

HOW ARE YOU PLANNING TO VERIFY ALL THAT DFT?

STYLIANOS DIAMANTIDIS, GLOBETECH SOLUTIONS

```
vpi='h05
vci='h1020
hec='h39
src_port='h02
seq_number='h04
description="cellReceived"
error_count='d0
```



ABSTRACT

As gate counts continue to swell at a rapid pace, modern systems-on-chip (SoCs) increasingly are integrating more design-for-testability (DFT) capabilities. Test and diagnosis of complex integrated circuits (ICs) will soon become the next bottleneck, if, in fact, they have not already. With up to 30% of a project’s cycle already being spent debugging silicon, and typically 30–50% of total project costs being spent on test, DFT quickly is becoming the next wild card. As daunting a task as reining in all the variables related to DFT infrastructure can seem, an enormous opportunity awaits those ready to take up the challenge.

AND A CHALLENGE IT IS...

Today, DFT usually is nothing more than a collection of ad-hoc hardware put together by different people, using different tools, with neither a common strategy nor a vision of an end quality of result. The inability to deliver a reliable test infrastructure inevitably leads to missed market opportunity, increased manufacturing costs, or even a product that is not manufacturable. Instead, a carefully designed and verified DFT scheme, reflecting coherent test intent across the board, can be an excellent value differentiator throughout the lifetime of a product.

This brief article discusses how to plan DFT verification against test intent, ensure compatibility with standards and functional correctness, and create a complete, methodical, and fully automated path from specification to closure.

PLANNING FOR SYSTEM-WIDE DFT VERIFICATION

The foundation for systematic DFT verification is a well-defined set of goals, supported by a methodology developed to provide integration-oriented test methods for chip-level DFT, to enable compatibility across different embedded cores, and to incorporate high levels of reuse. A DFT-verification plan must satisfy these three separate objectives:

INTENT/SPECIFICATION

Does the test infrastructure adhere to the test strategy and specification set forth by the design and test engineers? You must verify that the global test intent is designed and implemented properly.

COMPLIANCE

Does the test infrastructure comply with industry standards for interoperability and universal facilitation of access? This is crucial to ensure reuse of hardware and software.

FUNCTIONALITY

Are there functional-design issues with the DFT resources? Although such resources may appear to operate within the parameters of the first two points, there could be logic bugs in the implementation.

Once the objectives are defined, the development of a complete system-level DFT-verification plan should follow these general steps:

1. Capture system-level test intent and translate it into an executable plan
2. Integrate heterogeneous core-level DFT plans into the system-level DFT plan
3. Provide a completely automated path from specification to closure
4. Provide quality-of-result metrics
5. Provide total-progress metrics
6. Integrate the DFT plan into the IC-level verification plan

CAPTURING SYSTEM-LEVEL TEST INTENT

In order to build a successful DFT-verification plan, you first must capture system-level test intent. During this step, test engineers need to work closely with verification engineers to ensure that the plan includes all the aspects of test that must be available in the final product. The global test-access mechanism (TAM) is the primary element at this point; however, other elements can come into play, such as top-level DFT features, integration to board-level testability, or hardware/software interfacing and information sharing.

Management must also be involved so that they gain an understanding of the implications and trade-offs of building reliable DFT. This will ensure total visibility and resolve contention for resources further down the road.

The preferable way to deliver the system-level test intent description is in executable form. The global verification plan must leave no room for doubt or misinterpretation. Furthermore, it needs to provide a solid basis for automation of subsequent steps down to design closure.

INTEGRATING HETEROGENEOUS CORE-LEVEL DFT PLANS

Bridging the gap between the ad-hoc world of spurious DFT resources and planned system-level DFT is not a trivial task. Individual intellectual property (IP) vendors' strategies for testability can vary significantly in terms of quality, coverage, and/or support deliverables.

In this phase of DFT-verification planning, it is important to work closely with vendors to align DFT strategies as closely as possible, and to enforce quality metrics. Optimally, vendors should work with their customers' test engineers to design pluggable DFT schemes and plans.

By capturing core-level test intent in an executable plan, and including it in the deliverables, IP vendors can provide a new added value to their customers. Such executable plans can then flow into the IC system-level test plan (see Figure 1). This is key to unlocking the paradox of driving a uniform SoC-level

test plan based on heterogeneous core-level DFT schemes from different vendors.

Finally, this methodology also needs to apply to internal engineering teams delivering design IP for integration. Such teams have different skills and management styles, and can operate in different geographies or business units. They too, must understand the need to plan DFT verification and provide the necessary components to enable this methodology.

PROVIDING A COMPLETELY AUTOMATED PATH FROM PLAN TO CLOSURE

Having a) captured the high-level test intent in an executable plan and b) integrated separate core-level DFT schemes, verification and test engineers are now empowered to drive their processes more effectively.

The result is a fully automated path from plan to closure for DFT verification, ensuring:

- **Completeness** — The verification plan includes a section on all DFT features and their specifics
- **Intent** — The verification scope has been defined early in the process by experts and with complete visibility
- **Uniformity** — Disparate test strategies can now be driven by a single process

During this stage, engineers should seek and incorporate various elements that will be used as building blocks to implement the verification strategy according to the plan. Such elements can include:

- **Verification IP**, used to run verification tests on individual DFT features
- **Test-information models**, used to exchange information with other tools and processes for enhanced automation

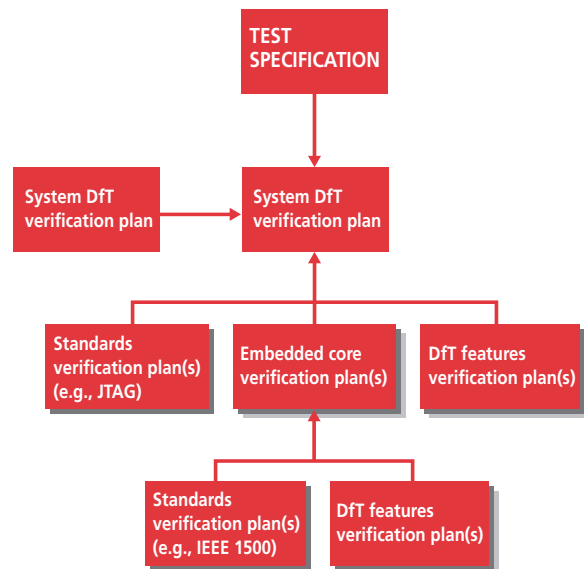


Figure 1: Simulation acceleration

PROVIDING QUALITY-OF-RESULT METRICS

But how can you conclude that the DFT-verification plan guarantees the necessary quality for reliable DFT? In order to address this inherent unpredictability, you must set expectations for quantifying results as part of your plan.

Quality-of-result metrics are measurable targets that can be entered as verification-plan attributes early in the planning phase. Such targets must result from collaboration among verification, design, and test engineers in order to ensure that all the aspects of the task at hand are addressed. They can include functional coverage metrics, such as the number of different instructions loaded in any given JTAG TAP, seed patterns used in automatic test-pattern generators (ATPGs), or isolation behavior of embedded-core scan cells. All these metrics should be associated neatly with a respective section of the executable test plan.

It is also a good idea to assign priorities or weights to different test-plan sections based on these metrics. For example, what is the purpose of exhaustively testing a built-in self-test (BIST) controller connected to a JTAG TAP if the TAP is not thoroughly verified first?

PROVIDING TOTAL-PROGRESS METRICS

Quality-of-result metrics can be a guide to understanding and reporting progress, and can identify critical paths. Once the project is underway, it is difficult to track the progress of specific tasks and the implications of prioritization. Tracking quality-of-result progress provides a way of correlating real DFT-verification progress while simultaneously enabling total visibility across the different teams. This way, test engineers can know at all times the progress of the verification of DFT across the board and use this information to drive other processes, such as test-vector generation or early fault analysis. They can also use this information to raise management awareness of issues that may arise during the design process.

INTEGRATING THE DFT PLAN INTO THE IC-VERIFICATION PLAN

Finally, a methodical DFT-verification plan must be integrated into the system-level IC-verification plan. This way, DFT quality-of-result metrics can be factored into chip-level metrics for total-quality management for closure. System-level planning should incorporate the processes and methodologies of effective DFT verification. This enhances the allocation of necessary resources, and ensures that expert knowledge is available.

Furthermore, a DFT-verification plan can help bridge the cultural gap that today divides test engineers from the rest of the design-cycle, and can advocate cooperation between the two main bottlenecks of today's SoC design: verification and test.

BENEFITS OF DFT-VERIFICATION PLANNING

There are a variety of motivating factors for planning and executing proper DFT verification. The investment made during the design cycle can be leveraged to reap a series of long-term benefits, including:

REAL DESIGN-FOR-TEST

Increased visibility into test intent across development teams results in better integration of the design and test engineering processes, skills, and cultures. Methodical plans to verify test infrastructures create a well-defined process for incorporating input from test engineers into the development cycle. Test engineers participate in creating the global test specification, helping to qualify vendors based on DFT-quality metrics, and/or prioritizing verification tasks against target results. This enhanced visibility also results in the reverse benefit of better communication and information feedback from manufacturing/test back to design in order to close the design-for-manufacturability (DFM) loop.

BETTER, FASTER, CHEAPER TEST

As semiconductor processes move deeper into nanometer scales, the cost of fabrication and test is exploding. Fabrication facility costs at 65nm are expected to hit \$4 billion. If the current test-capital-per-transistor ratio persists (it has been flat for 20 years), in several years the general cost of test will exceed fabrication cost. Associated low yield also increases the number of test cycles required to determine the quality of silicon.

DFT-verification planning aims to provide a reliable path from test intent to quality of results. Adding quality and efficiency to test planning leads to better testing strategies aimed at locating real silicon faults while minimizing costly over-testing and/or excessive vector sets. Testing on advanced automated test equipment (ATE) at 90nm can exceed \$0.10/second per unit; for a batch of 1 million at 100% yield, that's \$100,000/second. Improving the DFT-planning process can help companies make efficient use of this expensive tester time, or even help them to switch to cheaper ATE resources by partitioning more test resources on-chip.

DIRECT EFFECTS ON TIME TO MARKET

In a demanding consumer-driven electronics market, execution of a product strategy leaves no room for error. Re-spins simply are not an option when targeting 90nm (or below) process technologies. Lengthy silicon debugging, manufacturing-test time, low yield, and lack of diagnosability substantially impact the time-to-market window. Proper planning for DFT verification results in increased design- and test- schedule predictability and repeatability, better process automation, and enhanced efficiency with a direct, positive effect on time-to-market.

Designing verification IP that can be invoked directly and automatically from the plan results in additional, significant time savings. Such IP can include complete environments capable of generating test vectors, checking DFT state, and measuring the extent of exercise of the test infrastructure. Such IP should be designed only once for standard components (e.g. JTAG), and then enriched with feature-specific libraries for customization. Time invested up front results in overall project-time savings by ensuring that DFT is designed and verified only once. Such savings are optimized from project to project due to complete and calculated reuse.

BETTER VENDOR-QUALIFICATION METRICS

With third-party IP playing such an integral role in today's SoCs, DFT-verification planning can be used to increase levels of process integration and automation with strategic vendors. Furthermore, DFT-quality metrics can be incorporated in new vendor assessment by grading the vendor's test strategy, DFT implementation, DFT reuse, and applicability targets. Qualification metrics offer advanced vendors incentives to provide complete, executable verification plans, IP, and test information models for enhanced integration into their customer's test infrastructure.

CONCLUSIONS

Large and complex test infrastructures are a reality in today's dense SoCs, which comprise a multitude of diverse DFT resources. If companies are to meet their manufacturing-cost and time-to-market demands, they will need to ensure that such test infrastructures are well verified for specification, compliance, and functionality. At the foundation of the solution lies a detailed executable plan that can be used to provide an automated path from specification to closure with predictable quality of result.

How are you planning to verify all that DFT?