PrimeTime® ADV
Functional Safety Manual

March 2018, Revision 1.4

Copyright 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

Third-Party Links

Any links to third-party websites included in this document are for your convenience only. Synopsys does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

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

Document Control

Revision history

Version	Description	Date
1.0	First release of the document submitted for review.	15-Jan-2018
1.1	Added revision history, fixed template issues, made changes based on feedback.	06-Feb-2018
1.2	Fixed boilerplate changes from general feedback.	02-Mar-2018
1.3	Updates from certification review	09-Mar-2018
1.4	Added UPF as potential input file	14-Mar-2018

Contents

1
Customer Support

This section describes the customer support that is available through the Synopsys SolvNet® customer support website or by contacting the Synopsys support center.

Accessing SolvNet

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

To access the SolvNet site:

1. Go to the web page at https://solvnet.synopsys.com.
2. If prompted, enter your user name and password. (If you do not have a Synopsys user name and password, follow the instructions to register.)

If you need help using the site, click **Help** on the menu bar.

Contacting Synopsys Support

If you have problems, questions, or suggestions, you can contact the Synopsys support center in the following ways:

- Go to the Synopsys Global Support Centers site on synopsys.com. There you can find e-mail addresses and telephone numbers for Synopsys support centers throughout the world.
- Go to either the Synopsys SolvNet site or the Synopsys Global Support Centers site and open a case online (Synopsys user name and password required).

2
Scope of This Document

This section describes the scope of this document and defines terms used in this document.

Using This Document

The *PrimeTime Functional Safety Manual* describes the proper use of the PrimeTime tool in safety-related applications according to the ISO 26262 standard, and is intended to confirm the compliance of the PrimeTime tool to the standard when used in the context of a tool chain.

PrimeTime tool enables the user to perform full-chip, gate-level static timing analysis, which is an essential part of the design and analysis flow for chip designs. The tool exhaustively validates the timing performance of a design by checking all paths for timing violations, ensuring correct timing operation of the design.

Section 3 describes an overview of the ISO 26262-8, clause 11 and the approach adopted by Synopsys to comply with the requirements of the standard. Section 4 defines the general information such as where to find the latest documentation and installation requirements regarding the use of the PrimeTime tool as a software tool in the development of safety-related applications. Section 5 shows the high-level overview of the tool chain that this product belongs to. Section 6 details the safety-related requirements for safety-qualified use cases of the PrimeTime tool. Section 7 lists the known limitations of the use cases.

Specific documentation for performing design and analysis as part of an ISO 26262 compliant flow is provided in Section 3, Section 5, Section 6, Appendix A, and Appendix B of this document, the *PrimeTime Functional Safety Manual*.

Terms and Definitions

Term	**Definition**
AoU	Assumption of Use. An action that is assumed and required to be taken by the user of a software tool.

Term	**Definition**
ASIL	Automotive Safety Integrity Level. This is a risk classification scheme defined by the standard ISO 26262. The standard identifies four levels: ASIL A, ASIL B, ASIL C, and ASIL D. ASIL D dictates the highest integrity requirements on a product and ASIL A dictates the lowest.
Component	A part of an electronic system that implements a function in a vehicle. See also Part 1 of the standard ISO 26262 for the definition. The standard also refers to elements and items, but for the *PrimeTime Functional Safety Manual*, there is no difference.
CoU	Condition of Use. A condition of the design, software tool, design environment, or situation that is assumed and required to be fulfilled by the user.
CRM	Customer Relationship Management. Internal Synopsys database that manages customer STARs.
.db	Synopsys database, a binary file format used for storing designs, logic libraries, clock information, back-annotated parasitic information, and timing models.
Defect	Product nonconformance.
ECO	Engineering Change Order; a design change (cell sizing, buffer insertion, or cell replacement) that the PrimeTime tool has determined will correct a violation or improve the quality of the design.
Error	An error is a discrepancy between the actual and the specified or theoretically correct operation of an element. The root causes of an error can be manifold. In this document, the focus is on errors that are introduced or left undetected in a design, due to the malfunction in a software tool (e.g. generation of bad logic by a logic synthesis tool, failure of a static timing analysis tool to detect a timing violation).
False path	A timing path exception that specifies where a path exists in a design that should not be analyzed for timing, for example, a path between two multiplexed blocks that are never enabled at the same time.
Fault	An abnormal condition that can cause an element or item to fail.
Fault analysis	An analysis that determines the behavior of a system when a fault is introduced.

Term	**Definition**
FMEA	Failure Mode and Effects Analysis. An analysis that looks at different parts of a system, identifies ways the parts could fail, and determines the causes and effects of these potential failures.
GPD	Galaxy Parasitic Database, a Synopsys standard format for storing circuit parasitic data, supporting more types of information than the SPEF standard.
Hold constraint	A timing constraint that specifies how much time is necessary for data to be stable at the input of a device after the clock edge that clocks the data into the device, which enforces a minimum delay on a timing path relative to a clock.
HSPICE	A Synopsys-branded SPICE circuit simulator.
HyperScale	A PrimeTime hierarchical analysis method that analyzes the block-level and top-level portions of the design using separate runs and accurately handles the timing interfaces across hierarchical boundaries.
.lib	A standard text-based library file format, also known as the Liberty format.
LEF/DEF	Library Exchange Format and Design Exchange Format, a standard set of formats for exchanging library and design data between different tools.
Multicycle path	A timing path exception that specifies where a path is designed to take more than one clock cycle for the data to propagate from the startpoint to the endpoint.
PrimeTime	The Synopsys full-chip, gate-level static timing analysis tool. For definitions of terms specific to the PrimeTime tool, see the *Glossary* section of the *PrimeTime User Guide*.
PT	Abbreviation of the PrimeTime tool as used in the AoU and CoU IDs.
SDF	Standard Delay Format, a standard file format used to store circuit information for back-annotation, including delay information and timing checks.
SDC	Synopsys Design Constraints, a standard set of command syntax for specifying design constraints such as clocks, input delays, output delays, and timing exceptions.

Term	Definition
Setup constraint	A timing constraint that specifies how much time is necessary for data to be available at the input of a device before the clock edge that clocks the data into the device, which enforces a maximum delay on a timing path relative to a clock.
SI, signal integrity	The immunity of a signal or net against electromagnetic crosstalk effects, or the PrimeTime tool feature that analyzes these effects.
Software / software tool	The PrimeTime tool.
Software tool criteria evaluation	Analysis according to ISO 26262 to determine the required TCL of a software tool.
Software tool qualification	Means to create evidence, that a software tool with low or medium TCL is suitable to be used in the development of safety related products according to ISO 26262.
SolvNet	Synopsys customer support site.
SPEF	Standard Parasitic Exchange Format, a format used to store circuit parasitic information for back-annotation.
SPICE	Simulation Program with Integrated Circuit Emphasis, a time-based simulation tool that analyzes circuit operation at the level of transistors, capacitors, and resistors.
Standard	In this document, refers to *ISO 26262 Road Vehicles – Functional Safety*, 2011 and 2018 versions.
STAR	Synopsys Technical Action Request. A STAR documents and tracks a product Bug or Enhancement request (called a B-STAR or an E-STAR, respectively). It is stored in the Synopsys CRM database. Only Synopsys employees can access the CRM database. However, limited STAR information is available from SolvNet for customers who are associated with the user site of a STAR. Customer contacts are notified automatically when a STAR is filed or when its status changes.

Term	Definition
TCL	Tool confidence level, as defined by ISO 26262-8, clause 11. Note: The TCL of a software tool does not necessarily indicate whether the tool may malfunction or not. The TCL defines the confidence level that an error in the safety-related design, which is introduced or left undetected by the software tool, can be prevented or detected in subsequent steps of the development flow, before the erroneous safety-related design is released.
Tcl	Tool Command Language, a scripting language for PrimeTime commands.
TD	Tool error detection, as defined in ISO 26262-8, clause 11.
TI	Tool impact, as defined in ISO 26262-8, clause 11.
UPF	Unified Power Format, the IEEE 1801 standard for specifying multivoltage features of the design.
Use case	A use case is a specific way of using a software tool, that can be characterized by: - a limited set of tool functions and features that are used; - a set of restrictions and constraints that are regarded while using the tool; and - a specific goal to be achieved or output to be generated by using the software tool Use cases may be associated with different steps or phases in the design process, or they may describe alternative ways of using the tool for a specific design step.

3
Confidence in the Use of Software Tools According to ISO 26262-8, Clause 11

This section provides an overview of the ISO 26262-8, clause 11. It then describes the approach adopted by Synopsys to comply with the requirements of the standard, and how this is mapped to activities performed by Synopsys and the end user of the Synopsys tools.

Overview of ISO 26262-8, Clause 11

Synopsys EDA software tools contribute significantly to the design specification, implementation, integration, verification and validation of electrical and electronic (E/E) systems and components. If these E/E systems and components are used as part of a safety-related automotive product, an error in these systems or components could have severe consequences on functional safety. Such an error may arise as a result of unforeseen operating conditions or due to a fault introduced during product development, which in turn may be caused by a software tool malfunction. ISO 26262-8, clause 11 (Confidence in the Use of Software Tools) addresses this issue and specifies requirements and methods which aim to minimize the risk of faults in the developed product due to malfunctions of a software tool affecting the product’s functional safety.

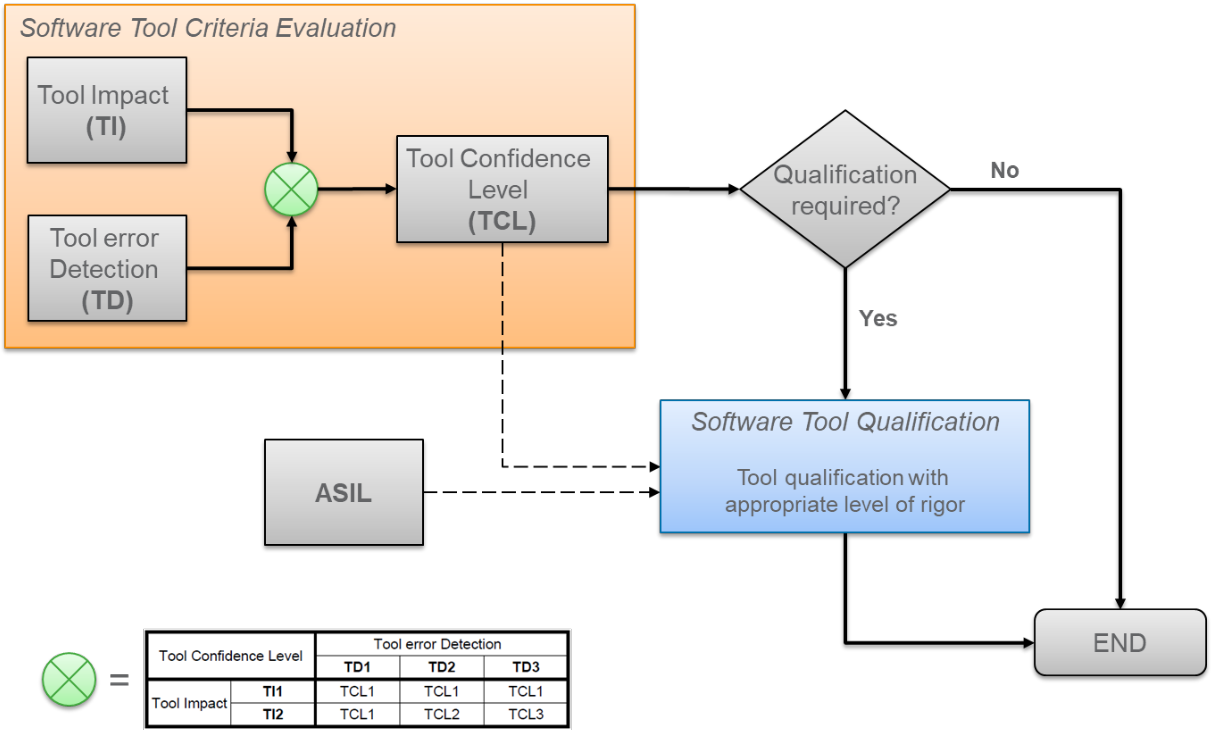
According to ISO 26262, to determine the required level of confidence in a software tool that is used in the development of a safety-related automotive product, the following criteria are evaluated:

- The possibility that the malfunctioning software tool and its corresponding erroneous output can introduce or fail to detect errors in a safety-related element being developed.
- The confidence in preventing or detecting such errors in its corresponding output.

This procedure is called Software Tool Criteria Evaluation, and it must be performed for all software tools that are involved in the development a safety-related element, resulting in a required Tool Confidence Level (TCL) for each software tool.

If the software tool criteria evaluation determines that a medium or high TCL is required, then appropriate Software Qualification methods must be applied, effectively reducing the risk of a critical software tool error. The choice of software qualification methods depends on the required TCL and the maximum ASIL of all the safety requirements allocated to the element developed using the software tool. However, if the software tool criteria evaluation determines that only a low TCL is required, then there is no need to apply such software qualification methods.

The software tool criteria evaluation and software tool qualification flow is summarized in Figure 1.


Software Tool Criteria Evaluation
Tool Impact (TI)
Tool error Detection (TD)
Tool Confidence Level (TCL)
Qualification required?
No
Yes
Software Tool Qualification
Tool qualification with appropriate level of rigor
ASIL
END
=

Tool Confidence Level		Tool error Detection		
		TD1	TD2	TD3
Tool Impact	TI1	TCL1	TCL1	TCL1
	TI2	TCL1	TCL2	TCL3

Figure 1: Software tool criteria evaluation and software tool qualification flow

Work Split between Synopsys and Tool Users

A software tool criteria evaluation must always be performed in the development environment of the final tool user, and in the context of the actual product development. It is in this context, where potential tool malfunctions, their effect on the safety-related product, and the effectiveness of prevention and detection measures must be analyzed.

However, the tool vendor can support the tool user by performing a software tool criteria evaluation (and, if required, a software tool qualification) on their own, based on assumed tool use cases and an assumed development environment. If the assumptions made by the tool vendor match the actual situation at the tool user, then the user can take over the evaluation (and qualification) results from the tool vendor. Besides significantly reducing the effort for the tool user, this approach can also result in a better quality for the software tool criteria evaluation and qualification, since the tool vendor typically has a more detailed understanding of the inner working and possible malfunctions of the software tool.

Synopsys has adopted exactly this approach, which is summarized in Figure 2.

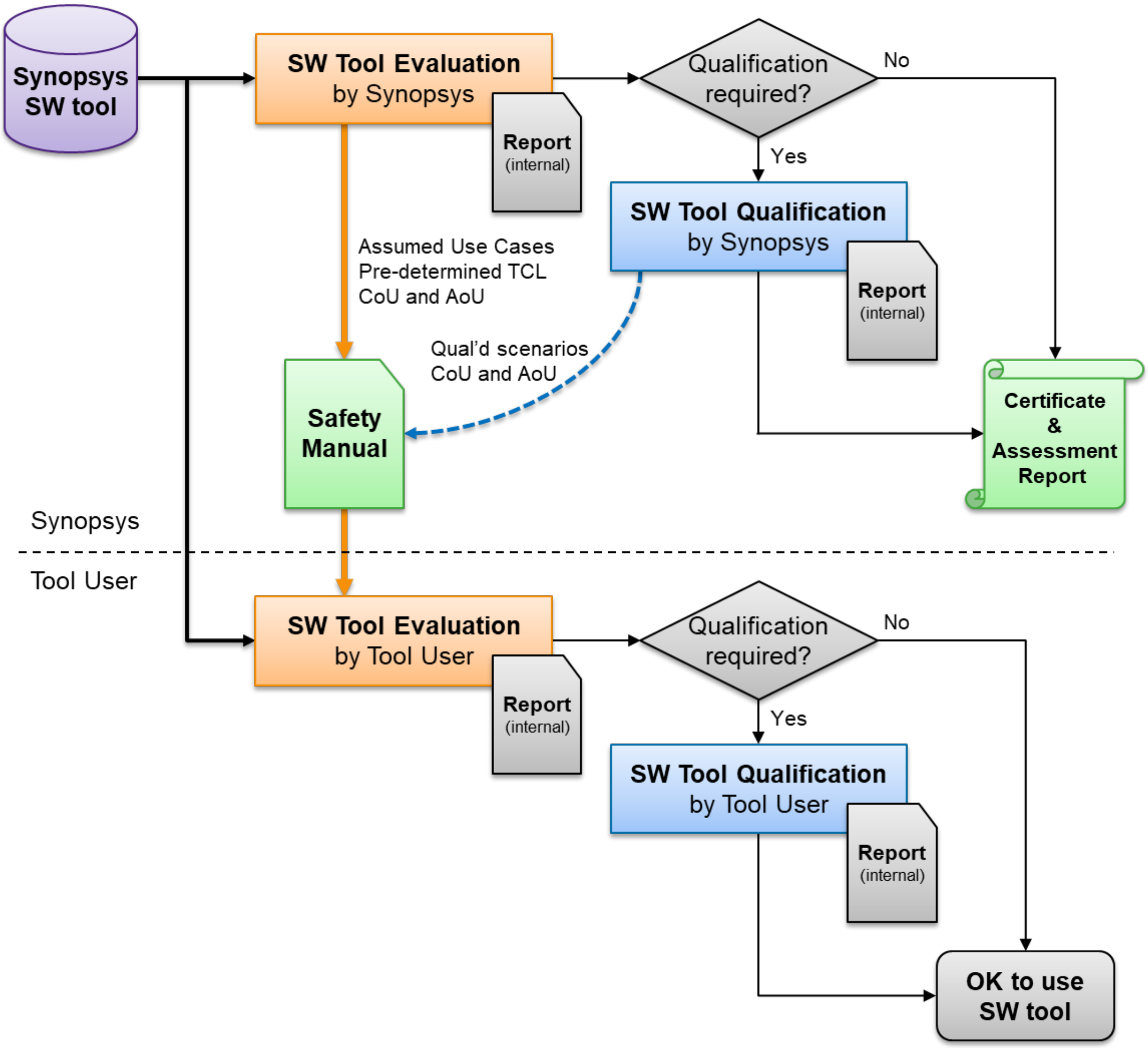


Figure 2: Work Split between Synopsys and Tool Users

Synopsys performs the following activities:

1. Software tool criteria evaluation
   - Identification of possible **use cases** for the software tool, together with required **inputs** and expected **outputs**
   - Specification of **conditions of use (CoU)** for each use case, related to the development environment in which the tool is assumed to be deployed, including tool usage procedures and constraints
   - Analysis of potential software tool **malfunctions**, and their effect on a safety-related product that is developed with this tool
   - Analysis of **prevention** and **detection measures** internal to the software tool, to avoid tool malfunctions, or to control and mitigate their effects
   - Specification of **assumptions of use (AoU)**, which are additional prevention and detection measures assumed to be performed by the end user of the tool

- Estimation of the **Tool Impact (TI)** for each malfunction, and the probability of **Tool error Detection (TD)** by the prevention and detection mechanisms (including assumptions of use)
- Determination of the required **Tool Confidence Level (TCL)** for each software tool malfunction, based on TI and TD
- Determination of the maximum TCL from all software tool malfunctions related to a use case. This is called the **pre-determined TCL** for the software tool use case
- Summary of the results in a software tool criteria evaluation report

2. Software tool qualification
    - If the pre-determined TCL indicates, that a medium (TCL2) or high (TCL3) tool confidence level is required for the software tool, then Synopsys may decide to perform a software tool qualification
    - The specific methods applied for tool qualification can vary for different tools and use cases, and they may include an evaluation of the software tool development process, the validation of the complete software tool, the validation of critical tool malfunctions with insufficient prevention and detection measures, or other methods
    - Summary of the qualification methods, procedures and results in a software tool qualification report

3. Safety manual for the software tool
    - The *PrimeTime Functional Safety Manual* (this document) is an important deliverable to the tool users, as it includes all end user-relevant information from the Synopsys software tool criteria evaluation and qualification
    - Software tool criteria evaluation related information, documented in Section 6, includes:
        - Description of software tool use cases
        - Description of the required inputs and expected outputs for each use case
        - Specification of conditions of use (CoU – conditions of the design, software tool, design environment, or situation that are assumed and required to be fulfilled by the user) for each use case
        - Specification of assumptions of use (AoU – actions that are assumed and required to be taken by the user of a software tool) for each use case
        - Pre-determined TCL for each use case
    - Software tool qualification related information (not required for this PrimeTime tool and therefore not included in this safety manual)
        - Description of the scope of the software tool qualification, including malfunctions and scenarios covered by the qualification
        - Specification of additional conditions of use (CoU) derived from the software tool qualification
        - Specification of additional assumptions of use (AoU) derived from the software tool qualification
    - Other information included in this safety manual
        - General information about the software tool needed by the tool user (see Appendix A)
        - Known limitations of the software tool, related to the described use cases as documented in Section 7

4. Certification and assessment report
    - Synopsys may decide to perform a functional safety assessment, to confirm the correctness, completeness and ISO 26262 conformance of the performed software tool criteria evaluation and qualification
    - Synopsys may also decide to achieve certification from an accredited third-party certification body, in addition to the functional safety assessment
    - The results of these activities are summarized in a functional safety assessment report and a certificate which can be viewed at exida Certificate for ISO 26262 Compliance

If the tool user wants to benefit from the work done by Synopsys, then according to the Figure 2 above, the user shall perform the following activities for each software tool:

1. Software tool criteria evaluation
    - Review and verify that the software tool criteria evaluation (and qualification) performed by Synopsys, as documented in the tool‘s Functional Safety Manual, matches the actual situation of the user’s product development process
        - Verify whether the actual use case(s) of the software tool match those evaluated by Synopsys
        - Verify whether the actual inputs and outputs are identical to or a sub-set of those as evaluated by Synopsys
        - Verify that all conditions of use (CoU) specified by Synopsys are met, or whether the development process can be adjusted to meet these CoU(s)
        - Verify that all assumptions of use (AoU) specified by Synopsys are met, or whether the development process can be adjusted to meet these AoU(s)
        - Verify that the pre-determined Tool Confidence Level (TCL) for the relevant use case(s) are TCL1, *or*
        - Verify that Synopsys has successfully performed an additional software tool qualification for all TCL2 and TCL3 scenarios to conclude that the tool is suitable to be used for the development of a safety-related element of the same or higher ASIL than required by the user
    - If all the verification steps described above are successful, then the results of the Synopsys software tool criterial evaluation (and qualification) are applicable to the tool user, which means:
        - The required TCL pre-determined by Synopsys can be taken over by the tool user for actual product development
        - If the pre-determined TCL is TCL1, then the tool can be used without the need to perform any additional software tool qualification
        - If the pre-determined TCL is TCL2 or TCL3, then the software tool qualification performed by Synopsys is sufficient, and the tool can be used without the need for further software tool qualification by the end user
    - All of the steps above must be documented in a software tool criteria evaluation report, including evidence for the successful conclusion of all verification steps, which may include reference to the Synopsys Functional Safety Manual, and optionally, to the Synopsys certification and assessment report

2. Software tool qualification
    - If any of the verification steps described above as part of the tool user’s software tool criteria evaluation fails (e.g. different use case, CoU or AoU cannot be met, pre-determined TCL is not TCL1 and Synopsys has not performed a software tool qualification), then the user must perform his/her own software tool qualification
    - The specific methods applied for tool qualification are decided and planned by the tool user -- Synopsys does not recommend any specific methods or procedures
    - The summary of the qualification methods, procedures and results shall be documented in a software tool qualification report

4
PrimeTime Description

This section provides a general description regarding the use of the PrimeTime tool as a software tool in the development of safety-related applications and describes where to get the latest product documentation and the runtime environment required to use the PrimeTime tool.

Coverage

The *PrimeTime Functional Safety Manual* is intended to be used starting with version 2017.12 and later versions of the PrimeTime tool per the use cases presented in this document. In general, unless otherwise noted, the failure modes and detection mechanisms noted in the use cases presented in Section 6 are tool version independent.

Compliance with ISO 26262

The PrimeTime tool can be used in the development of safety-related elements according to ISO 26262, with allocated safety requirements up to a maximum Automotive Safety Integrity Level D (ASIL D), if the tool is used in the context of a tool chain and in compliance with this document, the *PrimeTime Functional Safety Manual*.

See the exida Certificate for ISO 26262 Compliance of Synopsys PrimeTime when used in a tool chain flow.

Product Documentation and Support

Comprehensive documentation for using the PrimeTime tool is provided on SolvNet. The latest documentation for the PrimeTime tool can be accessed at the PrimeTime Suite Online Help page on SolvNet.

Specific documentation for performing design and analysis as part of an ISO 26262 compliant flow is provided in Section 3, Section 5, Section 6 and Appendix A of this document, the *PrimeTime Functional Safety Manual*.

Synopsys provides online customer support for the PrimeTime tool. See Section 1 for more information.

Installation and Supported Platforms

The installation of the PrimeTime tool must follow the guidelines in the *Synopsys® Installation Guide* as well as the specific *PrimeTime Installation Notes* document.

Users are required to download the tool executable and INSTALL_README from the SolvNet site at https://solvnet.synopsys.com/DownloadCenter/dc/product.jsp.

Supported platforms and operating systems requirements:

- For installation instructions, see the *Synopsys® Installation Guide* at https://www.synopsys.com/install.
- For the latest supported binary-compatible hardware platform or operating system, including required operating system patches, see https://www.synopsys.com/qsc.
- If updates (including security patches) to computing environments (including operating systems) are backward compatible with previous versions of the computing environment used to test the PrimeTime tool, the results of the testing performed by Synopsys using such previous versions are applicable.

Additional information:

- For information about the compute platforms roadmap, go to https://www.synopsys.com/support/licensing-installation-computeplatforms/compute-platforms/compute-platforms-roadmap.html.
- For platform notices, go to https://www.synopsys.com/support/licensing-installation-computeplatforms/compute-platforms/platform-notice.html.
- For information regarding the license key retrieval process, go to https://solvnet.synopsys.com/smartkeys/smartkeys.cgi.

User Competence

To properly use the PrimeTime tool, a user must have a good understanding and working knowledge of the following:

- Electrical engineering and circuit design
- The ISO 26262 standard
- Documentation of the PrimeTime tool, such as the *PrimeTime User Guide*, at https://solvnet.synopsys.com/dow_retrieve/latest/dg/ptolh/Content/ni/ptug/ni/ptug.pdf
- This Functional Safety Manual
- The published list of safety-related defects for the PrimeTime tool available at PrimeTime ADV Safety-Related Issues Master List
- Applicability of the PrimeTime tool in the overall tool chain

Managing Known Safety-Related Defects

Synopsys maintains current information for every reported defect through STARs. The PrimeTime team evaluates each reported issue for potential impact on functional safety.

A list of all known safety-related defects for each release of PrimeTime is available on a SolvNet knowledge base article and is referenced from the *PrimeTime Release Notes*.

PrimeTime users must assess, as part of their own software tool criteria evaluation, the potential impact of the known safety-related defects in their design and must ensure mitigation of any relevant safety-related defects.

Managing New Releases

Synopsys can release new versions of the PrimeTime tool at any time to extend its functionality or to fix defects. When a new version is available, notification is posted on the SolvNet site. A subscription service is available for users to be notified of any new product releases.

When installing a new version of the PrimeTime tool, users must evaluate the impact of any known safety-related defects in their design by checking the accompanying *PrimeTime Release Notes* for the following:

- Any changes that apply to safety-related use cases
- List of known safety-related defects in the new version of the PrimeTime tool

In addition, users must refer to the latest version of this document, the *PrimeTime Functional Safety Manual*, available with the product release contents.

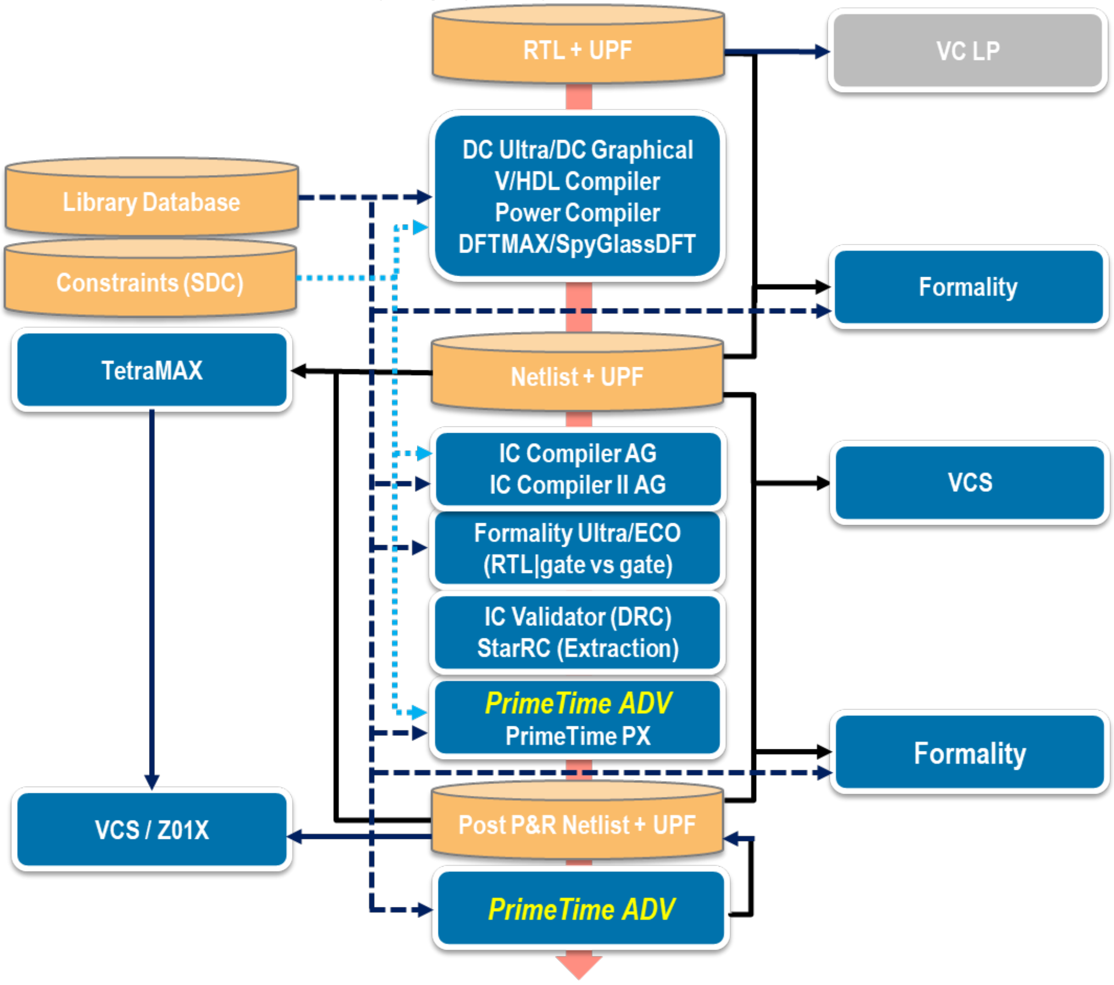
5
Synopsys Digital Tool Chain

This section provides an overview of where the PrimeTime is used in the tool chain.

The ISO 26262 standard provides a methodology and requirements for software tool criteria evaluation and qualification (see ISO 26262-8, clause 11). It applies to software tools used for the development of safety-related designs where it is essential that the tool operates correctly without introducing or failing to detect errors in the safety-related design.

The suitability of a software tool to be used in the development of a safety-related design is determined in the software tool criteria evaluation, which results in a Tool Confidence Level (TCL): a level of confidence that the software tool does not introduce or fail to detect an error in the design without being noticed, and mitigated before the design is released as a safety-related product. This evaluation is best performed in the context of the overall software tool chain and development flow, in which the individual software tool is used. The following high-level diagram reflects the tool chain for which the PrimeTime tool is applicable.

Synopsys Digital Tool Chain



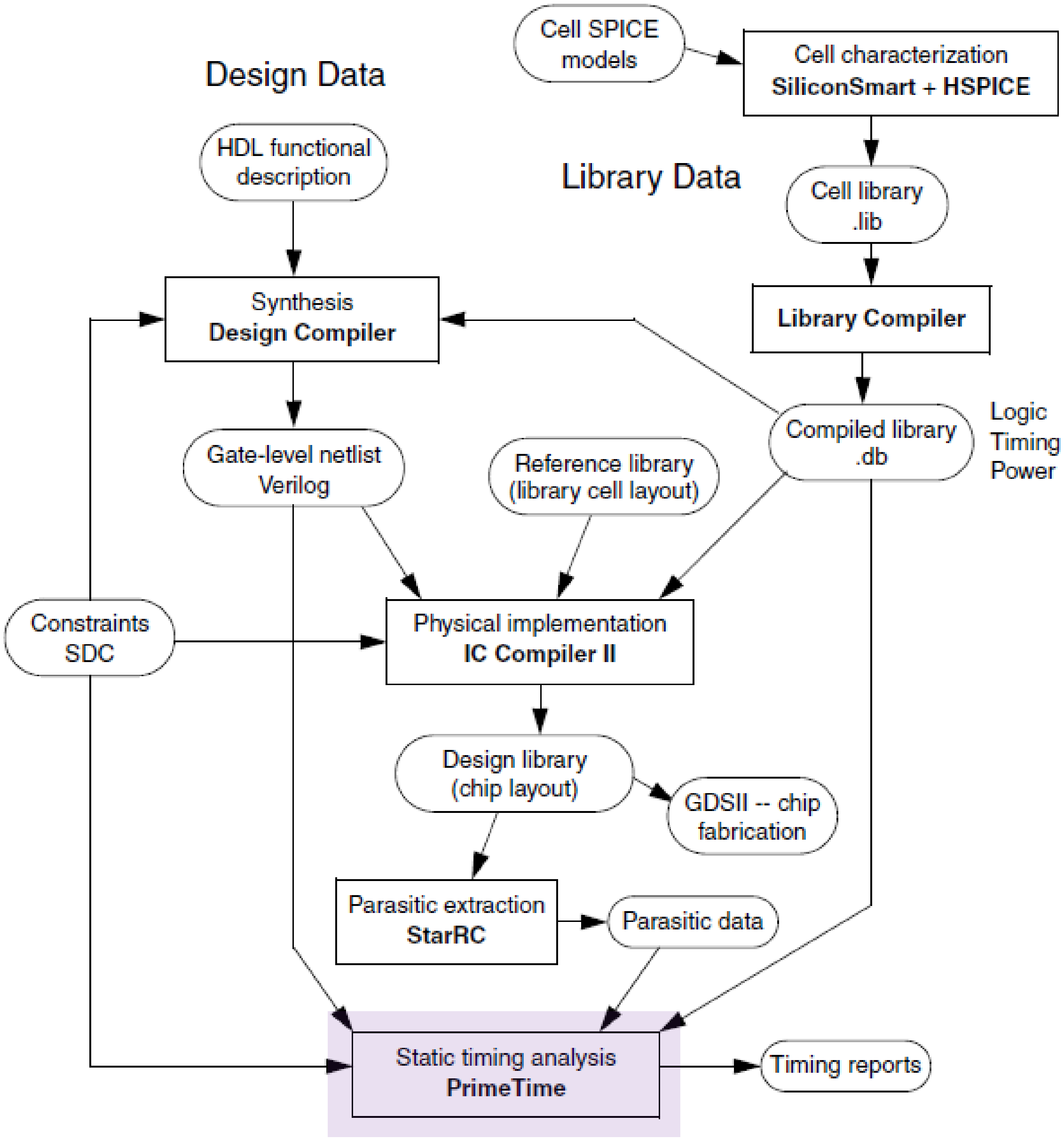
6
Use Cases

This section describes the safety-qualified use cases of the PrimeTime tool. Users should also perform TCL determination based on their specific Use Cases.

The PrimeTime tool performs full-chip, gate-level static timing analysis, which is an essential part of the design and analysis flow for chip designs. The tool exhaustively validates the timing performance of a design by checking all paths for timing violations, ensuring correct timing operation of the design. It is strictly an analysis tool, not a design tool, except for the generation of engineering change orders (ECOs) described later in this document.

Use Case 1: Static Timing Analysis

In the static timing analysis use case, the goal is to exhaustively analyze the timing characteristics of the design to find any places where timing errors can occur while the chip is running at the target clock rate. The tool determines the worst-case timing situations under all possible stimulus conditions. The following diagram shows a typical data flow leading up to the PrimeTime tool.



In this use case, the PrimeTime tool uses and generates the following main inputs and outputs.

- Inputs:
  - Gate-level netlist (.v)
  - Logic library (.db, .lib)
  - Timing constraints (.sdc)
  - Parasitics (SPEF, GPD)
  - Power intent (UPF)

  - Tcl scripts for PrimeTime configuration and execution of the static timing analysis
- Expected outputs:
  - Timing reports
  - Log files (ASCII)

For this use case of the PrimeTime tool, the following conditions of use (constraints for the design and design environment, recommended procedures for the tool usage, etc.) shall be met:

- CoU-PT-001: User shall review all error and warning messages and take appropriate action.
- CoU-PT-002: User shall follow the PrimeTime Reference Methodology or use equivalent scripts.
- CoU-PT-003: Exception settings (including `set_false_path`, `set_multicycle_path`, `set_min_delay`, and `set_max_delay`) shall be reviewed and used with caution.
- CoU-PT-004: Margin-of-safety adjustment settings (including `set_context_margin`, `set_extract_model_margin`, `set_noise_margin`, `set_path_margin`, `set_timing_derate`) shall be reviewed to prevent optimism.
- CoU-PT-005: For the final signoff static timing analysis run, Tcl script-based batch mode execution shall be used, without interactive command line entry or GUI manual command entry. Tcl scripts and log files shall be retained as sign-off records.

For this use case of PrimeTime, the following assumptions of use (required actions to be taken by the tool user to prevent or detect design errors due to possible tool malfunctions) shall be met:

- AoU-PT-001: User shall review the log files and timing reports for expected execution steps and expected results, and respond appropriately to all warning and error messages.
- AoU-PT-002: User shall check that all outputs are generated with an up-to-date timestamp.
- AoU-PT-003: When using a new library for the first time, user shall verify PrimeTime critical path timing results using a SPICE simulator such as the HSPICE tool.
- AoU-PT-004: User shall compare the timing reported by the PrimeTime tool against timing reported by the physical implementation tool (such as IC Compiler or IC Compiler II). Any significant timing discrepancy shall be reviewed and based on user judgement, shall be run in a SPICE tool (such as HSPICE).

All analyzed failure modes and the corresponding prevention, detection and mitigation measures (including conditions and assumptions of use listed above) are independent of the exact PrimeTime tool version.

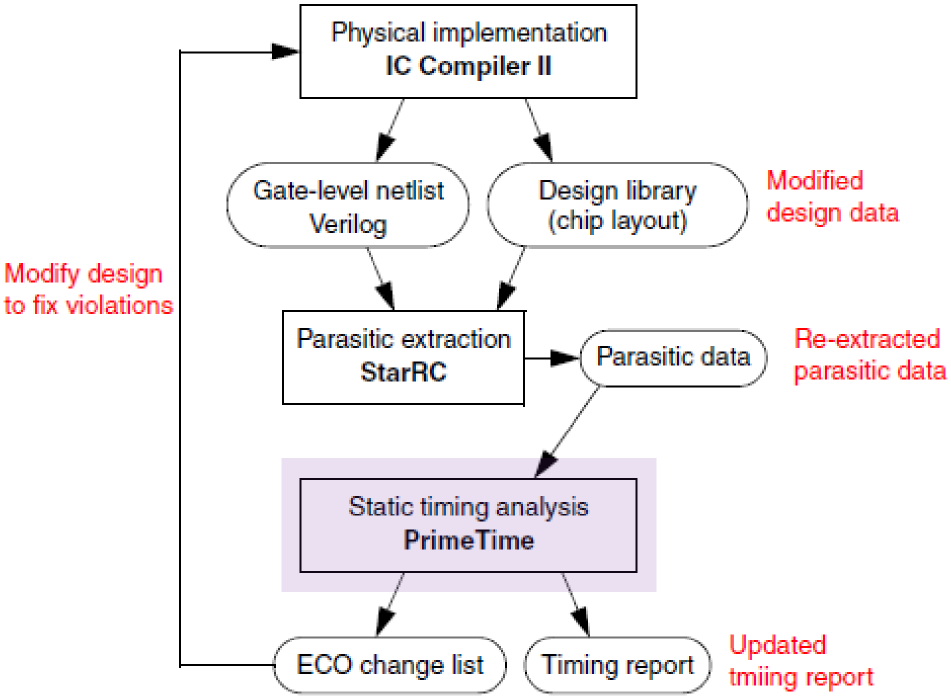
A software tool criteria evaluation performed by Synopsys according to ISO 26262-8, clause 11, which assumes the fulfillment of all conditions of use (CoU) and assumptions of use (AoU) as described above, results in a required tool confidence level:

TCL1 for PrimeTime Use Case 1 – Static Timing Analysis.

In this case, no further activities for software tool qualification are required.

Use Case 2: PrimeTime Engineering Change Order (ECO)

In the engineering change order (ECO) use case, the goal is to fix timing failures, improve margins against failure, and reduce power consumption of the chip. The PrimeTime tool does not itself modify the design database used for manufacturing the chip. Instead, it provides fixing guidance to an implementation tool such as IC Compiler II. The other tool performs the changes to fix or improve the design. To ensure safety, the modified design must undergo another parasitic extraction by the StarRC tool and timing analysis by the PrimeTime tool. The following diagram shows the overall ECO flow.



The PrimeTime tool, in the course of analysis, finds timing violations, design rule violations, and opportunities for reducing power consumption. It modifies its own local model of the design stored in RAM and re-analyzes the model to see if the design improvement is achieved.

After the PrimeTime tool determines the necessary set of design changes, it writes out the changes in the form of a script for the targeted tool, such as IC Compiler II. This is called a *change list file* or *change list script*. The IC Compiler II tool can execute the script to make the changes in the actual design database.

Note: The PrimeTime tool performs static timing analysis while it examines potential changes to the design for ECOs. Therefore, the failure modes, CoUs, and AoUs of Use Case 1, Static Timing Analysis, also apply to the ECO use case. Additional AoUs apply specifically to the ECO use case.

In the PrimeTime ECO use case, the tool uses and generates the following main inputs and outputs.

- Inputs:
  - Physical input data (LEF/DEF or GPD files)
  - Gate-level netlist (.v)
  - Logic library (.db, .lib)
  - Timing constraints (.sdc)
  - Parasitics (SPEF, GPD)
  - Power intent (UPF)
  - Tcl scripts for configuration and execution
- Expected outputs:
  - Change list files
  - Timing reports
  - Log files (ASCII)

For this use case of the PrimeTime tool, the following conditions of use (constraints for the design and design environment, recommended procedures for the tool usage, etc.) shall be met:

- CoU-PT-001: User shall review all error and warning messages and take appropriate action.
- CoU-PT-002: User shall follow the PrimeTime Reference Methodology or use equivalent scripts.
- CoU-PT-003: Exception settings (including `set_false_path`, `set_multicycle_path`, `set_min_delay`, and `set_max_delay`) shall be reviewed and used with caution.
- CoU-PT-004: Margin-of-safety adjustment settings (including `set_context_margin`, `set_extract_model_margin`, `set_noise_margin`, `set_path_margin`, `set_timing_derate`) shall be reviewed to prevent optimism.

- CoU-PT-005: For the final signoff static timing analysis run, Tcl script-based batch mode execution shall be used, without interactive command line entry or GUI manual command entry. Tcl scripts and log files shall be retained as sign-off records.

For this use case of PrimeTime, the following assumptions of use (required actions to be taken by the tool user to prevent or detect design errors due to possible tool malfunctions) shall be met:

- AoU-PT-001: User shall review the log files and timing reports for expected execution steps and expected results, and respond appropriately to all warning and error messages.
- AoU-PT-002: User shall check that all outputs are generated with an up-to-date timestamp.
- AoU-PT-003: When using a new library for the first time, user shall verify PrimeTime critical path timing results using a SPICE simulator such as the HSPICE tool.
- AoU-PT-004: User shall compare the timing reported by the PrimeTime tool against timing reported by the physical implementation tool (such as IC Compiler or IC Compiler II). Any significant timing discrepancy shall be reviewed and based on user judgement, shall be run in a SPICE tool (such as HSPICE).
- AoU-PT-005: User shall review the ECO log files and change script output for completeness and verify that they contain the expected execution steps and ECO results.
- AoU-PT-006: User shall verify that the implementation tool can read in and run the ECO change script, and shall respond appropriately to all warning and error messages from the tool.
- AoU-PT-007: After ECO fixing by the physical implementation tool, user shall rerun PrimeTime timing analysis on the modified design and verify that all timing violations are fixed.
- AoU-PT-008: User shall perform formal verification (using the Formality tool) of the ECO-modified netlist versus the original netlist in the physical implementation tool.
- AoU-PT-009: After all incremental and reduced-resource ECO fixing cycles are complete, user shall run a final-signoff full-chip re-extraction using the StarRC tool and rerun a full-chip timing analysis.
- AoU-PT-010: User shall examine the time stamps of the design data files from the physical implementation tool to verify that the design was not changed while the ECO was being generated by the PrimeTime tool.
- AoU-PT-011: User shall review physical implementation tool logs to confirm relevant ECO changes have been applied and that the log file contains the expected execution steps.
- AoU-PT-012: User shall perform a final signoff analysis that passes all constraints (`report_constraint`).

All analyzed failure modes and the corresponding prevention, detection and mitigation measures (including conditions and assumptions of use listed above) are independent of the exact PrimeTime tool version.

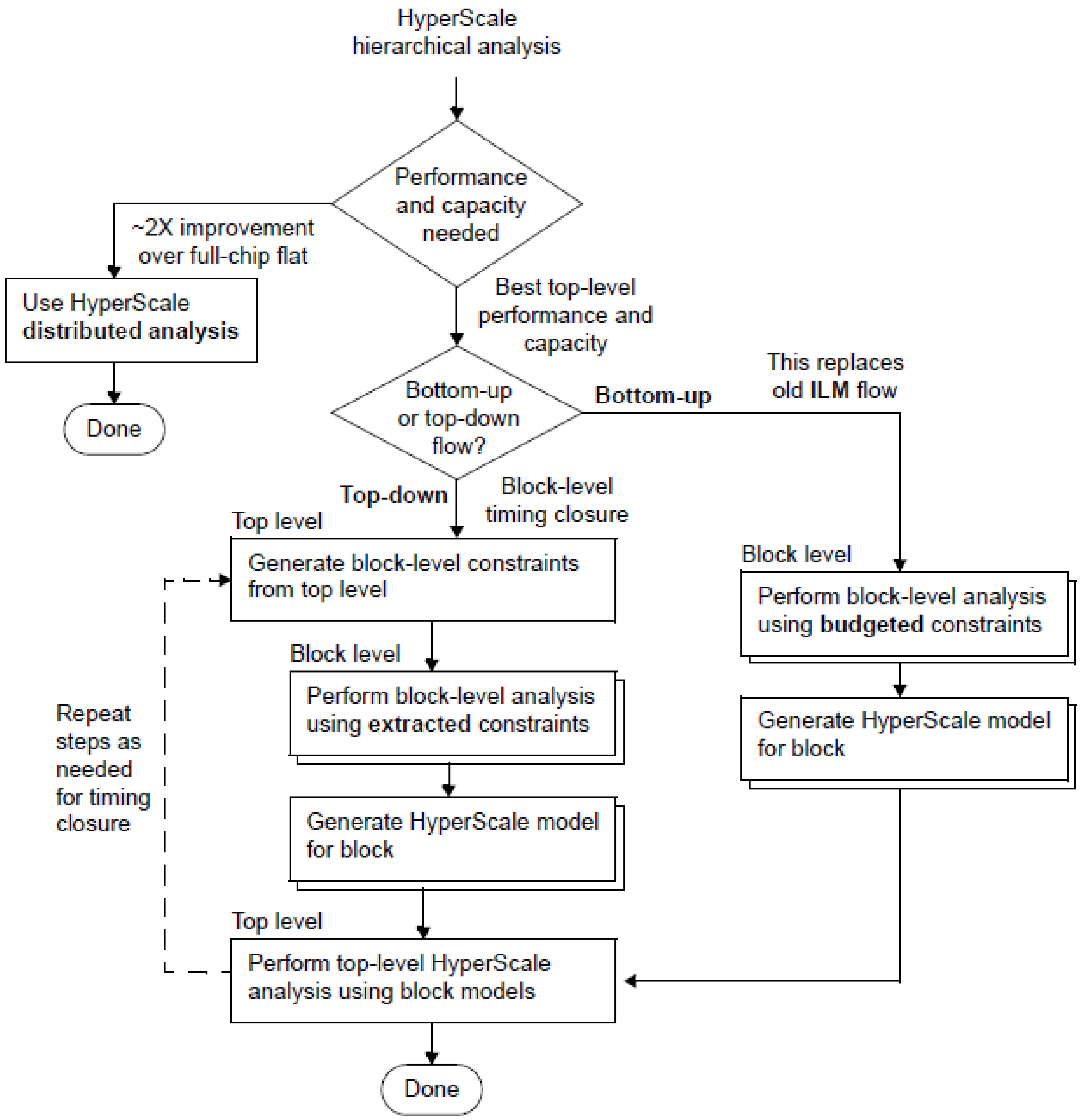
A software tool criteria evaluation performed by Synopsys according to ISO 26262-8, clause 11, which assumes the fulfillment of all conditions of use (CoU) and assumptions of use (AoU) as described above, results in a required tool confidence level:

TCL1 for PrimeTime Use Case 2 – PrimeTime Engineering Change Order

In this case, no further activities for software tool qualification are required.

Use Case 3: Hierarchical Analysis

In the hierarchical timing analysis use case, the goals are the same as for the static timing analysis use cases, but with the addition of techniques that analyze the block-level and top-level portions of the design using separate runs. These techniques are context characterization, generation and usage of block timing models, and HyperScale analysis. The following diagram shows the various HyperScale hierarchical analysis flows.


HyperScale hierarchical analysis
Performance and capacity needed
~2X improvement over full-chip flat
Use HyperScale distributed analysis
Done
Best top-level performance and capacity
Bottom-up or top-down flow?
Bottom-up
This replaces old ILM flow
Top-down
Block-level timing closure
Top level
Generate block-level constraints from top level
Block level
Perform block-level analysis using budgeted constraints
Generate HyperScale model for block
Block level
Perform block-level analysis using extracted constraints
Generate HyperScale model for block
Repeat steps as needed for timing closure
Top level
Perform top-level HyperScale analysis using block models
Done

Note: The failure modes, CoUs, and AoUs of Use Case 1, Static Timing Analysis, also apply to Use Case 3, Hierarchical Analysis. Additional CoUs and AoUs apply specifically to the hierarchical analysis use case.

In this use case, the PrimeTime tool uses and generates the following main inputs and outputs.

- Inputs:
    - Physical input data (LEF/DEF or GPD files)
    - Gate-level netlist (.v)
    - Logic library (.db, .lib)
    - Timing constraints (.sdc)
    - Parasitics (SPEF, GPD)
    - Power intent (UPF)
    - Tcl scripts for configuration and execution
- Expected outputs:
    - Change list files
    - Timing reports
    - Log files (ASCII)

For this use case of the PrimeTime tool, the following conditions of use (constraints for the design and design environment, recommended procedures for the tool usage, etc.) shall be met:

- CoU-PT-001: User shall review all error and warning messages and take appropriate action.
- CoU-PT-002: User shall follow the PrimeTime Reference Methodology or use equivalent scripts.
- CoU-PT-003: Exception settings (including `set_false_path`, `set_multicycle_path`, `set_min_delay`, and `set_max_delay`) shall be reviewed and used with caution.
- CoU-PT-004: Margin-of-safety adjustment settings (including `set_context_margin`, `set_extract_model_margin`, `set_noise_margin`, `set_path_margin`, `set_timing_derate`) shall be reviewed to prevent optimism.
- CoU-PT-005: For the final signoff static timing analysis run, Tcl script-based batch mode execution shall be used, without interactive command line entry or GUI manual command entry. Tcl scripts and log files shall be retained as sign-off records.
- CoU-PT-006: For top-level analysis in a hierarchical analysis flow, user shall ensure that the correct block models generated by block-level analysis are being used to represent the timing of lower-level blocks.
- CoU-PT-007: For block-level analysis in a hierarchical analysis flow, user shall ensure that the correct block context information generated by top-level analysis is used to represent the external timing requirements for the block.

For this use case of PrimeTime, the following assumptions of use (required actions to be taken by the tool user to prevent or detect design errors due to possible tool malfunctions) shall be met:

- AoU-PT-001: User shall review the log files and timing reports for expected execution steps and expected results, and respond appropriately to all warning and error messages.
- AoU-PT-002: User shall check that all outputs are generated with an up-to-date timestamp.
- AoU-PT-003: When using a new library for the first time, user shall verify PrimeTime critical path timing results using a SPICE simulator such as the HSPICE tool.
- AoU-PT-004: User shall compare the timing reported by the PrimeTime tool against timing reported by the physical implementation tool (such as IC Compiler or IC Compiler II). Any significant timing discrepancy shall be reviewed and based on user judgement, shall be run in a SPICE tool (such as HSPICE).
- AoU-PT-013: User shall review block model and block context data directories/files listed in the log files to ensure that the correct data is being used for analysis.
- AoU-PT-014: For final signoff analysis, user shall perform a full-chip flat analysis if memory and runtime resources allow it, or run multiple top-level analyses using flattened block data for selected blocks.
- AoU-PT-015: User shall examine the time stamps of the block model and context files to verify that the correct files are being used.

All analyzed failure modes and the corresponding prevention, detection and mitigation measures (including conditions and assumptions of use listed above) are independent of the exact PrimeTime tool version.

A software tool criteria evaluation performed by Synopsys according to ISO 26262-8, clause 11, which assumes the fulfillment of all conditions of use (CoU) and assumptions of use (AoU) as described above, results in a required tool confidence level:

TCL1 for PrimeTime Use Case 3 – Hierarchical Analysis

In this case, no further activities for software tool qualification are required.

7
Limitations of Use Cases

This section describes all known limitations of the use cases mentioned in the previous section.

All known safety-related issues for the PrimeTime tool are listed in the PrimeTime ADV Safety-Related Issues Master List available on SolvNet.

Each release of the PrimeTime tool may contain hidden, undocumented features for testing or evaluation purposes, known as “Limited Customer Availability” (LCA) features. Use LCA features only for testing and evaluating the proposed new features, not for production work.

Appendix A
Software Tool Information

This section provides general information about the PrimeTime software tool, which is needed by the tool user for performing his/her software tool criteria evaluation.

The following information about PrimeTime is required according to ISO 26262-8, for the planning of the usage of a software tool (clause 11.4.4) and the preparation of the own software tool criteria evaluation (clause 11.4.5).

Please note that some of the information below provided by Synopsys simply needs to be confirmed by the tool user and can be used without modification. Other information must be completed or updated by the tool user to reflect his/her actual situation.

Required Info	**Tool Information**	**Reference / Comment**
Tool vendor	Synopsys, Inc.	ISO 26262-8, 11.4.4.1.a
Tool name and version	PrimeTime	ISO 26262-8, 11.4.4.1.a To determine tool version, use: `report_version -options`
Tool use cases		ISO 26262-8, 11.4.4.1.c ISO 26262-8, 11.4.5.1.a To be completed by the tool user. Align with / verify against use cases described in Section 6 of this document.
Tool inputs and expected outputs		ISO 26262-8, 11.4.5.1.b To be completed by the tool user. Align with / verify against inputs and outputs described in Section 6 of this document.
Tool configuration and constraints		ISO 26262-8, 11.4.4.1.b ISO 26262-8, 11.4.5.1.c To be completed by the tool user. Align with / verify against CoU for the use cases described in Section 6 of this document.

Required Info	**Tool Information**	**Reference / Comment**
Tool environment (OS)	Refer to the PrimeTime Installation Notes at https://solvnet.synopsys.com/DownloadCenter. Click the PrimeTime Suite tool name, the release number, and then "View installation guide" for tool version-specific OS support.	ISO 26262-8, 11.4.4.1.d To be completed by the tool user. Align with / verify against the OS version evaluated by Synopsys. To determine Linux version, use: `uname -osr`
Tool environment (CAD tool chain)		ISO 26262-8, 11.4.4.1.d To be completed by the tool user. To determine name and version of your tool chain, please consult your CAD department.
Maximum ASIL	ASIL D	ISO 26262-8, 11.4.4.1.e
Tool qualification methods	Not applicable	ISO 26262-8, 11.4.4.1.f Software tool qualification is not required for PrimeTime
User manual and other usage guide documents	PrimeTime Suite Online Help PrimeTime User Guide PrimeTime Reference Methodology	ISO 26262-8, 11.4.4.2.a – d Tool user to include a link to these documents (Synopsys SolvNet or local copy), and to add any additional company-internal tool usage guidelines.
Known software tool malfunctions, and appropriate work arounds ...	For limitations, refer to Section 7 of this document: PrimeTime ADV Safety-Related Issues Master List	ISO 26262-8, 11.4.4.2.e Tool user to include a link to these documents (Synopsys SolvNet or local copy), and to add any additional company-internal work around descriptions.
Measures for the detection of tool malfunctions ...		ISO 26262-8, 11.4.4.2.f To be completed by the tool user. Align with / verify against AoU for the use cases described in Section 6 of this document.

Appendix B
Complete List of CoU and AoU IDs

The complete list of Conditions of Use (CoU) for PrimeTime is in the table below. CoU defines a condition of the design, software tool, design environment, or situation that is assumed and required to be fulfilled by the user.

ID	Description
CoU-PT-001	User shall review all error and warning messages and take appropriate action.
CoU-PT-002	User shall follow the PrimeTime Reference Methodology or use equivalent scripts.
CoU-PT-003	Exception settings (including `set_false_path`, `set_multicycle_path`, `set_min_delay`, and `set_max_delay`) shall be reviewed and used with caution.
CoU-PT-004	Margin-of-safety adjustment settings (including `set_context_margin`, `set_extract_model_margin`, `set_noise_margin`, `set_path_margin`, `set_timing_derate`) shall be reviewed to prevent optimism.
CoU-PT-005	For the final signoff analysis run, Tcl script-based batch mode execution shall be used, without interactive command line entry or GUI manual command entry. Tcl scripts and log files shall be retained as signoff records.
CoU-PT-006	For top-level analysis in a hierarchical analysis flow, user shall ensure that the correct block models generated by block-level analysis are being used to represent the timing of lower-level blocks.
CoU-PT-007	For block-level analysis in a hierarchical analysis flow, user shall ensure that the correct block context information generated by top-level analysis is used to represent the external timing requirements for the block.

The complete list of Assumptions of Use (AoU) for PrimeTime is in the table below. AoU defines an action that is assumed and required to be taken by the user of a software tool.

ID	Description
AoU-PT-001	User shall review the log files and timing reports for expected execution steps and expected results, and respond appropriately to all warning and error messages.
AoU-PT-002	User shall check that all outputs are generated with an up-to-date timestamp.
AoU-PT-003	When using a new library for the first time, user shall verify PrimeTime critical path timing results using a SPICE simulator such as the HSPICE tool.
AoU-PT-004	User shall compare the timing reported by the PrimeTime tool against timing reported by the physical implementation tool (such as IC Compiler or IC Compiler II). Any significant timing discrepancy shall be reviewed and based on user judgement, shall be run in a SPICE tool (such as HSPICE).
AoU-PT-005	User shall review the ECO log files and change script output for completeness and verify that they contain the expected execution steps and ECO results.
AoU-PT-006	User shall verify that the implementation tool can read in and run the ECO change script, and shall respond appropriately to all warning and error messages from the tool.
AoU-PT-007	After ECO fixing by the physical implementation tool, user shall rerun PrimeTime timing analysis on the modified design and verify that all timing violations are fixed.
AoU-PT-008	User shall perform formal verification (using the Formality tool) of the ECO-modified netlist versus the original netlist in the physical implementation tool.
AoU-PT-009	After all incremental and reduced-resource ECO fixing cycles are complete, user shall run a final-signoff full-chip re-extraction using the StarRC tool and rerun a full-chip timing analysis.
AoU-PT-010	User shall examine the time stamps of the design data files from the physical implementation tool to verify that the design was not changed while the ECO was being generated by the PrimeTime tool.
AoU-PT-011	User shall review physical implementation tool logs to confirm relevant ECO changes have been applied and that the log file contains the expected execution steps.
AoU-PT-012	User shall perform a final signoff analysis that passes all constraints (`report_constraint`).

ID	Description
AoU-PT-013	User shall review block model and block context data directories/files listed in the log files to ensure that the correct data is being used for analysis.
AoU-PT-014	For final signoff analysis, user shall perform a full-chip flat analysis if memory and runtime resources allow it, or run multiple top-level analyses using flattened block data for selected blocks.
AoU-PT-015	User shall examine the time stamps of the block model and context files to verify that the correct files are being used.