DFTMAX™, DFTMAX™ Ultra, and DFTMAX™ LogicBIST Functional Safety Manual

N-2017.09

SYNOPSYS®

Copyright Notice and Proprietary Information Notice

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Trademarks

Synopsys, Inc.
690 E. Middlefield Road
Mountain View, CA 94043
www.synopsys.com

Contents

Customer Support

Customer support is available through SolvNet online customer support and through contacting the Synopsys Technical Support Center.

Accessing SolvNet

SolvNet includes an electronic knowledge base of technical articles and answers to frequently asked questions about Synopsys tools. SolvNet also gives you access to a wide range of Synopsys online services, which include downloading software, viewing Documentation on the Web, and entering a call to the Support Center.

To access SolvNet:

Go to the SolvNet Web page at http://solvnet.synopsys.com/.

If prompted, enter your user name and password. (If you do not have a Synopsys user name and password, follow the instructions to register with SolvNet.)

If you need help using SolvNet, click SolvNet Help in the Support Resources section.

Contacting the Synopsys Technical Support Center

If you have problems, questions, or suggestions, you can contact the Synopsys Technical Support Center in the following ways:

- Open a call to your local support center from the Web by going to http://solvnet.synopsys.com/ (Synopsys user name and password required), then clicking “Enter a Call to the Support Center.”
- Send an e-mail message to your local support center.
  - E-mail support_center@synopsys.com from within North America.
  - Find other local support center e-mail addresses at https://www.synopsys.com/support/global-support-centers.html.
- Telephone your local support center.
  - Call (800) 245-8005 from within the continental United States.
  - Call (650) 584-4200 from Canada.
  - Find other local support center telephone numbers at https://www.synopsys.com/support/global-support-centers.html.

1
Scope of This Document

This document assists the end user with running the DFT Compiler tool. This is a feature of Design Compiler tool to fulfill the requirements of the ISO 26262 standard if the tool is used in the development of a safety-critical component. This assistance implies achieving confidence in the use of the compilation tool in the DFT Compiler tool, as defined in ISO 26262 - Part 8, Clause 11.

You should have experience in the following areas of Digital IC design: RTL, Verilog and UNIX.

This document assists with creating confidence in the use of the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST generated by the DFT Compiler tool. Design Compiler is used to generate test insertion logic instrumental in the identification of a properly-functioning (safety critical) component. The fault coverage report for these test vector data establishes a metric for the thoroughness of these tests. Thus, the netlist requires qualification according to the ISO 26262 standard. The DFT Compiler tool contains other features that are not part of the compilation tool. The IDE is an example of this. These other features are not safety critical and are not qualified under ISO 26262.

The main goals of the present guidelines are to ensure that the following are satisfied:

- The requirements of the ISO 26262 safety standard for ASIL D as defined by the standard
- The restrictions posed on the use of the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST resulting from the qualification of the Tool.

Confidence in the use (as defined by the ISO 26262 standard) of the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST is achieved by:

- Verifying the pre-defined Tool Confidence Level (TCL) of the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST netlist compliance with the restrictions and recommendations in this Tool Qualification Report and the DFT Compiler User Guides for the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST

The process to achieve the required confidence in the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST are described in this Tool Qualification Report. The results of this process must be documented in the Tool Criteria Evaluation Report for the fault coverage report.

The completeness of the process is achieved by tracing the requirements of the ISO 26262 standard to the Tool Qualification Report (this document), and the SGS-TÜV certification of Software Tool Qualification Report for DFTMAX, DFTMAX Ultra, and LogicBIST.

The design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST are qualified for the highest safety level, ASIL D. This implies that it can also be used for lower safety levels, if the processes described in this Tool Qualification Report are implemented.

This document is structured as follows:

- Chapter 2 contains a glossary of terms used.
- Chapter 3 describes the three products and provides a general description of safety-related requirements and definitions concerning the scope and environment of design-for-test insertion flows.
- Chapter 4 describes the Software Tool Classification Process. The goal of this Tool Qualification Report Package is to assist the application developer with fulfilling the safety requirements for the use of the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST. These requirements are defined by the ISO 26262 standard (ISO 26262), referred to as "The Standard" in the following pages. This chapter describes the general method used to fulfill the requirements.
- Chapter 5 has the description of the tool workflow with design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST
- Chapter 6 describes the Tool Use Case Details for the three Design-for-Test Insertion Flows.
- Chapter 7 describes the Tool Requirements of the Design-for-Test Insertion Flows
- Chapter 8 describes the Tool Restrictions of the Design-for-Test Insertion Flows

User Competence

To properly use with the DFT Compiler for design-for-test insertions flows, you must have a good understanding and working knowledge of the following:

You should have experience in the following areas:

- Digital IC design
- Verilog or VHDL
- UNIX
- A Unix based text editor
- Design-For-Test
- The flow to create the DFT Compiler basic scan insertion described in DFT Compiler Design-for-Test User Guide Part II Chp6-Chp16
- Access to the on-line Synopsys Technical Support Center

2
Glossary

This section defines technical terms used within this document.

Terms and Definitions

Term	Definition
Component	(Part of) an electronic system that implements a function in the vehicle. See also Part 1 of the ISO 26262 standard for the definition. The Standard also refers to elements and items, but for the purpose of this *Tool Qualification Report* this makes no difference.
DFT flows	Design-for-test flows include DFTMAX, DFTMAX Ultra, and DFTMAX Logic BIST
DFT Compiler	DFT Compiler is the Synopsys test synthesis solution
DFTMAX Compression	If you are currently implementing standard scan logic, you can insert DFTMAX compression into your design by using a single additional command. Typically, no other changes to your design are required.
DFTMAX Ultra Compression Architecture	DFTMAX Ultra compression uses a shift-register scan-data architecture and a single scan clock to deliver very high compression without restriction on the number of I/O pins. The input shift register feeds the decompression logic that provides data to many internal scan chains. The output shift register compresses the scan-out data using XOR logic. This architecture delivers high scan compression levels while providing a scan-compatible interface that retains the simplicity of a basic scan design.
LogicBIST	The LogicBIST tool is a synthesis-based solution for in-system self-test of digital integrated circuits used in automotive, medical, and aerospace applications. LogicBIST addresses functional safety requirements set forth by standards such as ISO 26262 for the automotive semiconductor industry.

LogicBIST Architecture	LogicBIST self-test enables a design to test itself using the same scan chains already implemented for manufacturing test. It uses a pseudo-random pattern generator (PRPG) to create scan data, and a multiple-input signature register (MISR) to capture the design response. At the end of the test, if the actual signature matches the expected signature, the self-test asserts a PASS status.
TetraMAX	The TetraMAX ATPG program that runs on a soft component in a safety-critical process for fault coverage report.
Error	In this document, refers to a *potential error*
(Potential) error	An error that might occur as a result of using the tool
RMgen	Reference methodologies script generator
Software Tool	Synopsys DFT Compiler
STAR	Synopsys Technical Action Request. Reports, documents, and tracks a product Bug or Enhancement request (called a "B" or "E"). Stored in a CRM database. Well maintained, documented, tracked and monitored comprehensive graphical interface. Web-based access and searches Business Warehouse (BW) allows complex reporting. CRM stores all relevant information and the testcase data is stored in UNIX. CRM is accessible by Synopsys employees, but *not* customers. However, customer contacts are automatically notified when STARs are filed or when status changes. Limited STAR info is displayed on SolvNet for the customer who are associated with the STAR's User Site.
Tool	The Design Compiler tool according to ISO 26262
The Standard	The ISO 26262 *Standard for Road Vehicles – Functional Safety* (ISO26262)
Tool classification	Determination of the required tool confidence level (TCL)
Tool evaluation	See tool classification
TCL	Tool Confidence Level (ISO 26262): required confidence in the tool when used in the analyzed tool chain. • TCL1=low confidence required • TCL2=medium confidence required • TCL3=high confidence required
Use case	An application scenario of the tool

3
Product Description

This section provides a general description of safety-related requirements and definitions concerning the scope and environment of Design-for-test insertion flows*.*

Version

The *DFT Compiler* report produced by DC Compiler version N-2017.09 (see section 4 for a more detailed description of the individual tools and qualification status).

Tool	**Version**
DFT Compiler	N-2017.09
Design Vision	N-2017.09
HDL-Compiler	N-2017.09

You must have the Design Compiler, DFTMAX, and DFTMAX Ultra tools installed and licensed at your site.

You must have an HDL-Compiler license for compressed scan insertion.

Supported Platform and Installation Guide

The supported platforms for the safe use of the DFT Compiler installation are:

- x86_64 Red Hat Enterprise Linux v6.6 and v6.7
- x86_64 SUSE Enterprise v11

- Binary-compatible hardware platform or operating system. See http://www.synopsys.com/qsc for the latest on supported platforms, including required OS patches.
- The following link provides users with guidance compute platform roadmaps http://www.synopsys.com/Support/LI/SupportPlatform/Pages/PlatformsRoadmap.aspx
- The following link provides platform notices: http://www.synopsys.com/Support/Li/SupportPlatform/PlatformNotices/Pages/default.aspx
- The following link provides Synopsys Installation Guide https://www.synopsys.com/support/licensing-installation-computeplatforms/installation.html

Product Documentation and Support

The Synopsys reference methodologies provide a set of product- and release-specific reference scripts that are intended to show you the latest recommended methodology for a specific product and release. These scripts are used as a starting point that you can adapt and build upon for your design requirements. You should use these scripts as a template for developing your flows. The scripts are documented with embedded comments that are designed to be clear and self-explanatory.

Standard versions of the reference methodology scripts are available for the Design Compiler Tool. The scripts are available for download from SolvNet:
https://solvnet.synopsys.com/rmgen

RMgen and Product Reference Methodology Training Link via SolvNet

https://solvnet.synopsys.com/retrieve/025090.html

DFT flows are part of the Design Compiler scripts. The DFT flow options include:

DFTMAX (Scan Compression)
DFTMAX Ultra (Scan Compression)
DFTMAX Logic BIST

Using the RMgen Web-Based User Interface

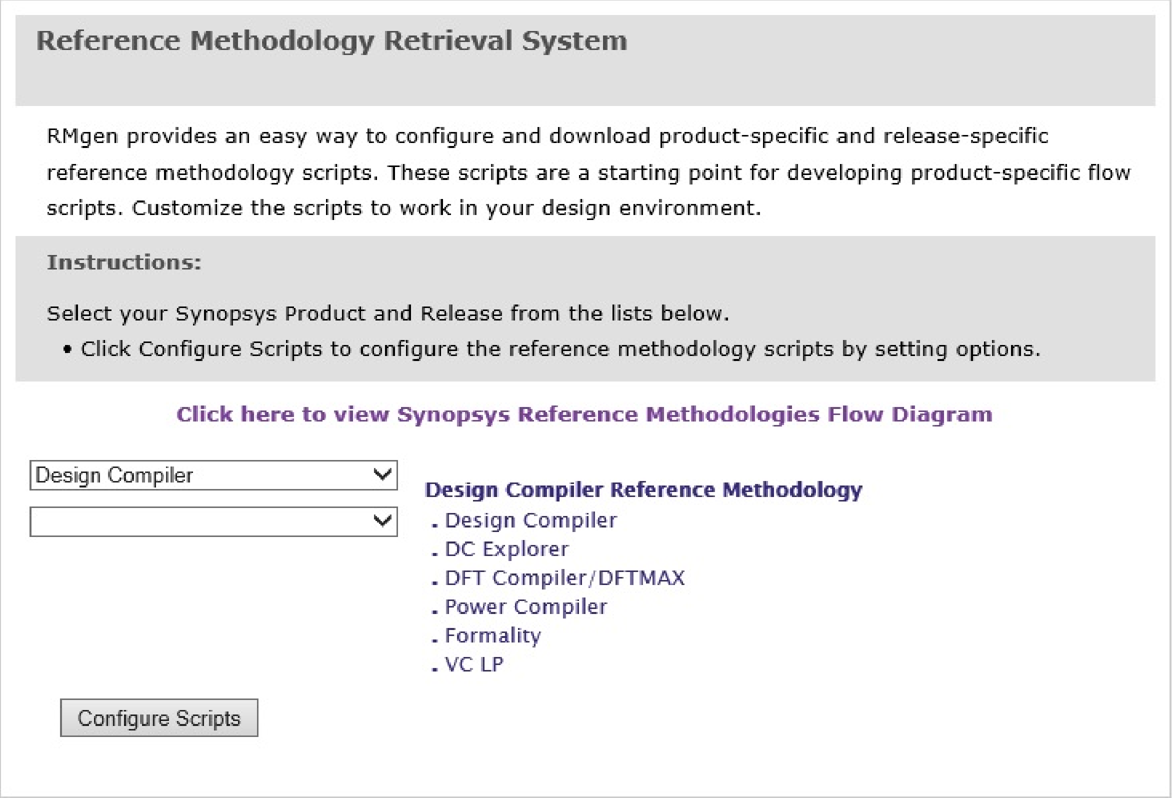
RMgen provides an easy-to-use Web-based environment for you to obtain default or customized reference methodology scripts for your design environment. The user interface has Web pages for selecting, configuring, and downloading the reference methodology scripts.

Select Product and Release

The Reference Methodology Retrieval System page shown in **Figure** 1 on the following page allows you to select the product and release for the intended script.

Select Product and Release -> Configure Scripts -> Download Scripts

Figure 1: RMgen Reference Methodology Retrieval System Page: Select Product and Release



Reference Methodology Retrieval System

RMgen provides an easy way to configure and download product-specific and release-specific reference methodology scripts. These scripts are a starting point for developing product-specific flow scripts. Customize the scripts to work in your design environment.

Instructions:

Select your Synopsys Product and Release from the lists below.

- Click Configure Scripts to configure the reference methodology scripts by setting options.

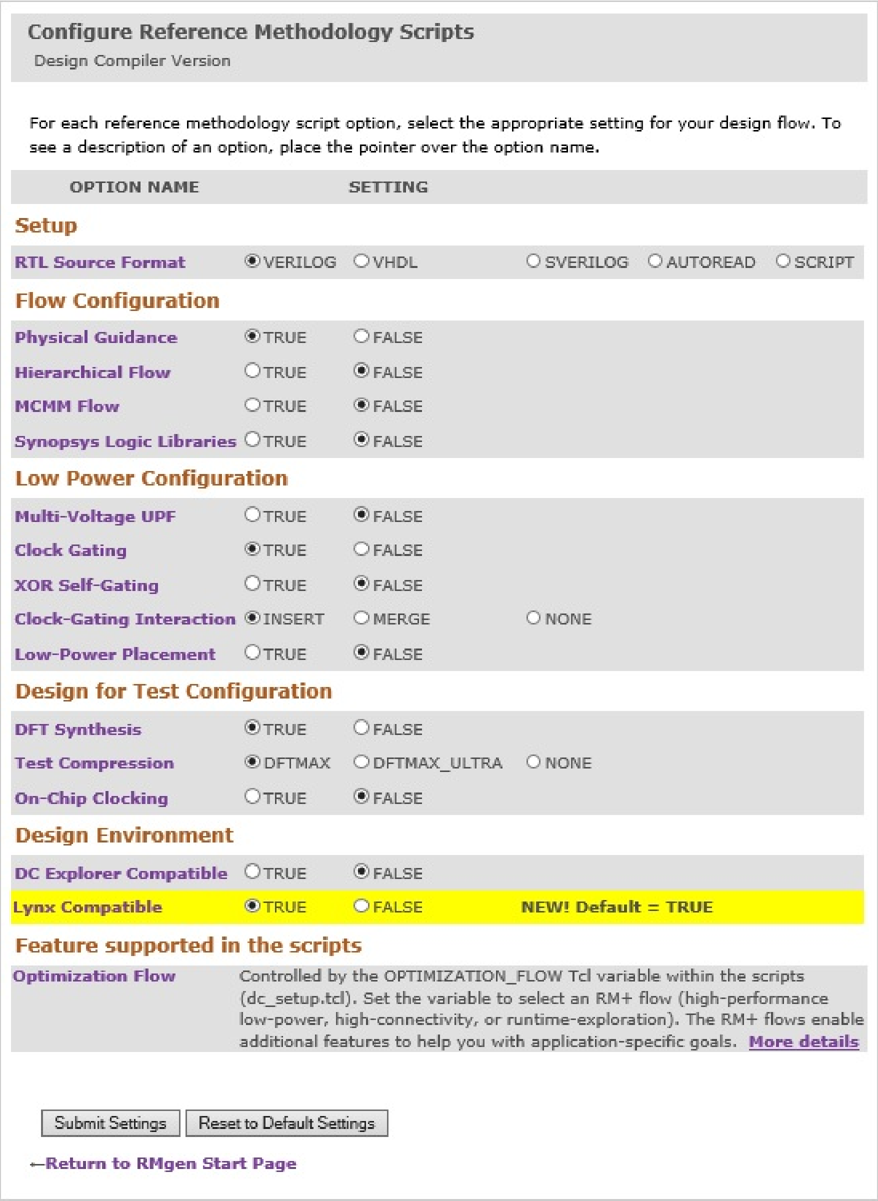
Click here to view Synopsys Reference Methodologies Flow Diagram

Design Compiler Reference Methodology

- Design Compiler
- DC Explorer
- DFT Compiler/DFTMAX
- Power Compiler
- Formality
- VC LP

Select the product and release using the two selection boxes. Click the “Configure Scripts” button to select the product’s option settings in the Configure Scripts page.

The Design Compiler Reference Methodology scripts include options for running the Synopsys DFT Compiler and Synopsys DFTMAX™. Note that additional licenses are required when you run the DFT Compiler and DFTMAX tools inside the Design Compiler or DC Explorer tool. The sections of the reference scripts that involve using these tools are clearly marked, so you can easily include or exclude these sections when preparing your scripts.



Configure Reference Methodology Scripts

Design Compiler Version

For each reference methodology script option, select the appropriate setting for your design flow. To see a description of an option, place the pointer over the option name.

OPTION NAME	SETTING				
Setup					
RTL Source Format	● VERILOG	○ VHDL	○ SVERILOG	○ AUTOREAD	○ SCRIPT
Flow Configuration					
Physical Guidance	● TRUE	○ FALSE			
Hierarchical Flow	○ TRUE	● FALSE			
MCMM Flow	○ TRUE	● FALSE			
Synopsys Logic Libraries	○ TRUE	● FALSE			
Low Power Configuration					
Multi-Voltage UPF	○ TRUE	● FALSE			
Clock Gating	● TRUE	○ FALSE			
XOR Self-Gating	○ TRUE	● FALSE			
Clock-Gating Interaction	● INSERT	○ MERGE	○ NONE		
Low-Power Placement	○ TRUE	● FALSE			
Design for Test Configuration					
DFT Synthesis	● TRUE	○ FALSE			
Test Compression	● DFTMAX	○ DFTMAX_ULTRA	○ NONE		
On-Chip Clocking	○ TRUE	● FALSE			
Design Environment					
DC Explorer Compatible	○ TRUE	● FALSE			
Lynx Compatible	● TRUE	○ FALSE	NEW! Default = TRUE		
Feature supported in the scripts					
Optimization Flow	Controlled by the OPTIMIZATION_FLOW Tcl variable within the scripts (dc_setup.tcl). Set the variable to select an RM+ flow (high-performance low-power, high-connectivity, or runtime-exploration). The RM+ flows enable additional features to help you with application-specific goals. More details				

Using RMgen and Reference Methodology Scripts Application Note, Version 4.2

https://solvnet.synopsys.com/retrieve/025091.html

https://solvnet.synopsys.com/retrieve/customer/application_notes/attached_files/025091/RMgen_and_Reference_Methodology_Application_Note_4.2.pdf

Using RMgen for Design Compiler Reference Methodology

https://solvnet.synopsys.com/retrieve/021023.html

Use SolvNet to Download User Guides

After choosing the "DFT Compiler / DFTMAX™" from the SolvNet Documentation Retrieval, you will have access to all the PDF documentation files that are see at this link:

https://solvnet.synopsys.com/dow_retrieve/latest/dftolh/Default.htm

The documentation of the DFT Compiler Design-for-Test Product flows is included as part of Design Compiler.

If design-for-test product flows are used for developing safety-critical components for road vehicles (as defined by ISO 26262), the development process must comply with the guidelines and constraints defined in the DFT Compiler design-for-test product flows.

Guidelines to assist with achieving the goals of the ISO 26262 standard for the safe use of DFT Compiler can be found in this Tool Qualification Report.

The Synopsys Technical Support Center has on-line support and information for DFT Compiler DFT flows. Among other items, it has the latest information on known defects and workarounds related to the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST report. The Synopsys Technical Support Center is the channel by which DFT Compiler users can get support and report defects. See the section on "Customer Support" at the start of this document for details of the Synopsys Technical Support Center.

Use SolvNet For Training Education and Labs

SolvNet users have access to hands-on training and education for Synopsys tools and methodologies. Flexible options for learning online or in the classroom.

Browse our training and education curriculum by product or service:

https://www.synopsys.com/support/training/rtl-synthesis/dft-compiler.html

The lab files, for recent courses, download page can be accessed directly via SolvNet (article # 002471):

https://solvnet.synopsys.com/retrieve/002471.html

DFT Compiler Update Management

Synopsys may release new versions of DFT Compiler at any time to extend functionality or fix defects in the tool. When a new version is available, a notification of a new version is posted in the Synopsys Technical Support Center.

If a new version of the DFT Compiler Tool is installed, users must check the following:

The *Release Notes* of the software for any changes that are related to safety-critical aspects of using DFT Compiler.

4 Software Tool Classification Process

The goal of this Tool Qualification Report Package *is to assist the application developer with fulfilling the safety requirements for the use of the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST. These requirements are defined by the ISO 26262 standard (ISO 26262) – referred to as "The Standard" throughout this document. This chapter describes the general method used to fulfill the requirements.*

The process described in this Tool Qualification Report Package is based in Part 8 Clause 11 of The Standard. This clause defines two "Work Products" as its result (in Part 8 Clause 11).

The first work product, the *Software Tool Qualification Report*, describes the qualification process and results for the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST. Synopsys has already qualified the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST internally

The Software Tool Qualification Report requires no additional work needs to be done because the qualification of the tool is independent of its actual use, and has already been performed by Synopsys. The only case where the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST user must create the work product according to ISO26262-8 clause 11.5.2, "Software tool qualification report" is if the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST user cannot comply with the restrictions and recommendations described in this Tool Qualification Report and the DFT Compiler User Manuals.

The second work product, the *Software Tool Criteria Evaluation Report for the Design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST*, places the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST into the actual context of its use. Therefore this second work product can only be created when the following are documented:

- What the Component is for which the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST is used to generate a netlist
- How the tool is used

This has to be done by the developer or designer of the Component.

Process to Achieve Confidence in the Use of Software Tools

The general process to achieve safe use of a software tool is to determine if the tool can handle the specific use case of the tool for the specific Component. This general process is defined in Part 8 Clause 11 of the Standard. For the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST (the compilation tool in the Design Compiler tool, which is the software tool) this process is simplified because the qualification of the tool has already been performed. What remains is to confirm that the actual use case is within the constraints set by the qualification.

The Tool qualification package for DFTMAX, DFTMAX Ultra, and LogicBIST is qualified by Synopsys and a predetermined Tool Confidence Level TCL1. This qualification is confirmed by SGS-TÜV. Using a predetermined confidence level is described in Part 8 Clause 11.4.2.

The process to achieve confidence in the tool requires documentation and confirmation of the predetermined TCL level. Confirmation of TCL level 1 depends on the actual use case of the tool, which must match the requirements and constraints that are described in the DFT Compiler User Manual.

An important step in the process is to collect a set of detailed information, including a description of how use the tool is used.

Strict adherence to the DFT Compiler User Manuals is also required. This conforms to Part 8 Clause 11.4.3, which requires that the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST be used within the constraints that follow from the qualification of the compilation tool.

If the predetermined TCL level 1 is confirmed, no additional work is needed to qualify the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST, as defined in Part 8 Clause 11.4.6.1. If TCL level 1 cannot be met by the actual use because that use is outside the scope of the Tool Qualification Report, the user of the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST may need to do additional qualification of the Tool. This is not further elaborated in this guide, but it may require setting up a test environment for the tool by the user.

Section 4 provides a detailed guide of the steps needed to achieve confidence in the use of the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST.

Detailed Overview of Part 8 Clause 11

The following is an overview of the sub-clauses of Part 8 Clause 11 and how they fit into the process described in this Tool Qualification Report Package. This overview can be used as a reference.

- 11.1 Describes the objectives of Clause 11. The objectives are to determine the applicable TCL, and to create evidence for the suitability of the tool for the task.
- 11.2 Gives an overview and some background to Clause 11.
- 11.3 Describes which inputs are needed before commencing on achieving confidence in the use of the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and

LogicBIST. These inputs are expected to be available from external sources (the user manuals of the DFT Compiler Tool from Synopsys), or from previous steps in the safety process.

- 11.4 Establishes the general requirements on the safe use of the fault coverage report and confirms that these requirements are met.
- 11.4.1 States the general requirement for when the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST should comply with Part 8 Clause 11.
- 11.4.2 Requires checking of the validity of the predetermined TCL 1 if that was established independently of the context of use. For the Tool Qualification Package for DFTMAX, DFTMAX Ultra, and LogicBIST the TCL is confirmed by SGS-TÜV.
- 11.4.3 Requires that the use of the software tool must comply with the constraints of its qualification, which are defined in this Tool Qualification Report and the DFT Compiler User Manuals for the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST.
- 11.4.4.1 Requires a detailed description of the usage of the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST including identification of its version number and the environment in which it is used. It also requires a description of how the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST are qualified.
- 11.4.4.2 Describes the prerequisite information that is needed for proper use of the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST.
- 11.4.5 Determines the TCL of the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST in the context of its use. Because TCL1 is already predetermined, confirmation that usage of the tool complies with the requirements of the *Tool Qualification Report* is necessary.
- 11.4.6 Describes how to choose the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST qualification method based on the TCL. For the predetermined TCL1, no additional qualification is needed.
- 11.4.9 Qualification method based on validation of the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST. Validation must show that the tool complies with its specification; that malfunctions and their consequences are documented. This is one of the qualification methods used for the Design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST, which in turn results in the predetermined TCL1.
- 11.4.10 Requires a review of the predetermined TCL and the method for the qualification of the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST. It refers back to Part 2 Table 1 of The Standard.
- 11.5 Defines the two work products of Part 8 Clause 11: the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST criteria evaluation report, which can be created using the guidelines in this *Software Tool Qualification Report*. The user must meet the restrictions and recommendations described in this Tool Qualification Report.

5
Tool Workflow with Design-for-Test Insertion Flows for DFTMAX, DFTMAX Ultra, and LogicBIST

This section contains the guide to obtaining confidence in the safe use of the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST, as required by Part 8 Clause 11 of the Standard. Following the steps in the guide results in the Tool Criteria Evaluation Report for the Design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST*; this is the use-case-specific product of Part 8 Clause 11.*

This guide is broken into several steps. For each different use of the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST in the Component, and for each different Component, these steps must be repeated.

Step 1: Prerequisites

This step implements Part 8 Clause 11.3 of the Standard.

Obtain the following documents and information and add the relevant information such as title, origin, publication date, and version number from the DFT Compiler tool.

- The Safety Plan for the Component in accordance with Part 4 Clause 5.5.2 of the Standard. This document is specific to the development of the Component and not part of the DFT Compiler Tool.
- In the Tool Qualification Report, find the prerequisites for the use of the Design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST according to the safety lifecycle (See Part 3 Clause 6).
- Determine the maximum ASIL from the Safety plan.
- The user manuals for the DFT Compiler Tool provided by Synopsys for the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST provided by Synopsys.
- The SGS-TÜV Certificate of ISO 26262 Compliance for Tool Qualification Report for the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST provided by Synopsys (SGS-Cert).

- Determine, for the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST, the system requirements on the (host) computer that is used to run the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST.

Step 2: General Requirement

This step implements Part 8 Clause 11.4.1 of the Standard.

Part 8 Clause 11.4.1.1 states the condition under which the Design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST must comply to clause Part 8 Clause 11 "Confidence in the use of software tools". The Standard states the following:

> If the safety lifecycle incorporates the use of a software tool for the development of a system, or its hardware or software elements, such that activities or tasks required by ISO 26262 rely on the correct functioning of a software tool, and where the relevant outputs of that tool are not examined or verified for the applicable process step(s), such software tools shall comply with the requirements of this clause.

Thus, in the general case where the software issue that is created with the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST becomes part of the Component under development, Part 8 Clause 11 applies. The clause makes an exception if there is a procedure that ensures that no possible tool error remains undetected in the executable.

If Part 8 Clause 11 does not apply, one does not have to create the *Software Tool Criteria Evaluation Report for the Design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST* and one can stop here. Otherwise, continue with the next step.

Step 3: Validate the Predetermined Tool Confidence Level

This step implements Part 8 Clause 11.4.2 and Part 8 Clause 11.4.10 of the Standard, as well as Part 8 Clause 11.4.5-9.

The design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST comes with a predetermined Tool Confidence Level of 1 (TCL1) for applications with maximum ASIL D. This TCL1 level is documented in this Tool Qualification Report for design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST and is confirmed by SGS-TÜV. The tool can also be used for a lower maximum ASIL.

For this step, consult Table 1 in Part 2 of the Standard. From the table it follows that, at the highest ASIL D, the required independence of the authority to perform the confirmation review is I1. Independence level I1 is defined as "a different person." Thus, SGS-TÜV, being independent from Synopsys, has the required independence.

Implications of and Rationale for TCL1

The predetermined TCL1 is a low TCL, which implies a high degree of confidence that malfunction of the Design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST is prevented. As a result, no additional qualification of the Design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST is needed. However, TCL1 does require that the safety guidelines that follow from the qualification of the tool be followed (see Step 5).

Part 8 Clause 11.4.5 describes how the TCL of a software tool is determined by analysis. In Part 8 Clause 11.4.5.2b, the Tool Error Detection (TD) class is determined. TD1 is used here because the tool is based on the analysis performed by Synopsys, and this qualification has documented in the excel sheet which has been reviewed by SGS-TÜV. Therefore, following the *tool qualification report*, results in a high degree of confidence that malfunction of the tool is prevented or detected.

Given TD1, TCL1 follows from Table 3 (at Part 8 Clause 11.4.5.5). Consequently, Part 8 Clause 11.4.6.1 states that no further qualification of the Design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST is needed.

Step 4: Plan and Verify Usage of Design-for-Test Insertion Flows for DFTMAX, DFTMAX Ultra, and LogicBIST

This step implements Part 8 Clause 11.4.4.1 of the Standard.

11.4.4.1a

Print the identification and version number of the DFT Compiler tool with the following command:

```
dc_shell -version
```

Verify that the *DFT Compiler Manual* for the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST is valid for this installed version of the tool.

11.4.4.1b and c

These steps, in Section 6, document use cases of the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST. The goal of these steps is to verify that the actual use of the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST falls within the guidelines and constraints that are defined by this Tool Qualification Report.

This Tool Qualification Report contains a list of tool commands and a list of options for each command that can be used safely.

11.4.4.1d

Add the version identification string of the system on which the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST is installed. The version string can be retrieved on a Linux system with:

```
   uname -osr
```

Verify that this version of the system matches with the system requirements that are stated in the *Tool Qualification Report Package.*

11.4.4.1e

The maximum ASIL of the Component in which the software generated by the Design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST is used is already determined in Step 1, above. The user should verify that the maximum ASIL does not exceed the ASIL for which the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST is qualified. Since the Design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST has ASIL D, which is the highest ASIL, this verification is always true.

11.4.4.1f

The methods used to qualify the Design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST are in accordance with Part 8 Clause 11.4.9 (“Validation of the software tool”).

Step 5: Ensure that the DFT Compiler is Used in Compliance with the Tool Qualification Report

This step implements Part 8 Clauses 11.4.3 and 11.4.4.2 of The Standard.

Verify that the use of the Design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST complies to the guidelines and constraints of the “Design-for-test insertion flows” Tool Qualification Report.

Step 6: Completing the Work Products

This step implements Part 8 Clause 11.5 of the Standard.

Following and documenting Steps 1 to 5 in this Tool Qualification Report results in the completion of the Software Tool Criteria Evaluation Report for design-for-test insertion flows. Together with the confirmation, by SGS-TÜV, of the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST classification of TCL1, this fulfills the requirements of Part 8 Clause 11 to obtain confidence in the use of Design-for-test insertion flows.

6 Description of the Tool Use Case Details for the Design-for-Test Insertion Flows

This section describes the various use case details that apply to the design-for-test insertion flows.

DFT Compiler Flows Supported

There are three DFT Compiler insertion flow methodologies supported when performing design-for-test insertion for DFTMAX, DFTMAX Ultra and DFTMAX LogicBIST:

1. Top-Down
   Scan insertion is performed at the top-level for the entire design in one step

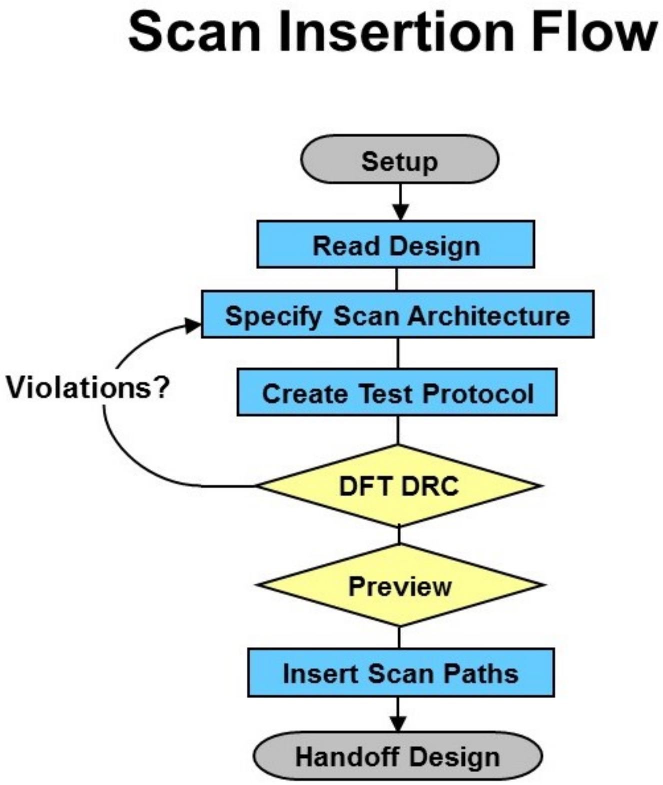
2. Bottom-Up
   Scan insertion is performed at the block-level and results are later combined at the top-level

3. Extraction (Inference)
   This design flow has existing scan chains already inserted that are extracted (or inferred) and added to the protocol. If the existing scan design was created by DFT Compiler, saved in .ddc format and no test attributes removed (three important conditions), DFT Compiler automatically recognizes the chains it inserted earlier.

If you were to bring in a Verilog or VHDL netlist with scan chains in it, however, there is nothing in the netlist to describe all the test attributes. Therefore, DFT Compiler has to use your test protocol and defined test attributes to "extract“ (or “infer”) the scan chain from the netlist.

Basic Steps in a Typical Scan Insertion Flow

This section discusses the basic steps of a scan insertion flow, as shown in the following figure.



There are seven basic steps in a typical scan insertion flow:

1. **Read the Design**
   Read a mapped design (read_ddc, read_verilog) and scan insertion is performed on mapped designs.

2. **Specify the Scan Architecture**
   Various commands control the scan architecture (number of scan chains, how clock domain are handled, etc.). The set_scan_configuration and set_scan_path commands are primarily used to control the scan architecture.

   Details for Scan Architectures DFTMAX, DFTMAX Ultra and DFTMAX LogicBIST are included in section 6 below.

3. **Create the Test Protocol**
   The test protocol describes how the design operates in scan mode. The signals involved in the protocol are declared with the set_dft_signal command. The protocol is created by the

create_test_protocol command.

4. **Run DFT DRC**
The dft_drc command performs DRC checks prior to scan insertion. The DRC violations can be debugged graphically with DesignVision or fixed by DFT Compiler with AutoFix.
Signals that need to be identified for dft_drc usage of “-view exist -type ( )” for the set_dft_signal command are: ScanClock, Reset, Constant, ScanEnable, ScanMasterClock, and ScanSlaveClock. This only covers signal types required for a typical scan chain insertion flow. There are additional signal types used for other DFT flows.

   The create_test_protocol is used to create the test protocol based on user-defined DFT signals and timing parameters. Once a test protocol has been created, it can be written out to disk with the write_test_protocol command. The syntax for this command is as follows:

```
write_test_protocol -output <filename>
```

   Refer to the DFT Compiler User Guide for more details.

   Note: The test_setup section must first be read with read_test_protocol –section test_setup command. The remaining portion of the protocol is created using the create_test_protocol command. If the commands are specified in reverse order, an error is reported.

   Note: Even if a protocol file already exists, it’s best to allow DFT Compiler to recreate the protocol each time. This ensures that the design and protocol are always in sync.

   There are three types of DRC runs that can be performed on a design with dft_drc:

   a) RTL DRC: Run during RTL development phase to identify DFT issues early in the flow.

   b) Pre-DFT DRC: Run prior to scan insertion to determine which scanable elements can be put on scan chains.

   c) Post-DFT DRC: Run after scan insertion to validate that the implemented scan chains can be traced. This is an optional post-insertion “sanity check” that can also be used to provide an ATPG coverage estimate. It can extract (infer) scan chain information from a design with existing scan chains.

5. **Preview**
The preview_dft command is used to get a preview of the scan architecture before it is actually implemented in the design. The preview step allows for quicker iteration cycles when changes need to be made to the scan architecture.

   Details on scan architectures, are included below in the descriptions of DFTMAX, DFTMAX Ultra and DFTMAX LogicBIST.

6. **Insert Scan Paths**
   The scan architecture is inserted into the design using the insert_dft command. Scan paths are inserted into the design, along with any other test logic specified (i.e. AutoFix, Test Points, OCC, DFTMAX, etc.)

DFTMAX

DFTMAX the Synopsys compression tool, uses the same flow as traditional scan. It gets the same coverage as scan and has similar ATPG runtime as scan.   DFTMAX compression provides synthesis-based scan data compression technology to lower the cost of testing complex designs, particularly when fabricated with advanced process technologies. DFTMAX compression reduces these costs by delivering a significant test data and test time reduction with very low silicon area overhead. The DFTMAX tool uniquely enables scan compression in Design Compiler synthesis and scan pattern generation in TetraMAX ATPG.

The command file below is an example of how to edit the basic scan flow to DFTMAX compression scan flow. The blue text represents the additional command required to support DFTMAX compression.

```
#############################################################################
# Modify Basic Scan to DFTMAX Compression Configuration
#############################################################################
set_dft_configuration -scan_compression enable
set_scan_compression_configuration -chain_count 5
set_dft_signal -type ScanClock -port clk
create_test_protocol
dft_drc
preview_dft
insert_dft
write -f Verilog -hier -o block1.v
write_test_protocol -out scan.spf -test_mode Inernal_Scan
write_test_protocol -out scancompress.spf -test_mode ScanCompression_mode
```

DFTMAX Ultra

DFTMAX Ultra compression is an advanced test compression technology that is designed for hierarchical flows to deliver high quality results as measured by test time, data volume, design area and congestion, and time to implementation. The technology delivers very high compression, even with few scan I/O pins. It uses the same signal interface as standard scan with minimal impact to the clock tree. The technology is designed to deliver good results with few internal chains to minimize any impact on layout.

The user interface for DFTMAX Ultra is very similar to DFTMAX. Compression insertion is enabled with the set_dft_configuration –streaming_compression enable command. The compressor can be configured with the set_streaming_compression_configuration command. You need to set at least the core chain count or max length, and the number of codec inputs and outputs.

```
#############################################################################
# DFTMAX Compression Configuration
#############################################################################
set_dft_configuration -scan_compression enable
set_scan_compression_configuration -xtolerance default -min_power true
# Define the test-mode to be used for adaptive scan compression
# set_dft_signal -view spec -type TestMode -port scan_compression_enable
write_test_protocol -out scancompression.spf -test_mode ScanCompression_mode
```

```
#############################################################################
# DFTMAX Ultra Compression Configuration
#############################################################################
set_dft_configuration -streaming_compression enable
# set_streaming_compression_configuration -inputs <> -outputs <> -chain_count <>
# Define the test-mode to be used for adaptive scan compression
# set_dft_signal -view spec -type TestMode -port scan_compression_enable
write_test_protocol -output STREAMING.spf -test_mode ScanCompression_mode
```

The protocol file and test model are written out as usual using the following command:

```
write_test_protocol -output STREAMING.spf \
   -test_mode ScanCompression_mode
```

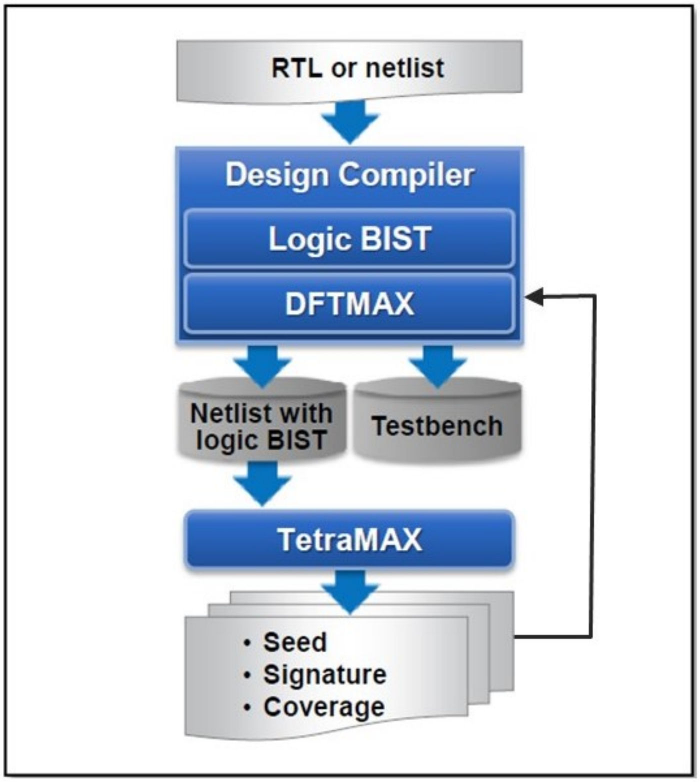
The Streaming DFT Planner provides better visualization of the DFTMAX Ultra architecture. This can help to ensure your specifications match what was intended. It can also help to see if chains and CODECs are balanced. You will also get information on the target compression and implementation details like OCC chains, pipeline depths, etc. To display the report, use the streaming_dft_planner command. Use this command after the create_test_protocol command and before the insert_dft command.

DFTMAX LogBIST

DFTMAX LogicBIST is a synthesis-based solution tool for in-system self-test of digital integrated circuits used in automotive, medical, and aerospace applications. LogicBIST addresses functional safety requirements set forth by standards such as ISO 26262 for the automotive semiconductor industry. The built-in self-test (BIST) capability enables a design to test itself autonomously without using external test data. The LogicBIST tool provides a low-overhead logic BIST (LBIST) solution for digital logic designs, such as automotive applications.

```
### DFTMAX LogicBIST Script
############################
set_dft_signal -type MasterClock -port clk_st -timing {45 55} -view
existing_dft
set_dft_configuration -logicbist enable -wrapper enable
set_dft_configuration -streaming_compression enable
set_wrapper_configuration -maximize_reuse disable -class core_wrapper
set_scan_configuration -chain_count 4 -clock_mixing mix_clocks
set_logicbist_configuration –chain_count 30 -clock clk_st -
pattern_counter_width 12
Set_streaming_compression_configuration -inputs 1 -outputs 1 -chain_count 30
set_dft_signal -port SI0 -type ScanDataIn -view spec -test_mode all
set_dft_signal -port SI1 -type ScanDataIn -view spec -test_mode all
create_test_protocol
dft_drc
preview_dft
set_dft_insertion_configuration -synthesis_optimization none
insert_dft
change_names -rules verilog -hier
write -f verilog -o design.v -h
write_test_protocol -test_mode LBIST -o logicbist.spf
```

```
### TetraMAX LogicBIST Script
##############################
read_net design.v
run_build des_unit
set_drc -seq_comp_jtag_lbist_mode light_lbist
run_drc logicbist.spf
set_faults -model stuck
run_atpg -auto -jtag_lbist { seed_values pattern_count capture_cycles }
write_patterns lbist_serial.stil -format stil -replace -unified -serial
```



7. **Handoff the Design**
   The design handoff is where files are written to disk that will be needed later on in the design process. Examples include design DDC, Verilog netlist, protocol file (for TetraMAX), test model (for the bottom-up flows), SCANDEF (for backend scan chain reordering), etc.
   See the DFTMAX™ Design-for-Test User Guide for more details.

   All the information required to run TetraMAX is stored in the DFTMAX protocol file
   This protocol is generated during DFT insertion, and can be written out as follows (same as in the snapshot example above in #6):

   write_test_protocol -out scancompress.spf -test_mode ScanCompression_mode.

   The ATPG flow in TetraMAX for DFTMAX is the same as for regular scan. Both DFTMAX and regular scan designs use the same fault diagnosis flow in TetraMAX

   The protocol is generated during DFTMAX Ultra insertion, and can be written out as follows
   write_test_protocol -output STREAMING.spf -test_mode ScanCompression_mode (also as the example above #6)

DFTMAX LogicBIST Support Documentation

The DFTMAX™ Design-for-Test User Guide Part V: DFTMAX LogicBIST Self-TestDFTMAX LogicBIST Application Notes available include:

Finding Optimal Seed Values for the LogicBIST PRPG
Article: TetraMAX - SCRIPT - 2220819 - 26 May 2017
https://solvnet.synopsys.com/retrieve/2220819.html

Setting the Seed and Signature Values in a LogicBIST Design
Article: DFT Compiler - SCRIPT - 2231010 - 17 Aug 2017
https://solvnet.synopsys.com/retrieve/2231010.html

Additional Design-for-Test Flows:

Use Case - DFT Compiler	Reference Guide
DFTMAX Compression	Design-for-Test User Guide Part III Chp17-Chp22

Use Case - DFT Compiler	Reference Guide
DFTMAX Ultra Compression	Design-for-Test User Guide Part IV Chp23-Chp29

Use Case - DFT Compiler	Reference Guide
DFTMAX LogicBIST Self-Test	Design-for-Test User Guide Part V Chp30-Chp34
	Application Note: Finding Optimal Seed Values for the LogicBIST PRPG TetraMAX https://solvnet.synopsys.com/retrieve/2220819.html
	Application Note: Setting the Seed and Signature Values in a LogicBIST Design DFT Compiler https://solvnet.synopsys.com/retrieve/2231010.html

Messages

The users of these tools should review any errors, warnings, and information related messages in the logfile and or displayed in the Shell or GUI.

IC test is composed of two primary approaches: functional testing and manufacturing testing. User functional testing should consist of a functional verification suite and use Formal Verification (such as Synopsys Formality).

7 Description of Tool Requirements of the Design-for-Test Insertion Flows

This section contains details about flow requirements of the design-for-test insertion flows

DFTMAX Compression Requirements

You need to consider both design and pin requirements when using DFTMAX compression. The following two topics describe these requirements.

Design Requirements

Designs that use DFTMAX compression are generally scan-replaced, that is, test-ready The supported DFTMAX flows include top-down and bottom-up methodologies. When starting with an RTL design, run the compile or compile_ultra command with the -scan option to bring your design into a test-ready state.

Compressed scan requires a pre-clock strobe. You should ensure that the test_default_strobe variable is set so that the strobe occurs before the active edges of the test clock waveforms. The default DFT Compiler test timing values meet this requirement.

DFTMAX Compression Scan Register Synchronization

DFT Compiler has the following requirements for scan synchronization:

- DFT Compiler checks whether synchronization is possible between head pipeline registers and scan chains.

  It also checks for synchronization between scan chains and tail pipeline registers. If there is a discrepancy, an error message is displayed and scan architecting or insertion is prevented.

- The clock signal triggering the lock-up element at the beginning of the chain is the inversion of the clock signal triggering the first flip-flop of the same chain. Similarly, the clock signal triggering the lock-up element at the end of the chain is the inversion of the clock signal triggering the last flip-flop of the chain.

DFTMAX Ultra Compression Requirements

To use DFTMAX Ultra scan compression:

- You must have the following cell types available for mapping when using the insert_dft command:
  - ❍ Level-sensitive latch

- ❍ Flip-flop with asynchronous reset

- You must use a precook strobe (which is the default). If you set the test_default_strobe variable, ensure that the strobe occurs before the active edges of the test clock waveforms.

This connection method has the following requirements:

- You must explicitly define the scan data signals with the set_dft_signal -partition all command. You cannot use this connection method with automatically created scan data signals.

Mixing DFTMAX and DFTMAX Ultra Compression Modes

You can mix DFTMAX and DFTMAX Ultra compression modes in the same design. However, note the following requirements:

- Both compression types cannot be active in the same test mode.
- User-defined DFTMAX Ultra test modes must be defined with the streaming_compression usage so that the tool can differentiate them from DFTMAX test modes defined with the scan_compression usage.

LogicBIST Requirements

The LogicBIST flow requires the following:

- You must have the Design Compiler, DFTMAX LogicBIST, and TetraMAX tools installed and licensed at your site.
- You must have an HDL Compiler license for compressed scan insertion.
- Blocks must be X-clean.
- The design must have four self-test signals (LBIST_EN, START, STATUS_0, STATUS_1), plus a scan-in and a scan-out that can be shared with non-LogicBIST scan modes.

8
Description of Tool Restrictions of the Design-for-Test Insertion Flows

This section contains details about tool limitations. Note that these restrictions do not supersede documented limitations and details in the DFT Compiler User Guides.

DFTMAX Compression Limitations and Known Issues

This topic covers current limitations and known issues associated with DFTMAX compression.

Post-DFT DRC Limitations

Post-DFT DRC is not supported when the following features are used:

- Integrating compressed scan cores
- Using implicit scan chains
- Performing reordering in the ASCII netlist flow

Current Limitations

The following features are currently not supported:

- You cannot use the existing scan flow to insert compression logic into a design that already has standard scan chains at the top level.
- You cannot use the `set_scan_path` command, unless it is used with a multiple test-mode specification.
- No graphical design rule checking (DRC) debugging support exists in the Synopsys Design Vision™ GUI for compressor design rule violations.

Limitations of DFT Partitions in Scan Compression Flow

The following limitation applies to DFT partitions in DFTMAX compression flows:

Pipelined Scan Data Limitations

These are the requirements and limitations for implementing pipeline registers in a compressed scan flow:

- For maximum observability, the head pipeline flip-flops must hold state during the capture cycle to avoid capturing unknown X values.
- Compressed scan does not support unbalanced pipelining across chains. The number of head pipeline stages does not need to match the number of tail pipeline stages, but all decompressor inputs and all compressor outputs—including any associated with compressed clock chains—must have the same pipeline depth.
- If you are using external (uncompressed) chains, such as external clock chains or other user-defined external chains, their pipeline depths must match other scan chains.
- Scan-enable pipelining is independent of compressor scan data pipelining. Scan-enable signals can have any number of pipeline stages; however, the logic must be such that the load_unload and capture

operations can be independently verified by DRC. For example, test_setup and/or load_unload preamble must set the design in shift mode, so that when the scan-enable signal is at the nonshift value, the flip-flops are able to capture the system data.

Any combinational logic between the scan ports and the pipeline registers must be sensitized to a known state.

In designs with OCC controllers, an ATE clock cannot be used to directly clock head or tail pipelined scan data registers:

- For DFT-inserted OCC controllers, when you specify the ATE clock as the pipeline register clock, the tool uses an OCC-controlled clock associated with the ATE clock.
- For details, see SolvNet article 2685005, “How Are OCC Clocks Chosen for Pipelined Scan Data Registers?”
- For user-defined OCC controllers, if the ATE clock is manually connected to the pipeline registers, no DRC violations will be reported, but incorrect ATPG patterns might be generated.
- Multiple test-mode operation with user-defined pipeline registers is not supported. However, multiple test-mode operation with automatically inserted pipeline registers is supported.
- User-defined pipeline registers and automatically inserted pipeline registers are mutually exclusive.

DFTMAX Ultra Limitations and Known Issues

The following DFT synthesis requirements and limitations apply to DFTMAX Ultra compression:

- The minimum compression ratio for DFTMAX Ultra compression is 3. For example, with five scan inputs and five scan outputs, a minimum of 15 internal chains must be used.
- In core integration flows, post-DFT DRC of test modes that contain active compressed scan cores is not supported.
- When DFT partitions are used, all partitions must use streaming DFT compression. Any partitions that contain uncompressed logic (top-level logic or standard scan cores) must have a codec configured using the `set_streaming_compression_configuration` command. However, scan logic can remain uncompressed by defining external chains.
- When you use the `-target` option of the `define_test_mode` command in the Mixed Insertion Integration core integration flow, a top-level codec is inserted in a test mode only when you target the top-level logic by including the name of the current design in the target list. You cannot insert a codec for targeted cores without also compressing the top-level logic, which includes any untargeted standard scan cores and any wrapped cores in outward-facing mode.
- When using the OCC feature, the clock chain must be implemented as an external clock chain (which is uncompressed and outside of the codec logic).
- You can mix DFTMAX and DFTMAX Ultra compression modes in the same design. However, both compression types cannot be active in the same test mode. Compression types other than DFTMAX and DFTMAX Ultra are not supported.
- When integrating pipelined cores, the top-level depths reported by the TEST-1433 and TEST-1434 information messages are incorrect, and DFT insertion does not abort if the total pipeline depth is smaller than the largest core-level depth.
- Synthesis of static-X chains is not supported.
- Scan groups, defined with the `set_scan_group` command, are not supported.
- Scan-through-TAP, which provides access to internal scan chains through the TDI and TDO ports of the IEEE Std 1149.1 test access ports (TAP), is not supported.
- Terminal lockup latches, enabled with the `-insert_terminal_lockup` option of the `set_scan_configuration` command, are not supported.

- `set_scan_path -edge` specifications are not supported.
- Flows other than top-down insertion (TDI) flows, streaming codecs cannot use scan-enable signals defined with the `-usage` option of the `set_dft_signal` command.
- You should define at least one scan-enable signal without a usage for proper codec insertion and operation.
- Modifications to the streaming compression IP blocks, such as adding inversions in codec scan paths, are not supported.

LogicBIST Limitations and Known Issues

The following requirements, limitations, and known issues apply to LogicBIST self-test:

- LogicBIST settings are not stored in .ddc files; you must apply any `set_logicbist_configuration` settings if you read a pre-DFT .ddc file back in.
- Designs that capture X values are not supported.
- Clock-gating cells require a dedicated ScanEnable signal defined with the `-usage {clock_gating}` option of the `set_dft_signal` command. If there is no such signal, no clock-gating testability logic is created. If the signal is defined with `-usage {scan clock_gating}`, scan cells are hooked up to the clock gating testability scan-enable signal, which is incorrect. TestMode signals cannot be used as clock-gating control signals.
- External (uncompressed) chains cannot be used in LogicBIST test modes.
- Clock-gating cells with pre-existing test-mode pin connections are not supported. All clock-gating cells must be identified prior to DFT insertion. For details, see the TEST-130 man page.
- If any scan chains have head flip-flops that are clocked on the trailing edge, you must enable beginning retiming registers by setting the `set_scan_configuration -add_test_retiming_flops` option to `begin_only` (or `begin_and_end`) to avoid scan timing issues from the leading-edge-clocked PRPG register.
- The LogicBIST test mode requires at least one user-defined scan-in signal; it is not automatically created.
- Sharing scan-in ports with functional ports is not supported.
- You can integrate a core that contains a LogicBIST test mode, but there is no automation provided to access the LogicBIST functionality at the integration level.
- For designs with OCC controllers,
  - A mix of non-OCC-controlled and internal OCC-controlled scan clock domains is not supported. In this case, you must also control the port-driven clocks with an OCC controller.
  - All OCC controllers must have the same clock chain length.
  - If you are using weighted clock capture groups, there must be at least seven clock chain bits across all clock chains in the design.
  - Synchronous OCC controllers are not supported.
    User-defined OCC controllers are not supported.
  - Existing OCC controllers inside DFT-inserted cores are not supported.
  - External clock chains cannot be used in LogicBIST test modes.
- If any DFT signals use bused ports, the bus indexes must be ordered highest-to-lowest.
- If you are using test points, you must specify an existing clock as the test point clock. You cannot use the DFT-created default test point clock.
- The following DFT features are not supported:
  - Pipelined scan data
  - Pipelined scan enable

- ❍ DFT partitions
- ❍ Multiple LogicBIST test modes
- ❍ Domain-based scan enable
- ❍ Terminal lock-up latches

• In TetraMAX ATPG,

- ❍ Diagnostics is not supported.
- ❍ The `-observe_file` option of the `run_atpg` command is not supported

Note the following requirements and limitations when using core wrapping with LogicBIST self-test:

- If you have functional ports reused as scan ports, you must isolate them with user-defined test points. The tool does not wrap these ports. For simplicity, it is best to avoid this where possible.
- If your design has feedthrough paths, restrictions apply to the use of shared wrapper cells on them. For details, see SolvNet article 2506549, “Feedthrough Path Caveats in Maximized-Reuse Wrapped LogicBIST Designs.”
- The tool-created wrapper clock (wrp_clk) is not controlled by the BIST clock controller (OCC or non-OCC) and cannot be used. To avoid this, ensure that it is not present in the wrapper preview report.

9
References

This section contains references for further information.

Standards, User Manuals, and Reference Methodology

- (ISO26262) International Organization for Standardization. *ISO 26262 Road Vehicles –Functional Safety*–. 1st Edition, 2011-11-15.
- (SGS-Cert) SGS-TÜV Certificate of ISO 26262 Compliance of the Tool Qualification Report for Design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST.
- (DFT Compiler-UM) DFT Compiler User Manuals
- (RMgen Flow) Synopsys® Design Compiler ® Reference Methodology Training

Revision Requirements

This completes the Tool Criteria Evaluation Report for the DFT Compiler design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and LogicBIST. This report must be revised if there are any changes or update of the tool itself, the system on which the tool is installed, the use of the tool, the scripts to generate the design-for-test insertion flows for DFTMAX, DFTMAX Ultra, and DFTMAX LogicBIST.