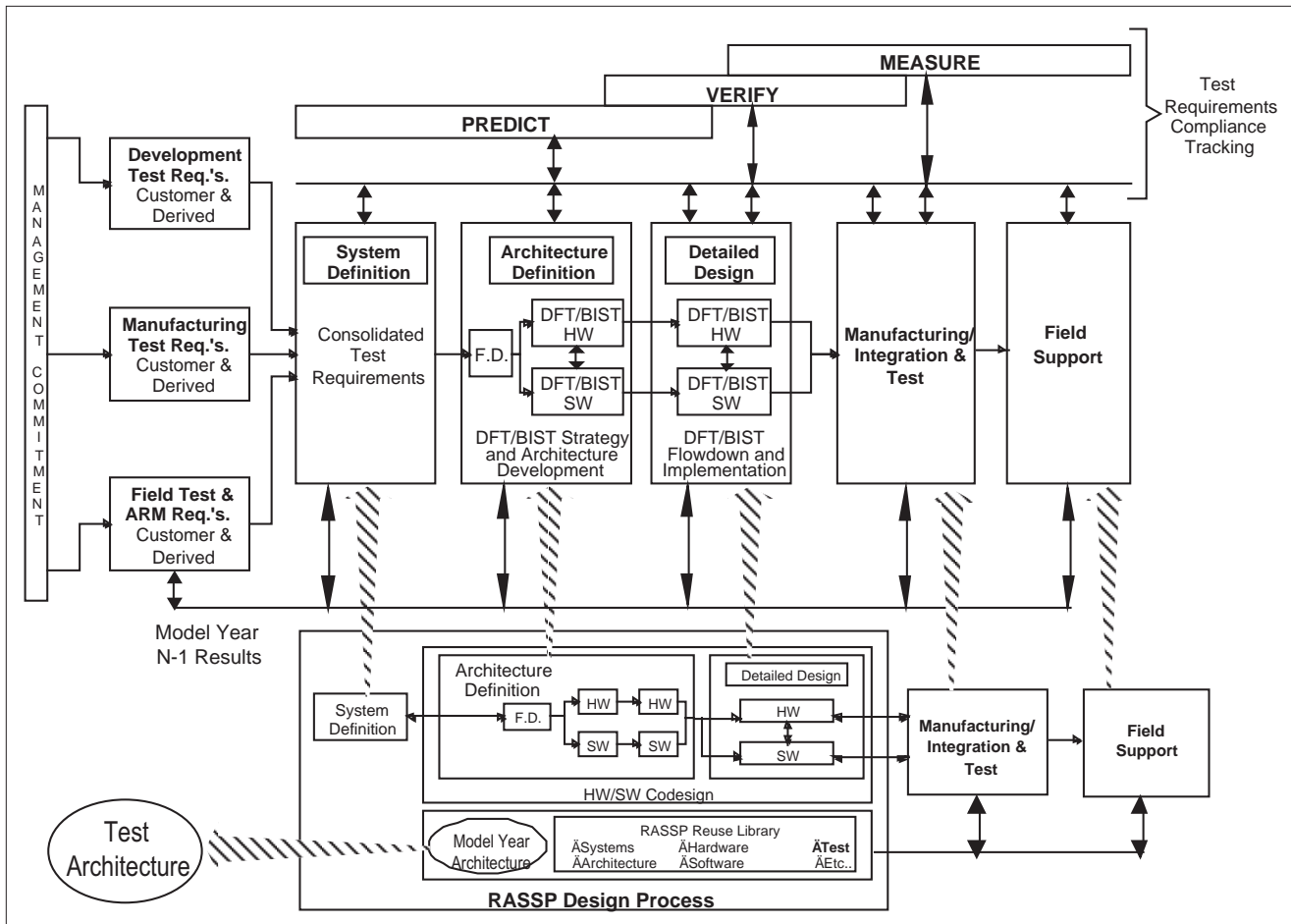


Figure 1. RASSP DFT methodology in the overall methodology.

and system levels. The payback for early testability emphasis includes lower test cost throughout the life cycle of the product, reduced design cycle time, improved system quality, and enhanced system availability and maintainability. Key features of the approach include:

- A life-cycle view to used to unify requirements and solutions across design verification, manufacturing test, and field support
- Quantifiable testability metrics are used to conduct tradeoffs and to provide consistent tracking mechanisms
- Test strategy diagrams are used to predict, verify, and measure compliance to requirements and to flow test strategies down across the system hierarchy
- Reuse of test objects or resources across model years and product lines, throughout the product life cycle (design verification through field support); throughout the packaging hierarchy; and within a given level of the packaging hierarchy.

In Figure 1, an overview of the methodology is reviewed. The methodology begins with a tangible management com-

mitment. This tangible commitment provides the budget and resources to proceed with the methodology.

The second step, System Definition, involves test requirements specification. The test requirements come from an integration of customer and derived requirements for the three phases of testing: design, manufacturing, and field support. Specification involves a preliminary life-cycle cost of test analysis, a test technology assessment, and a design impact analysis to determine the reasonableness, consistency, and validity of the requirements. Subsequently, during the same step, the three sets of requirements are consolidated into a consolidated requirements specification.

The Architecture Definition phase consists of three steps: functional design, architecture selection, and architecture verification. (Architecture selection and verification are shown as one step in the summary flow chart of Figure 1). In the functional design step, the consolidated test requirements are used to develop the top level test strategy for the design, manufacturing, and field test phases.

In the architecture selection step, the top-level test strategy is used to develop and evaluate various candidate, top-level test architectures. The impact of each test architecture on the candidate functional architectures is assessed and incorporated into the tradeoff and selection process for them. The test requirements, test strategy, and test architecture are then allocated to BIST/DFT hardware and software for one or more of the selected architectures. In addition, candidate DFT and BIST techniques are identified for later implementation, based on the specified requirements. Also in this step, any top-level BIST supervisory software development will begin, as will on-line BIST code, because it may have an impact on performance and throughput. Prediction and verification processes begin in this stage as appropriate for compliance tracking.

In the Architecture Verification step, the next level of detail of the selected test architecture(s) is generated and additional details are provided regarding the test architecture impact on the selected functional architecture(s). For example, behavioral and performance simulations will include effects of DFT/BIST techniques, such as the estimated performance degradation due to hardware concurrent fault-detection circuits or due to periodic execution of on-line BIST software diagnostics. Prediction and verification processes continue during this stage for compliance tracking.

In the Detailed Design phase, test strategies, architecture, and requirements are flowed down to the detailed design of BIST/DFT hardware and software. The detailed design of the BIST/DFT hardware is performed concurrently with functional design using automatic or manual insertion and then is reflected into behavioral and structural simulation models, whenever possible. Any remaining BIST software (e.g., for power-up or other off-line BIST functions) is implemented. Test vector sets are developed and verified for each packaging level for physical prototype test, production test, and field test. All test vector sets are documented using WAVES. Prediction, verification, and measurement processes are used in this stage for requirements compliance tracking.

In the Manufacturing phase, functional and performance testing of the overall prototype is performed, with DFT and BIST hardware and software included. Production test is performed. Verification and measurement processes are used in this stage for compliance tracking. Test cost, manufacturing-defect profiles and statistics, and performance data are captured and encapsulated for the reuse library.

In the field phase, BIST and DFT capabilities are in use. BIST and DFT functions are also used for lower level (e.g., organizational and depot level) testing. Verification and

measurement processes are used in this stage for compliance tracking. Test cost, field defect profiles and statistics, and performance data are captured and encapsulated for the reuse library.

Throughout the entire DFT methodology, interfacing is done to the RASSP reuse library to access existing candidates and to add to the library, when appropriate. In addition, feedback is continuously provided from the compliance tracking process back to the responsible persons to assure corrective action is taken.

The RASSP Testability Architecture

There is a close relationship between the methodology document and the RASSP Testability Architecture. While the DFT Methodology prescribes DFT processes, the Testability Architecture prescribes a structured architecture based on a hierarchy of built-in, self-tested components connected by independent (from functional) Test and Maintenance (T&M) busses. The prescribed architecture is compatible with and can be derived from the DFT Methodology.

The prescribed RASSP testability architecture (Figure 2) relies heavily on the use of BIST and IEEE 1149.1 and P1149.5 T&M busses — such as 1149.1 boundary-scan incorporated at the IC level and 1149.5 at the backplane level. Incorporation of COTS designs (chips and/or multi-chip assemblies) are accommodated by either selecting COTS designs that already have testability incorporated — such as BIST and/or 1149.1 — or by making use of testability features in the circuits that surround the COTS designs. Test-related communication across the hierarchy of packages is accomplished through a system of hierarchical test and maintenance controllers and busses.

Integration Within the RASSP System

The RASSP program will deliver an integrated system called the RASSP system, which integrates the CAD tools used in the RASSP design process under a framework referred to as the enterprise framework. An *enterprise framework* provides the facilities and services necessary to integrate the automated processes of an enterprise.

In the RASSP system, the enterprise framework provides support for workflow management; design data management; library management; computer-supported collaborative work; and remote tool access. The *workflow management subsystem* of the RASSP enterprise system enables a RASSP system administrator to model and enforce a particular design methodology for a project. The *data management subsystem* of the enterprise framework provides

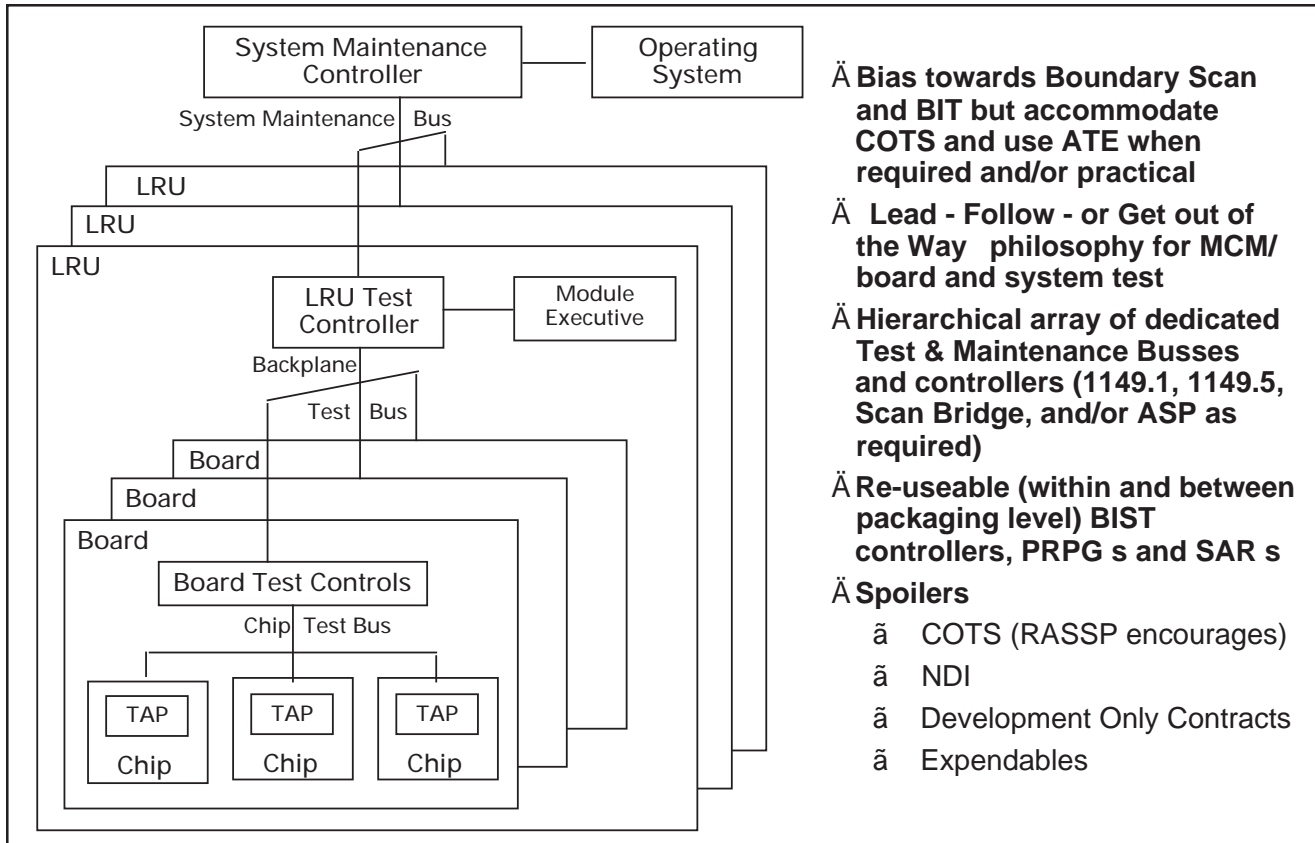


Figure 2. Recommended testability architecture.

- Ä Bias towards Boundary Scan and BIT but accommodate COTS and use ATE when required and/or practical
- Ä Lead - Follow - or Get out of the Way philosophy for MCM/ board and system test
- Ä Hierarchical array of dedicated Test & Maintenance Busses and controllers (1149.1, 1149.5, Scan Bridge, and/or ASP as required)
- Ä Re-useable (within and between packaging level) BIST controllers, PRPG s and SAR s
- Ä Spoilers
 - ã COTS (RASSP encourages)
 - ã NDI
 - ã Development Only Contracts
 - ã Expendables

facilities for configuration managing and controlling access to design data files that may reside at various sites in a computer network. *Library management* in the RASSP system involves the release, cataloging, and searching of reusable design components.

Integration of DFT within the Enterprise System involves:

- a. Capturing DFT process steps within the workflows and activity definitions.
- b. Encapsulating DFT tools within the Enterprise Integration Framework.
- c. Defining, selecting, developing (if required), and integrating initial items for the DFT reuse library elements.
- d. Defining (and/or selecting), developing, and integrating templates and standards for test-related product data information for documentation and manufacturing release.
- e. Training design teams on the methodology and testability architecture.

Lockheed Martin ATL is currently initiating these activities within its program. The program plan calls for working prototypes and baseline data packages ready in time for use by benchmark and demonstration design teams during

the Spring of 1996. Implementation of the DFT components will be according to the blueprints, the DFT Methodology Document, and the Testability Architecture Description. The ATL RASSP newsletter, *Enterprise*, describes the roles of the ATL DFT teammates.

All of these activities are important and will result in significant enhancements. Implementation of the DFT reuse library elements is highlighted because of the strong emphasis within RASSP on reuse. The RASSP library management system is referred to as the RASSP Reuse Data Manager (RRDM). The RRDM stores descriptive data about all the reusable components released by the CAD tools. The user of a CAD tool invokes the RRDM from the CAD tool and locates a reusable component by running a query on the descriptive data stored about the reusable components. The user may then view the reusable component using a standard viewer or a viewer specific to the tool that created the reusable component and import the component into a design object.

The reusable components are stored in the format native to the tool that created it, and possibly in standard interchange formats (VHDL, Ada, EDIF, etc.). The design data about the reusable component is stored within the environ-

ment of the tool itself, while the descriptive data of the component is stored within the RRDM. The RASSP Enterprise Product Data Manager (EPDM) manages and controls the access to the design data files of the reusable components. The descriptive data of the reusable components are modeled using classes of an object-oriented class hierarchy.

One of the key ways in which the DFT Methodology contributes to the achievement of the RASSP goals is through the enforcement of reuse of DFT in four different dimensions. The four dimensions of reuse are as follows:

- Across the life-cycle phases within a given model year.
- Across the packaging hierarchy within a given model year.
- Across a single packaging level within a given model year.
- Across new model years.

Test-related reuse items can include: test requirements; test strategies; DFT/BIST techniques for certain logic structures; testable chips; MCMs; BIST software modules; and test vector sets, for certain library elements.

Libraries are needed for reuse across model years and product lines. DFT reuse library elements for these needs are managed within the enterprise system by RRDM. Current component reuse library software applications, including RRDM, do not support the reuse proposed by the RASSP DFT methodology for:

- Reuse across the life-cycle phases within a given model year.
- Reuse across the packaging hierarchy within a given model year.
- Reuse across a single packaging level within a given model year.

Control and verification is required for these reuse elements, just as it is required for component library elements. Control and verification mechanisms are implemented by Test Strategy Diagrams described in the DFT Methodology document. Details of the relationships between TSDs and RRDM are being developed. The baseline concept is to encapsulate TSDs under the test DOCH sub-classes — therefore providing a uniform view of reuse at the enterprise level of components and DFT elements.

Contribution of DFT to Meeting the RASSP Goals

The DFT Methodology contributes to achievement of the overall RASSP goals in two ways:

1. The adoption of DFT practices results in reduced cycle-time, reduced cost, improved quality, predictable schedules (including integration and test), and improved time-to-market (as well as time-to-profit). For example, companies have seen 4-5X reduction in board test time by using boundary-scan-based testing. These benefits are realized because such practices facilitate debugging (helping both design and test engineers); hardware/software integration; transition of design data to test, test generation; and other test-related activities.
2. The structured DFT methodology provides improvement of the DFT process itself compared to current industry practice. This is achieved by introducing proven system engineering practices, such as the consolidation of test requirements. It is also achieved by leveraging the top-down development of the overall RASSP methodology to flow-down the test strategies and architecture from the system to chip packaging levels and across life-cycle phases of the product.

Specific contributions of the DFT Methodology to meeting the RASSP goals include:

- a. Promoting concurrent engineering by providing a specific methodology for integrating test with design activities. The methodology integrates tightly with the RASSP methodology and leverages top-down design, virtual prototyping and hardware/software codesign.
- b. Enhancing the probability of first-pass success by providing specific steps to ensure requirements are consistent, valid, and reasonable and by providing a structured process for flowing requirements, strategies, and architecture down to lower levels. The impact of errors on schedule and cost are minimized by incorporation of DFT/BIST, which detect, isolate, and correct as appropriate and/or required.
- c. Promoting a singular test strategy to reduce test development time and cost across the product life cycle. An example of this would be a PC-based boundary-scan test for design and manufacturing followed by reuse of the boundary-scan test in the field via embedded boundary-scan controllers.
- d. Leveraging reuse of any output of any DFT methodology step. Reuse for subsequent model years and other products is supported via the RASSP Reuse Data Manager within the RASSP System. In addition, a structured process and mechanism is provided for ensuring reuse of test resources within a model year:
 - Across the life cycle phases
 - Across the packaging hierarchy

— Across a single packaging level

Control and verification is required for these reuse elements, just as it is required for library elements. This control and verification is implemented by the Test Strategy Diagrams described in this document. The test strategy diagram method enforces reuse analysis and knits all of the DFT Methodology steps together.

- e. Providing a framework for codesign of DFT/BIST with functional hardware/software and tightly integrating with the RASSP methodology. The framework facilitates automation and high-level synthesis.

Conclusions

The Lockheed Martin ATL team has accomplished a major milestone by defining the blueprints for the integration of DFT within the RASSP system — the DFT Methodology and Testability Architecture. The efforts over the next year will be focused upon implementing the blueprints, as described in the *Enterprise* newsletter. The DFT team is coordinating efforts with the Benchmark and Demonstration design teams to provide examples and feedback on the selected approach.

Acknowledgments

The authors gratefully acknowledge the contributions of the members of the DFT Methodology Working Group, who provided reviews, comments, drawings, references, and numerous discussions: Vinod Agarwal, Larry Apfelbaum, Greg Barnett, Dave Brown, Joe Buemi, Paul Bompastore, Gerry Caracciolo, Jim Fasoli, Chris Flynn, Steve Fortier, Terry Frederick, Andy Hutchinson, Nick Kanopoulos, Jake Karrafalt, Bill Kline, Mike McCloskey, Mark Noneman, Ron Press, Mark Pronobis, Janusz Rajski, Shanti Sharma, Richard Sheldon, Jeff Stavash, Kermit Sunnay, Janet Wedgwood

References

1. RASSP Methodology, Version 1.0, Dec, 1994.
2. A Hierarchical, DFT Methodology for RASSP, Version 1.0, June, 1995.
3. RASSP Testability Architecture Description, version 1.0, July, 1995.