

Synthesis-Based Test for Maximum RTL Designer Productivity

November 2010

Author Introduction

Robert Ruiz
Product Marketing
Manager,
Synopsys, Inc.

Exponential growth in size and complexity of systems on a chip (SoCs), coupled with increasingly stringent quality mandates, necessitates an efficient and productive approach for register-transfer-level (RTL) designers implementing design-for-testability (DFT). Synopsys provides test technology embedded in synthesis, or “synthesis-based test”, to implement the key aspects of DFT for scan testing, boundary scan, embedded memory test, on-chip testing of high-speed blocks like USB and PCI Express[®] cores and connections to yield analysis. Synopsys synthesis-based test is the cornerstone of the most widely adopted DFT methodology. This methodology allows designers to achieve highly testable implementations with minimal impact on design goals and maximizes productivity. Synopsys provides the most current generation of synthesis-based test technology as part of DFTMAX compression and DFT Compiler. These products include capabilities to architect and verify DFT, including scan and compression logic, and also allow for correction of test rule violations prior to the final synthesis the design. In addition, they are integrated with the DesignWare STAR Memory System[®] for embedded memory test and repair, DesignWare high-speed SERDES such as USB and PCI Express[®] IP with built-in on-chip testing, and Yield Explorer for yield analysis.

With Synopsys synthesis-based test, DFT logic is synthesized directly from the RTL to testable gates with full optimization of design rules and constraints (Figure 1). This is possible because, unlike bolt-on test tools operating outside of the synthesis flow, DFT implementation and verification are performed directly within the synthesis environment, allowing problems to be found and fixed early in the design cycle. Area, low-power, and congestion optimization techniques are applied concurrently for the design and test logic. The final design generated with DC Ultra[™] synthesis is “ATPG-ready” with all test logic verified and test design rules checked, leading to very high and predictable test coverage results.

Synopsys synthesis-based test methodology is widely used in conventional ASIC and SoC design flows in order to keep up with today’s design complexity, size, and testability considerations. These design and test aspects must be addressed together throughout the entire design process rather than separately. To successfully meet all the design goals of these immensely complex devices, including function, timing, area, power and testability, swift convergence of all requirements must be simultaneously attained. Achieving successful closure of test requirements requires that RTL designers assume as much as responsibility as was previously owned by specialized DFT engineers. Synopsys’ test solution enables RTL designers to take on DFT tasks by transparently implementing compression, scan, and boundary scan as well as incorporating DesignWare IP with test during the design process, eliminating the need for highly specialized test knowledge.

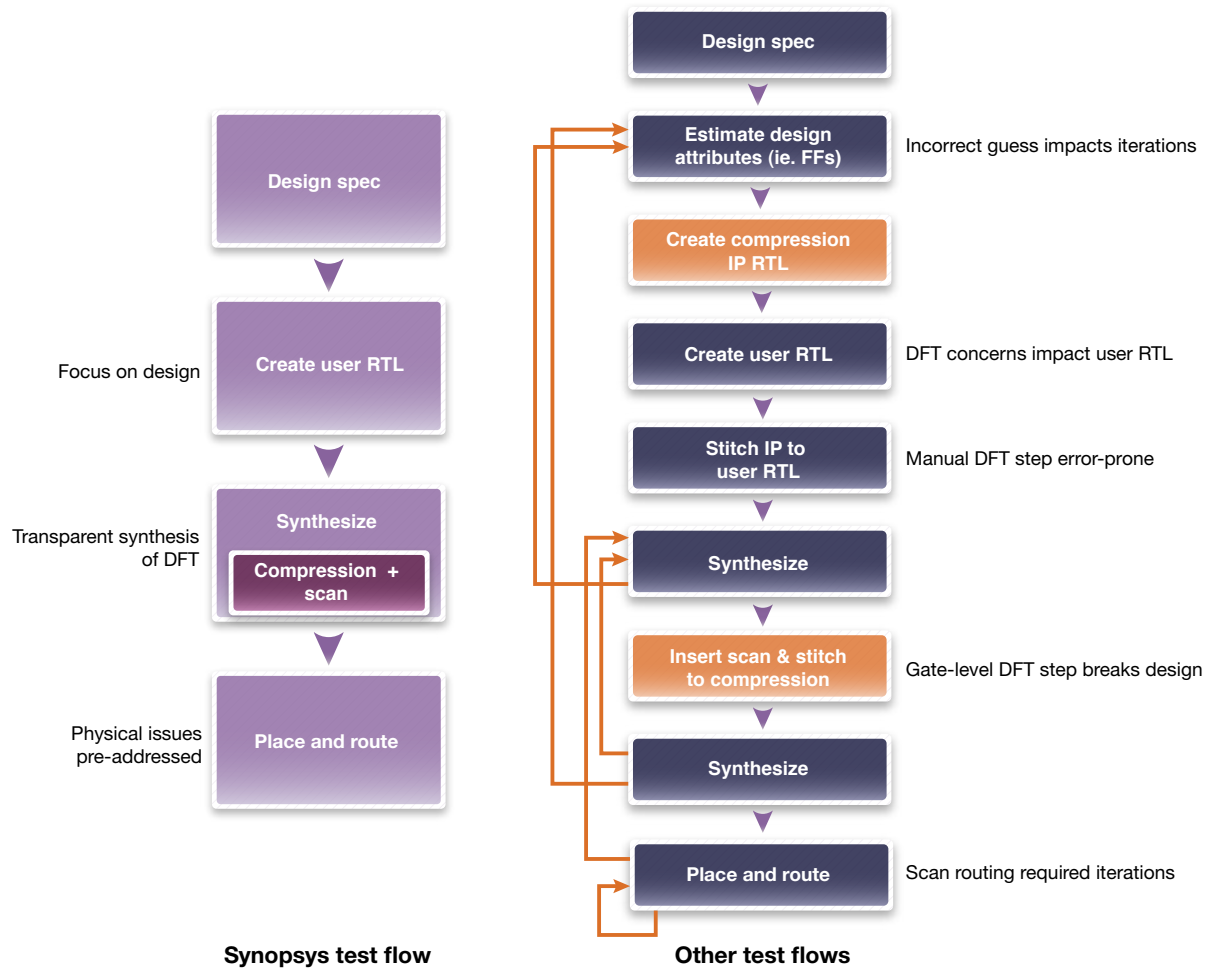


Figure 1: Comparison of Synopsys and other “bolt-on” test flows

DFTMAX compression makes DFT implementation transparent in the implementation flow, without interfering with the designers’ need to meet functional, timing and power requirements. DFTMAX compression works with DC Graphical to proactively minimize congestion associated with high scan compression. For maximum productivity, DFTMAX compression provides design rule checking (DRC) and automatic test rule violation correction (known as AutoFix) capabilities that are tightly integrated within the synthesis environment. DFTMAX compression enables RTL designers to quickly and accurately account for testability and resolve any test issues early in the design cycle, thus rapidly achieving their DFT goals without costly design iterations.

DFT Issues Impacting Productivity

With bolt-on methodologies, test-related problems show up late in the design cycle. These flows (shown at the right in Figure 1) require the user to go through several steps before discovering test errors late in the flow. When test errors occur, the user must iterate the entire flow. These iterations are necessitated by discrete test tools operating prior to synthesis (and therefore lacking knowledge of the final design) and after synthesis on the gate-level netlist (requiring an additional synthesis processes to address the DFT logic impact on timing). In many cases, iterations requires many hours. Because of the length and number of iterations, the design schedule is usually significantly lengthened with bolt-on flows. There are multiple causes for it test errors that are described below.

Inaccurately estimated scan compression architecture creation

Most mid-to-large size designs utilize scan compression logic to minimize test costs. When the compression architecture must be determined before the design or block is completed, there is a chance the architecture will be incorrect. As seen in the top right of Figure 1, this approach requires an estimate of the total number of flip-flops early in the design flow in order to create compression logic that assumes a connection to a corresponding number of scan chains. If the estimate is incorrect, or more likely, if the designer changes a state machine at the RTL, the number of flip-flops will differ. Due to the change, all the steps must be re-executed. In some cases, the amount of time to re-execute these steps is significant enough that the RTL designer must consider avoiding doing so in order to meet project schedule milestones. The downside of forgoing re-execution is a less-than-optimal compression architecture and likely decrease in the quality of the test program generated by an automatic test pattern generation (ATPG) tool.

Error-prone manual stitching of compression logic IP

Compression logic created as a separate RTL block prior to synthesis suffers from the drawback of manual stitching to the final RTL created the designer or design team. As design methodologies become more complex and compression techniques improve, the stitching becomes more intricate and consequently more error prone. Most errors introduced at this step are not caught until validation after synthesis. There are some cases where unintended, unconnected test logic is removed during synthesis and not caught in the post-validation stage. The validation step ensures that the inserted scan chains operate correctly, but does not flag missing compression logic which does not disrupt scan chain shifting. In other words, scan chain validation does not predict the final ATPG result and therefore does not detect all compression architecture errors. In short, an error during compression IP stitching could cause iterations late in the design flow or decrease the test quality due to missing scan compression logic.

Scan chain insertion into the gate-level netlist without design constraint considerations

Replacing regular flip-flops with scan flip-flops and stitching these cells into a scan chain without regard to design constraint considerations is an archaic technique developed several decades ago. The process of the designer “throwing the design over the wall” to the test engineer was possible and necessary because 1) scan chains were only composed of scan flip-flops and the wire interconnects, 2) scan chains connected only to primary input and output pins, 3) the timing delays of the signal propagation across gates were relatively greater than across the interconnects, 4) area impact of the scan test adopters was not an constraint, and 5) the DFT tools required a great amount of dedicated engineering time. Unfortunately, scan chain insertion at the gate-level is a technique that is no longer applicable to today’s designs.

Gate-oriented DFT tools ‘break’ the design by not considering the connections across voltage/power domains, the timing impact from the dominate wire connection delays, the area constraint for mobile and high volume consumer designs. The most common problem is critical paths that no longer met timing because of additional delay from the connection to the scan-input pin of the scan flip-flops. However, an increasingly common problem results from stitching scan chains across voltage domains without the use of a voltage level-shifters. Rectifying this voltage domain issue by simply adding many level-shifters will not minimize domain crossings and can significantly increase the silicon area. In the end, using gate-level DFT tools requires manual effort to fix the design and often includes extra synthesis steps as part of the repair. As discussed later, this manual effort can be tremendous for blocks requiring fixes for hundreds to thousands of test rule violations.

Addressing compression and scan chain congestion only during physical design

The need to decrease test costs has been a consistent theme since before the development of scan techniques. This need continues today as demonstrated by user forecasted plans for higher test compression. While advanced compression algorithms are useful in this endeavor, a key consideration for design teams is whether or not such extremely high test compression can be successfully implemented in a design flow.

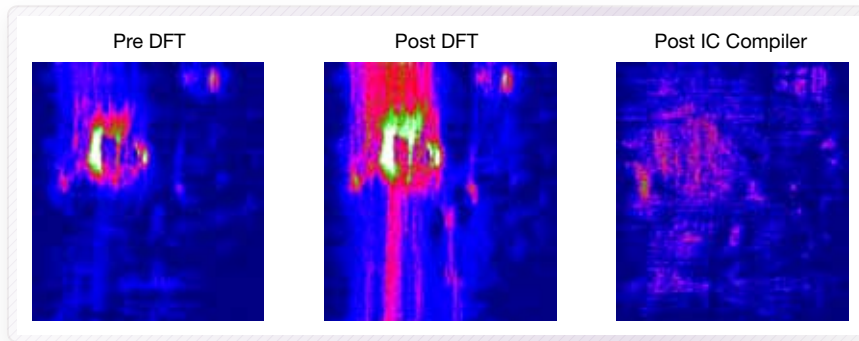


Figure 2: Congestion from very high scan compression

Among the challenges to implement high test compression is the congestion it creates. While physical design tools can deal with congestion, the goal during physical design is to complete the design as soon as possible to reach tape-out. Inserting high compression impacts the physical design (see Figure 2) and the schedule as congestion is introduced from the many connections between the compression logic and the internal scan chains. As seen in the middle congestion map of the figure, high compression (i.e. over 50x for many designs) introduces congestion. Synopsys IC Compiler attempts to minimize the congestion, but some congestion remains and requires extra effort and area to alleviate.

Productivity with DFTMAX Compression—Synthesis-Based Test

To ensure maximum designer productivity, all aspects of DFT implementation and validation are tightly integrated into the Synopsys synthesis flow. A typical design flow (see the left flow in Figure 1) shows that the DFT logic is created as transparently as possible in the design cycle. This is possible since the compression architecture and scan chains are specified within the DC Ultra synthesis environment (often referred to as “DC shell”) and verified for testability within the same environment. Unlike bolt-on flows that use an estimated number of flip-flops, DFTMAX compression utilizes accurate design data since such data is always available in the synthesis environment. The Synopsys test flow also eliminates errors due to manual connections since the connectivity is automated and verified before final synthesis.

The ability to verify testability of the design before targeting to a specific technology enables the designer to fix problems at the RTL or using AutoFix capabilities before committing to gates. The use of the synthesis environment to architect and check DFT, combined with the full use of synthesis technology to apply design constraints to the design logic and DFT concurrently, reduces or eliminates the chance of back-end iterations. Several capabilities and technologies that increase RTL designers’ productivity when implementing a design with DFTMAX compression are described below.

Comprehensive, unified test DRC

In a typical high-level design flow, multiple designers focus on module designs and then the system architects assemble the blocks to create the final chip. By using Synopsys DFTMAX compression (or DFT Compiler), the block designer can invoke the test DRC feature on the module prior to synthesis to verify DFT rules. The primary function of test DRC is to provide feedback on the testability of the design during the pre-synthesis stage. The designer has the option to fix the violations in the RTL source code based on the feedback. This enables the designer to account for the RTL testability early in the design process and satisfy a key DFT closure requirement.

A test verification tool for use by designers not deeply experienced in DFT must have three important attributes:

1. Be able to identify, verify and cover a wide set of testability rules
2. Ensure consistency with downstream gate-level ATPG DRC to avoid false violations that cause iterations late in the design cycle
3. Provide feedback within the designer's environment

The test DRC capability of DFTMAX compression was designed around these key attributes to enable the designer to create “test-friendly” design that can then be easily synthesized in the synthesis environment. This DRC is the same as the DRC capability provided in TetraMAX ATPG (see Figure 3) to ensure consistency on test rule violations.

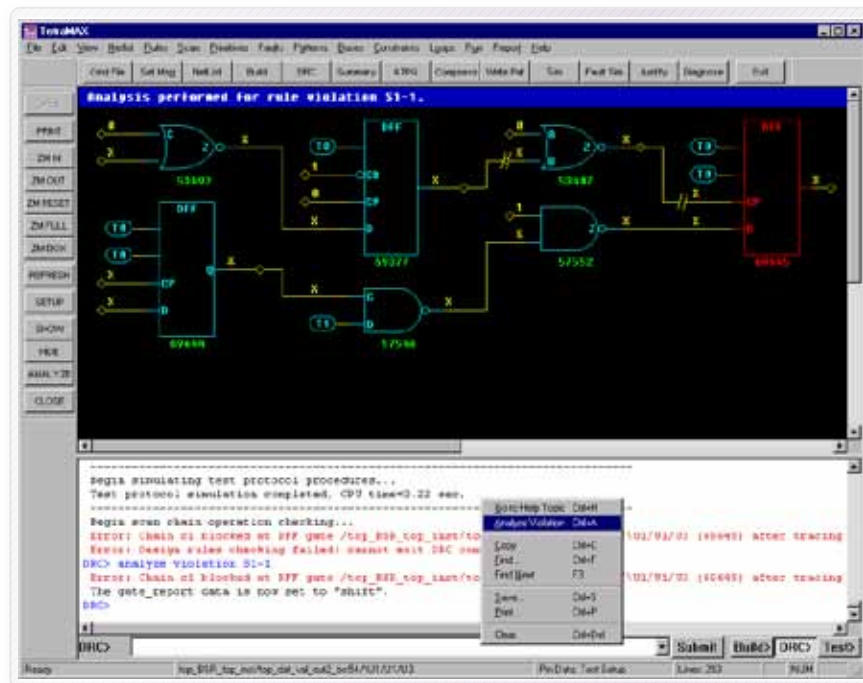


Figure 3: Unified test DRC graphical interface for DFT and ATPG

Using simulation techniques, test DRC ensures that the testability analysis is completely accurate, and will not cause false DRC violations in the downstream process. A comprehensive check of all pre-scan rules by test DRC ensures a majority of rules are checked, including the following set of violations:

- ▶ Violations that prevent scan insertion (e.g., uncontrollable clock or asynchronous set/reset to a flip-flop)
- ▶ Violations that prevent data capture (e.g., clock signal drives data pin of flip-flop)
- ▶ Violations that reduce fault coverage (e.g., combinational feedback loops)

An important requirement for any test DRC capability is to take into account test initialization constraints. The initial test constraints are set up using existing test synthesis commands. DFTMAX compression's test DRC simulates the initialization so that the test rules can be checked in their presence. Otherwise, the results of the DRC verification would be erroneous.

Automatically fixing test DRC during synthesis

The feedback from test DRC enables designer to identify the nature of the testability violation and then choose to fix any of these violations at the RTL. The designer also has the option to let the AutoFix capability fix these violations at the gate level during the synthesis stage. AutoFix is a capability developed to automatically fix the testability violations within the synthesis environment while meeting timing constraints. Manual implementation of testability fixes at the gate level can break design constraints and invariably leads to additional synthesis iterations to preserve timing constraints. AutoFix focuses primarily on the controllability of clocks and asynchronous set/reset signals, since these are some of the most common testability problems. After DRC violations, such as lack of controllability of clocks and asynchronous set and reset signals are detected, the designer uses the AutoFix capability to automatically insert test logic at the gate-level to fix these violations. It ensures the netlist is testable and ready for ATPG. Since AutoFix is integrated within test synthesis, the testability fixes have minimal or no impact on the overall timing and area constraints of the design.

The RTL designers' ability to deal with test design rule violations at the gate level rapidly diminishes as design size and complexity grow into the multimillion gate range. For example, a half-million gate block can have anywhere from 5,000-10,000 test design rule violations and would need 2,000-3,000 test points to fix the violation. To manually fix this many violations while preserving timing constraints, the designer could easily spend a month or longer. On the other hand, when using the AutoFix feature, the designer is able to rapidly fix the violations within a day while, at the same time, preserve the timing constraints. The design flow for fixing testability violations using AutoFix is a seamless part of the Synopsys test synthesis flow. The flexibility of AutoFix lets designers use the capability either in a top-down or bottom-up high-level design flow.

Minimizing area and congestion due to DFT

In addition to automatically achieving timing closure with DFT and providing automated DFT-related logical fixes that adhere to design constraints, DFTMAX Compression provides several other advantages stemming from its incorporation into synthesis technologies. Foremost is its ability to automatically minimize area for DFT. As an example consider the case of a shift register. Such structures are innately scanned and only require the first flip-flop be replaced with a scan flop (see Figure 4) and the output of the structure fan-out to the next scan flip-flop (seen as the wire labeled "SO"). Without leveraging synthesis, the alternative at the gate level is to replace each flip-flop with a scanned flip-flop. This method significantly increases the area compared to the area of the shift register with the first element replaced. In the case of large graphic processor design, the difference in area was up to 5% due to the heavy use of shift registers.

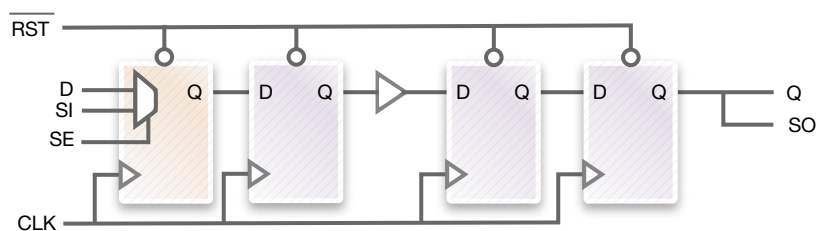


Figure 4: Example of minimal area impact on shift register using DFT MAX compression

Another advantage provided by DFTMAX Compression is the ability to operate in conjunction with DC Graphical to proactively reduce congestion from the test compression logic and scan chain connections, reducing the amount of effort and time for physical design systems such as IC Compiler. DC Graphical has specific knowledge of the compression architecture generated by DFTMAX Compression and can apply congestion optimization techniques unique to the architecture. Using the same design show in Figure 2, the reduction of congestion by DC Graphical is noticeable for all three stages – pre-DFT, post-DFT and after physical design with IC Compiler (see Figure 5). The result is faster execution with of physical design with IC Compiler.

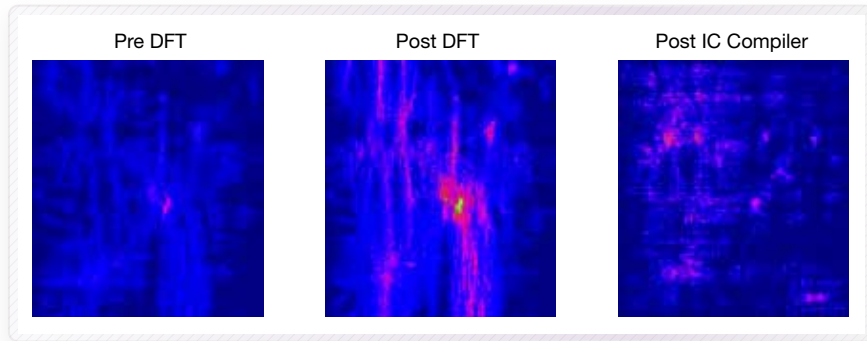


Figure 5: DC Graphical reduces congestion caused by high scan compression

Automatic handoff to ATPG

The final step in the test process for the design team is the generation of test patterns by ATPG. DFTMAX compression minimizes the RTL designers' effort for this step by automatically creating the description of the scan chain operation or protocol file. The protocol file drives ATPG and must be in a valid, readable format. DFTMAX Compression creates the file using the IEEE 1450 standard (commonly known as STIL), which TetraMAX ATPG reads. The designer is only required to supply the netlist and library into TetraMAX ATPG and then execute a few steps to generate patterns for the tester. TetraMAX ATPG has an easy-to-use graphical and command interface to guide the user.

Synthesis-Based Test Maximizes RTL Designer Productivity

With the incredible growth rate of chip complexity and increased demands for quality, the adoption of test compression, can, high-speed SERDES IP with built-in test, and embedded memory test and repair is mandatory. Scan compression is the dominant manufacturing test methodology because it provides high fault coverage while lowering test costs. However, there are still testability issues that impact the overall cost of the design and impact the productivity of the many RTL designers who are focused on high value-add engineering rather than DFT implementation. Many of these testability issues are due to bolt-on test tools that do not address the impact of DFT during the high-level design process or leverage the synthesis environment. Further problems of these flows arise due to lack of consistency between scan compression architecture and the design attributes, such as the estimated and actual number of flip-flops in the design, leading to unanticipated design iterations. DFTMAX compression, with its synthesis-based test technology, addresses these issues and is uniquely positioned to offer the most transparent design with test synthesis flow. Achieving testable designs require that designers consider testability at during the synthesis stage to minimize the costly back-end iterations that wreak havoc with design schedules. DFTMAX compression — with a unified test DRC, AutoFix, timing closure, area optimization, and congestion optimization capabilities — enables the designers to easily and predictably meet test requirements without impacting design schedules. Forward thinking companies that embrace these DFT tools and methodologies will ensure that quality products reach the market on schedule, and ahead of the competition.