ESP
Functional Safety Manual

March 2018, Revision 1.6

SYNOPSYS®

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.	18-Jan-2018
1.1	Added revision history, fixed template issues.	01-Feb-2018
1.2	Fixed boilerplate changes from general feedback, changed release month. Edited the entire document and updated the Appendix B	01-Mar-2018
1.3	Simplified Use Cases from 6 to 4; corrected some CoUs and most of the AoUs	04-Mar-2018
1.4	Update CoU and AoUs	09-Mar-2018
1.5	Incorporated exida review comments and updated Use cases	14-Mar-2018
1.6	Updated AoUs per exida feedback	15-Mar-2018

Contents

1
Customer Support

This section provides information about the customer support that you can avail 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 the terms used in this document.

Using This Document

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

The ESP tool is an equivalence checker intended to functionally verify custom circuit designs such as memories (SRAMs, Register Files, ROMs), custom digital circuit macros and standard cell libraries. The tool ensures that two design representations are functionally equivalent by using symbolic simulations to formally verify the following cases:

- Verilog view against a corresponding schematic SPICE netlist for a single design such as a custom SRAM.
- Verilog view against a corresponding schematic SPICE netlist for a family of designs or multiple design families such as an entire standard cell library.
- A reference Verilog view against a different or modified Verilog version of the same design.
- A reference SPICE netlist against a changed/modified or alternate SPICE netlist.

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 general information such as where to find the latest documentation and installation requirements regarding the use of the ESP 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 ESP 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 *ESP 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.
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 *ESP 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.
Defect	Product nonconformance.
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 (for example, generation of bad logic by a logic synthesis tool, failure of a static timing analysis tool to detect a timing violation).
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.
Software / software tool	ESP
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.
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.
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. 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.
TD	Tool error detection, as defined in ISO 26262-8, clause 11.
TI	Tool impact, as defined in ISO 26262-8, clause 11.

Term	**Definition**
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 - 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 Synopsys activities and end users of 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.

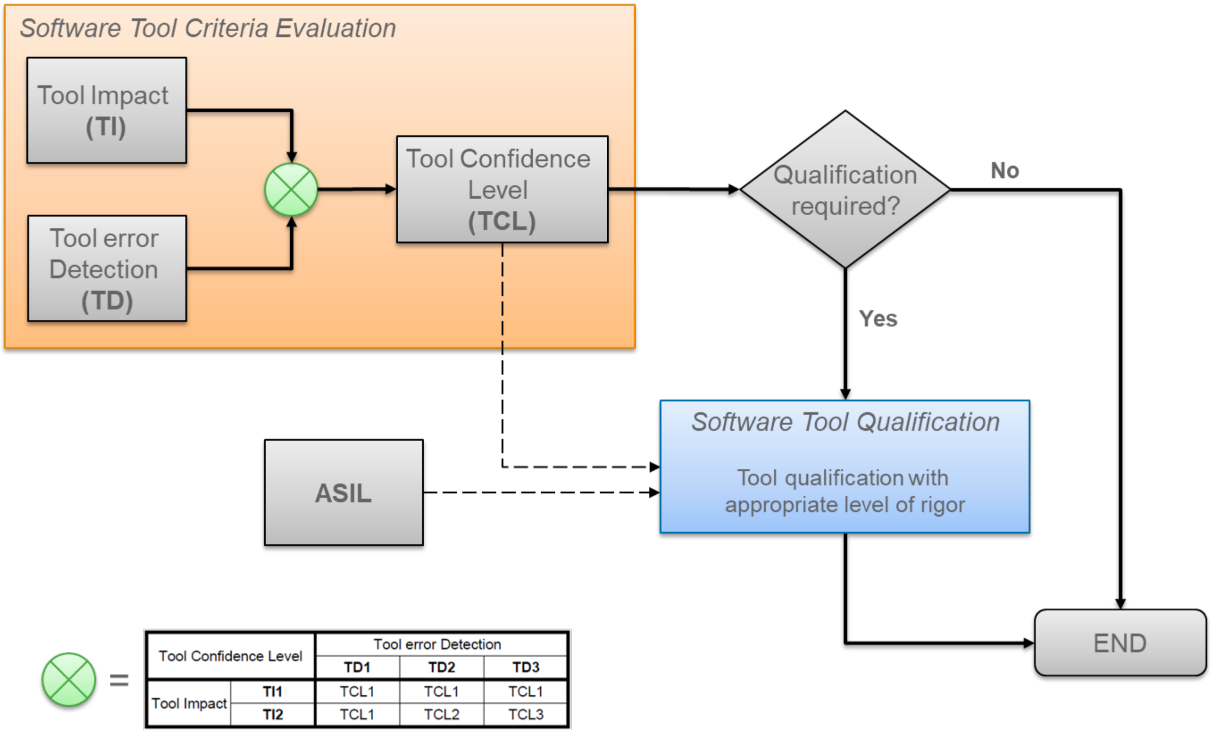
According to ISO 26262, to determine the required level of confidence in a software tool that is used in the development of a safety-related automotive product, the following criteria are evaluated:

- 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 of 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 are 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 of 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.

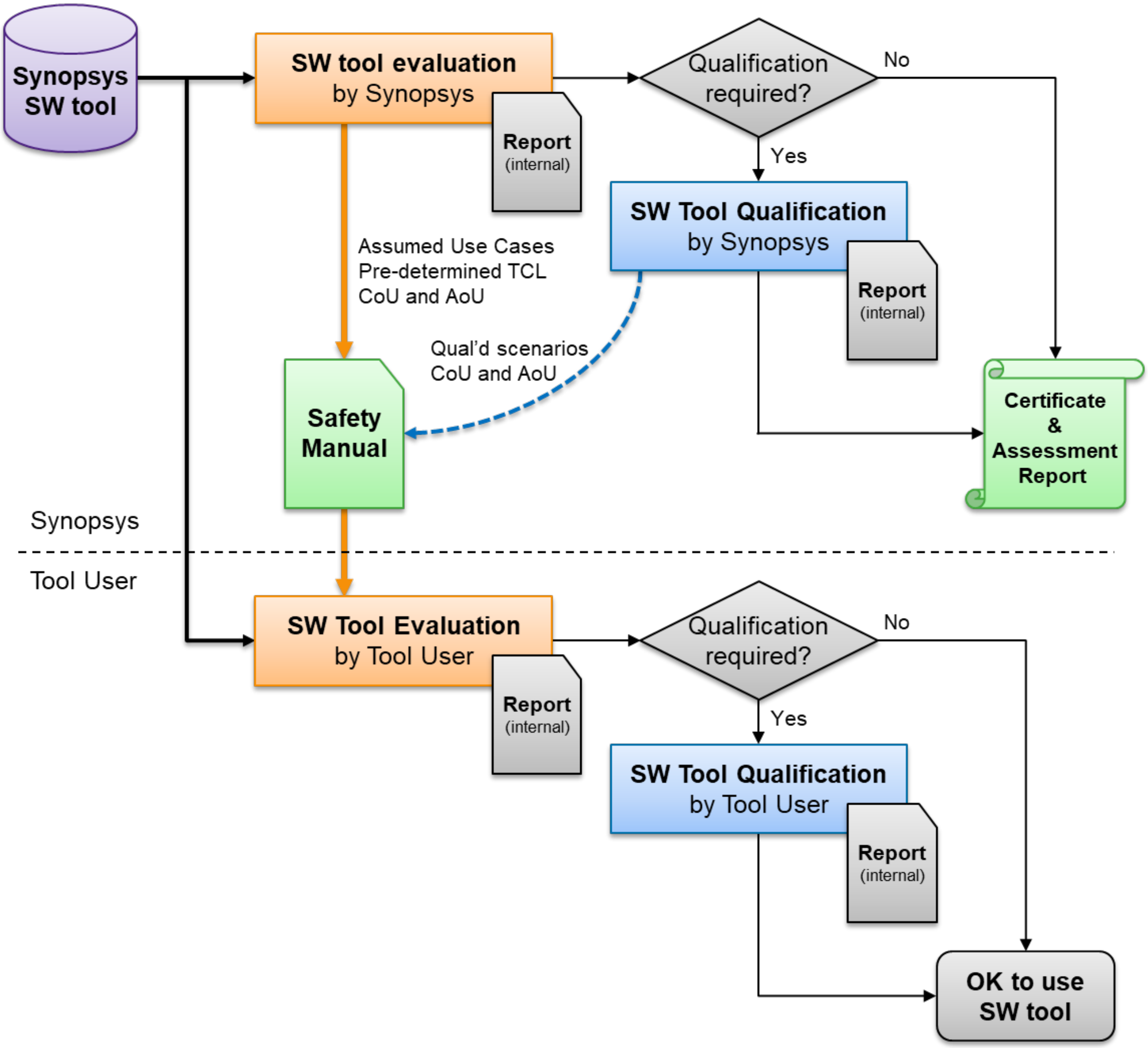


Figure 2: Work Split between Synopsys and Tool Users

Synopsys performs the following activities:

1. Software tool criteria evaluation
    - Identify possible **use cases** for the software tool, together with required **inputs** and expected **outputs**.
    - Specify the **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.
    - Analyse potential software tool **malfunctions**, and their effect on a safety-related product that is developed with this tool.
    - Analyse **prevention** and **detection measures** internal to the software tool, to avoid tool malfunctions, or to control and mitigate their effects.
    - Specify **assumptions of use (AoU)**, which are additional prevention and detection measures assumed to be performed by the end user of the tool.
    - Estimate 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).
    - Determine the required **Tool Confidence Level (TCL)** for each software tool malfunction, based on TI and TD.
    - Determine 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.
    - Summarize 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.
    - Summarize the qualification methods, procedures and results in a software tool qualification report.

3. Safety manual for the software tool
    - The *ESP 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 5, includes
        - Description of the 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 ESP 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 include
    - 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 might 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 might 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

To benefit from the work done by Synopsys, perform the following activities for each software tool according to Figure 2.

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 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 described verification steps are successful, the results of the Synopsys software tool criterial evaluation (and qualification) are applicable to the user, which means:
        - The required TCL pre-determined by Synopsys can be used 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 by Synopsys is sufficient, and the end user can use the tool without further software tool qualification

    - All the steps mentioned 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 as part of the software tool criteria evaluation fails (for example, different use case, CoU or AoU cannot be met, pre-determined TCL is not TCL1, and Synopsys has not performed a software tool qualification), you must perform your own software tool qualification.
    - The user plans and decides specific methods applied for tool qualification -- 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
ESP Description

This section provides a general description regarding the use of the ESP 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 ESP tool.

Coverage

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

Compliance with ISO 26262

The ESP 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 *ESP Functional Safety Manual*.

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

Product Documentation and Support

SolvNet provides comprehensive documentation for using the ESP tool. You can access the latest documentation for the ESP tool on SolvNet at the following address:

https://solvnet.synopsys.com/dow_retrieve/latest/dg/espolh/Default.htm.

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 *ESP Functional Safety Manual*.

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

Installation and Supported Platforms

To install the ESP tool, follow the guidelines in the *Synopsys® Installation Guide* as well as the specific *ESP Installation Notes* document.

You must download the tool executable and the INSTALL_README file 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 ESP 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 use the ESP tool, you must have the right understanding and working knowledge of the following:

- Electrical engineering and VLSI circuit design
- ISO 26262 standard
- Documentation of the ESP tool on SolvNet such as the ESP User Guide, (or by typing `esphelp` in the UNIX terminal command prompt)
- The Functional Safety Manual
- The published list of safety-related defects for the ESP tool available in the ESP Master List of Safety-Related Issues.
- Applicability of the ESP tool in the overall tool chain

Managing Known Safety-Related Defects

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

The *ESP Release Notes* references a list of all known safety-related defects for each release of the ESP tool in a SolvNet knowledge base article.

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

Managing New Releases

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

When installing a new version of the ESP tool, users must evaluate the impact of any known safety-related defects in their design by checking the accompanying *ESP 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 ESP tool

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

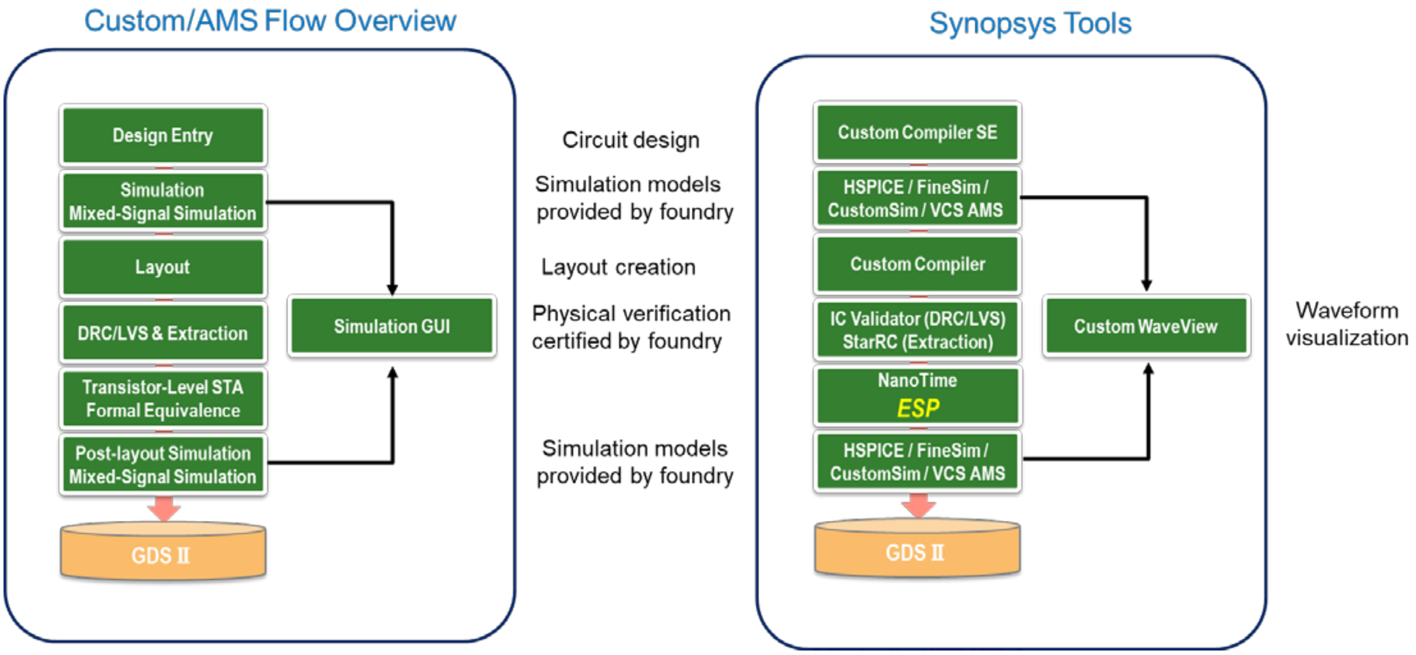
5
Synopsys Analog Tool Chain

This section provides an overview of where the ESP tool 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 ESP tool is applicable.

Synopsys Analog/Mixed-Signal Tool Chain



6
Use Cases

This section describes safety-qualified use cases of the ESP tool. You must perform TCL determination based on specific use cases.

The ESP equivalence checking tool is used for functional verification of custom designs such as memories (SRAMs, register files, ROMs), custom digital macros, and standard cell libraries. The tool serves as a complement to the user's circuit simulations for functionality (using HSPICE, CustomSim or equivalent) as it employs symbolic variables in place of binary input vectors in both the reference design (e.g. Verilog file) and the implementation design (SPICE netlist). Then it checks for matching symbolic values at both sets of outputs. In this way, a single ESP symbolic cycle can verify the same functionality as an exponentially larger number of circuit simulation binary input vector cycles. However, because ESP can only support a pre-layout schematic based SPICE netlist format, the equivalence check results should be cross-checked by post-layout extracted netlist circuit simulations (via CustomSim or equivalent) for completeness.

The ESP tool ensures that two design representations are functionally equivalent by using symbolic simulations to formally verify the following use cases:

- **Equivalence Check**
  - A reference Verilog view against a corresponding implementation schematic SPICE netlist for a single design such as a custom SRAM.
  - A reference Verilog view against a different or modified Verilog version for a single design.
  - A reference SPICE netlist against a changed/modified SPICE netlist for a single design.
- **Standard Cell Library Verification**
  - Any of the Equivalence Check modes as applied to a set of designs or multiple design families such as an entire standard cell library.

Once the basic Verilog versus schematic functional equivalence has been established, the ESP tool can be deployed in two auxiliary mode use cases:

- **Memory Redundancy Verification** mode to specifically test the row or column redundancy features, along with logical to physical mapping of the redundant rows or columns for SRAMs and other custom memories (not supported for Standard Cell Library Verification)
- **Power Integrity Verification** mode to detect short or open circuit sensitivities in multi-voltage domain circuits (not supported for Standard Cell Library Verification).

To see the tool overview in SolvNet, click on *ESP Online Help*.

Use Case 1: ESP Equivalence Check

In this use case, the main goal is to verify custom circuit logic functionality for large macros such as SRAMs, Register Files, ROMs and datapath circuits by comparing the Verilog (behavioral or structural) view against the schematic transistor-level circuit SPICE netlist view of the same design (V2S mode), two SPICE netlists for the same design (S2S mode), or two Verilog views of the same design (V2V mode).

In this use case, the ESP tool uses and generates the following main inputs and outputs:

- Inputs:
  - Verilog (behavioral or structural) file for listed in ASCII format
  - Verilog simulation libraries containing all the primitives and constructs needed by the ESP tool Verilog parser.
  - Schematic netlist for the design listed in SPICE netlist format.
  - SPICE device models for all components present in the netlist.
  - Tcl script containing the sequence of ESP commands needed to functionally verify the design and run functionality coverage analysis.

- Expected outputs:
  - Design log (log.tb) for detailed run status, design functionality correctness (`SUCCEEDED` or `FAILED)` and tool error or warning messages.
  - Coverage file (tb.cov) in binary format quantifying reference Verilog and implementation SPICE functionality coverage by the ESP testbench.
  - User defined summary report of design run `SUCCEEDED` or `FAILED` status, coverage, errors and warnings when the `report_log`, `report status` and `report_coverage` commands are invoked in the ESP Tcl script to post process the log.tb and tb.cov files.
  - Error vector files dump.vcd and esp.TestVector represent Verilog and SPICE netlist non-equivalence when the `debug_design` command is invoked in the ESP Tcl script.

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

- CoU-ESP-001: User shall not continue until all log files errors/warnings have been reviewed and corrective action is taken.
- CoU-ESP-002: User shall only use the Tcl script-based batch mode (and not the interactive command line mode) for the final ESP signoff run. The ESP Tcl scripts, user-generated summary files and tool logs shall be retained as design signoff records.
- CoU-ESP-003: User shall follow the commands and syntax specified in the latest released ESP User Guide and ESP Online Help to avoid potential obsolete syntax or tool settings.
- CoU-ESP-004: User shall not use ESP to functionally verify unsupported and non-standard circuit design styles, as described in Chapter 7 of the ESP Functional Safety Manual.
- CoU-ESP-005: User shall only use schematic based SPICE netlists for ESP. The ESP tool does not support layout-extracted SPICE netlists with back-annotated RC parasitics.
- CoU-ESP-006: User shall conduct a signoff review of all custom constraints applied to input pins, internal nets, device instances and voltage supplies as well as custom modifications of default tool parameters or internal algorithm settings.

This specific use case has been determined to be TCL1 as long as the following Assumptions-of-Use (AoU) measures for detection of erroneous outputs are followed:

- AoU-ESP-001: User shall review the cell log.tb and user summary log files for error messages, warnings, completeness (no missing sections), integrity (no corruption), logic functionality correctness and logic functionality coverage.
- AoU-ESP-003: User shall check that all output files are generated with an up-to-date timestamp.
- AoU-ESP-004: User shall ensure the run logs contain the status “SUCCEEDED” and the coverage report metrics meet signoff criteria.
  AoU-ESP-005: User shall cross check ESP functional verification results against the results from post-layout simulation with extracted netlists and binary input vectors (with a tool such as CustomSim).

All analyzed failure modes and prevention, detection and mitigation measures (including Assumptions-of-Use listed above) are independent of the exact ESP 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 *ESP* Use Case 1: ESP Equivalence Check

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

Use Case 2: ESP Standard Cell Library Verification Mode (SCLV)

In this use case, the goal is to verify an entire standard cell library while running the ESP tool in batch mode. The SCLV mode supports the main equivalence checks: Verilog versus Schematic (V2S), Schematic versus Schematic (S2S) and Verilog versus Verilog (V2V), though it does not support the Power Integrity Verification (PIV) and Memory Redundancy Verification (MRV) auxiliary modes. The ESP tool provides SCLV Tcl batch commands to help the user prepare the library cells for verification, run the cell verifications in batch mode and organize the results in composite summary files.

In this use case, the ESP tool uses and generates the following main inputs and outputs:

- Inputs:
  - Verilog (behavioral or structural) models for all library cells listed in ASCII format or inferred from Synopsys library .db binary format.
  - Verilog simulation libraries containing all the primitives and constructs needed by the ESP tool Verilog parser.
  - Schematic subcircuit netlists for all library cells listed in SPICE netlist format.
  - SPICE device models for all components present in the subcircuit netlists.
  - Tcl script containing the sequence of ESP commands needed to verify the library cells, run functionality coverage analysis for each cell and invoke the distributed processors to run as many ESP cell verification runs in parallel, contingent upon the available ESP licenses.

- Expected outputs:
  - Library cell log (log.tb) for detailed run status, design functionality correctness (`SUCCEEDED` or `FAILED)` and tool error or warning messages.
  - Coverage file (tb.cov) in binary format quantifying reference Verilog and implementation SPICE functionality coverage by the ESP testbench.
  - Composite summary report of library cell run status, coverage, errors and warnings when the `report_log`, `report status` and `report_coverage` commands are invoked in the ESP Tcl script.
  - Error vector files dump.vcd and esp.TestVector represent Verilog and SPICE netlist non-equivalence when the `debug_design` command is invoked in the ESP Tcl script.

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

- CoU-ESP-001: User shall not continue until all log files errors/warnings have been reviewed and corrective action is taken.
- CoU-ESP-002: User shall only use the Tcl script-based batch mode (and not the interactive command line mode) for the final ESP signoff run. The ESP Tcl scripts, user-generated summary files and tool logs shall be retained as design signoff records.
- CoU-ESP-003: User shall follow the commands and syntax specified in the latest released ESP User Guide and ESP Online Help to avoid potential obsolete syntax or tool settings.
- CoU-ESP-004: User shall not use ESP to functionally verify unsupported and non-standard circuit design styles, as described in Chapter 7 of the ESP Functional Safety Manual.
- CoU-ESP-005: User shall only use schematic based SPICE netlists for ESP. The ESP tool does not support layout-extracted SPICE netlists with back-annotated RC parasitics.
- CoU-ESP-006: User shall conduct a signoff review of all custom constraints applied to input pins, internal nets, device instances and voltage supplies as well as custom modifications of default tool parameters or internal algorithm settings.

This specific use case has been determined to be TCL1 as long as the following Assumptions-of-Use (AoU) measures for detection of erroneous outputs are followed:

- AoU-ESP-001: User shall review the cell log.tb and user summary log files for error messages, warnings, completeness (no missing sections), integrity (no corruption), logic functionality correctness and logic functionality coverage.
- AoU-ESP-002: User shall verify whether all the library cells are analyzed.
- AoU-ESP-003: User shall check that all output files are generated with an up-to-date timestamp.
- AoU-ESP-004: User shall ensure the run logs contain the status `“SUCCEEDED”` and the coverage report metrics meet signoff criteria.
- AoU-ESP-005: User shall cross check ESP functional verification results against the results from post-layout simulation with extracted netlists and binary input vectors (with a tool such as CustomSim).

All analyzed failure modes and prevention, detection and mitigation measures (including Assumptions-of-Use listed above) are independent of the exact ESP 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 *ESP* Use Case 2: Standard Cell Library Verification Mode

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

Use Case 3: ESP Memory Redundancy Verification

In this use case, the main goal is to specifically verify the row and column redundancy features within SRAMs and generate the logical to physical bitmap for the redundancy addresses. This mode should be run after ESP Verilog versus schematic has already verified basic functionality (with the redundancy features deactivated in the ESP testbench).

In this use case, the ESP tool uses and generates the following main inputs and outputs:

- Inputs:
  - Reference Verilog (behavioral or structural) file for the design listed in ASCII format
  - Verilog simulation libraries containing all the primitives and constructs needed by the ESP tool Verilog parser
  - Implementation schematic netlist for the design listed in SPICE netlist format.
  - SPICE device models for all components present in the netlist
  - Tcl script containing the sequence of ESP commands needed to functionally verify the design and run functionality coverage analysis

- Expected outputs:
  - Design log (log.tb) for detailed run status, design functionality correctness (`SUCCEEDED` or `FAILED)` and tool error or warning messages.
  - User defined summary report of design run `SUCCEEDED` or `FAILED` status, coverage, errors and warnings when the `report_log`, `report status` and `report_coverage` commands are invoked in the ESP Tcl script to post process the log.tb and tb.cov files.
  - Error vector files dump.vcd and esp.TestVector represent memory redundancy logic errors when the `debug_design` command is invoked in the ESP Tcl script.

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

- CoU-ESP-001: User shall not continue until all log files errors/warnings have been reviewed and corrective action is taken.
- CoU-ESP-002: User shall only use the Tcl script-based batch mode (and not the interactive command line mode) for the final ESP signoff run. The ESP Tcl scripts, user-generated summary files and tool logs shall be retained as design signoff records.
- CoU-ESP-003: User shall follow the commands and syntax specified in the latest released ESP User Guide and ESP Online Help to avoid potential obsolete syntax or tool settings.

- CoU-ESP-004: User shall not use ESP to functionally verify unsupported and non-standard circuit design styles, as described in Chapter 7 of the ESP Functional Safety Manual.
- CoU-ESP-005: User shall only use schematic based SPICE netlists for ESP. The ESP tool does not support layout-extracted SPICE netlists with back-annotated RC parasitics.
- CoU-ESP-006: User shall conduct a signoff review of all custom constraints applied to input pins, internal nets, device instances and voltage supplies as well as custom modifications of default tool parameters or internal algorithm settings.
- CoU-ESP-007: User shall neither run Memory Redundancy Verification nor Power Integrity Verification auxiliary modes until basic functional equivalence has been verified with Verilog versus Schematic mode.

This specific use case has been determined to be TCL1 as long as the following Assumptions-of-Use (AoU) measures for detection of erroneous outputs are followed:

- AoU-ESP-001: User shall review the cell log.tb and user summary log files for error messages, warnings, completeness (no missing sections), integrity (no corruption), logic functionality correctness and logic functionality coverage
- AoU-ESP-003: User shall check that all output files are generated with an up-to-date timestamp.
- AoU-ESP-004: User shall ensure the run logs contain the status “SUCCEEDED” and the coverage report metrics meet signoff criteria.
- AoU-ESP-005: User shall cross check ESP functional verification results against the results from post-layout simulation with extracted netlists and binary input vectors (with a tool such as CustomSim).

All analyzed failure modes and prevention, detection and mitigation measures (including Assumptions-of-Use listed above) are independent of the exact ESP 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 *ESP* Use Case 3: ESP Memory Redundancy Verification Mode

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

Use Case 4: ESP Power Integrity Verification

In this use case, the main goal is to check for inter-power (voltage) domain circuit connectivity violations such as incorrect voltage supply connections and missing level shifters. This use case requires the same inputs as the Verilog versus Schematic mode. Power Integrity Verification (PIV) is only supported in conjunction with the ESP Verilog versus Schematic (V2S) equivalence check mode. It is not supported for Verilog versus Verilog (V2V), Schematic versus Schematic (S2S), Memory Redundancy Verification (MRV) and Standard Cell Library Verification (SCLV).

Note: Power integrity verification mode is not supported for library cell verification in ESP N-2017.12 or prior versions.

In this use case, the ESP tool uses and generates the following main inputs and outputs:

- Inputs:
    - Reference Verilog (behavioral or structural) file containing the design module
    - Verilog simulation libraries containing all the primitives and constructs needed by the ESP tool Verilog parser.
    - Implementation schematic SPICE netlist containing the subcircuit module for the same design
    - SPICE device models for all components present in the subcircuit netlists.
    - Tcl script containing the ESP commands and constraints needed for Power Integrity Verification
- Expected outputs:
    - Design log (log.tb) for detailed run status, design functionality correctness (`SUCCEEDED` or `FAILED)` and tool error or warning messages.
    - Coverage file (tb.cov) in binary format quantifying reference Verilog and implementation SPICE functionality coverage by the ESP testbench.
    - User-defined verification summary file containing post-process Tcl query results for `report_log`, `report status` and `report_coverage` commands to summarize pass/fail status as well as tool error and warning messages.
    - Error vector files dump.vcd and esp.TestVector represent Verilog and SPICE netlist non-equivalence when the `debug_design` command is invoked in the ESP Tcl script.
    - User specified summary containing post-process query results for power integrity verification violations when `report_inspector_results` command is invoked in the Tcl script.
    - Power integrity rule violation error vector files with the filename extension.tv

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

- CoU-ESP-001: User shall not continue until all log files errors/warnings have been reviewed and corrective action is taken.
- CoU-ESP-002: User shall only use the Tcl script-based batch mode (and not the interactive command line mode) for the final ESP signoff run. The ESP Tcl scripts, user-generated summary files and tool logs shall be retained as design signoff records.
- CoU-ESP-003: User shall follow the commands and syntax specified in the latest released ESP User Guide and ESP Online Help to avoid potential obsolete syntax or tool settings.
- CoU-ESP-004: User shall not use ESP to functionally verify unsupported and non-standard circuit design styles, as described in Chapter 7 of the ESP Functional Safety Manual.
- CoU-ESP-005: User shall only use schematic based SPICE netlists for ESP. The ESP tool does not support layout-extracted SPICE netlists with back-annotated RC parasitics.
- CoU-ESP-006: User shall conduct a signoff review of all custom constraints applied to input pins, internal nets, device instances and voltage supplies as well as custom modifications of default tool parameters or internal algorithm settings.
- CoU-ESP-007: User shall neither run Memory Redundancy Verification nor Power Integrity Verification auxiliary modes until basic functional equivalence has been verified with Verilog versus Schematic mode.

This specific use case has been determined to be TCL1 as long as the following Assumptions-of-Use (AoU) measures for detection of erroneous outputs are followed:

- AoU-ESP-001: User shall review the cell log.tb and user summary log files for error messages, warnings, completeness (no missing sections), integrity (no corruption), logic functionality correctness and logic functionality coverage.
- AoU-ESP-003: User shall check that all output files are generated with an up-to-date timestamp.
- AoU-ESP-004: User shall ensure the run logs contain the status `“SUCCEEDED”` and the coverage report metrics meet signoff criteria.
- AoU-ESP-006: User shall cross check ESP power integrity verification results against results from another post-layout netlist electrical rule checker (such as CustomSim Circuit Check).

All analyzed failure modes and prevention, detection and mitigation measures (including conditions and assumptions of use listed above) are independent of the exact ESP 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 *ESP* Use Case 4: ESP Power Integrity Verification

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 ESP tool are listed in the ESP Master List of Safety-Related Issues available on SolvNet.

The following types of custom circuits should not use the ESP tool for functional verification signoff:

- Circuits that require continuous or intermediate voltage values, such as voltage dividers, voltage converter, voltage references, voltage regulators and other forms of bias circuits are not accurately modeled with ESP since the tool tries to resolve all intermediate voltages to simplified logic values 0, 1, X (unknown) or Z (tristate).
- Analog current based circuits such as current mirrors, operational amplifiers and analog multipliers or dividers cannot accurately be modeled with the ESP tool since it uses a simplified RC switch model in place of each transistor in the netlist for performance and capacity reasons.
- Analog frequency or time domain circuits such as phase locked loops, voltage controlled oscillators, analog-to-digital or digital-to-analog converters, modulators, demodulators or filters are also out of scope for ESP analysis for the reasons listed previously and because the ESP tool does not support inductor elements in the netlist.
- FLASH memories or other similar circuits that dynamically modify transistor threshold voltage or other device parameters cannot be verified with ESP.

If the exception circuits are part of a larger design that can be verified with the ESP tool, users must substitute these exception circuits with equivalent Verilog models to facilitate larger design verification while independently verifying the exception circuits with SPICE for design sign-off.

Appendix A
Software Tool Information

This section provides general information about the ESP software tool, required by the tool user for performing software tool criteria evaluation.

The following information about ESP 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	ESP	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.

Tool environment (OS)	Refer to the ESP Installation Notes at https://solvnet.synopsys.com/DownloadCenter. Click the ESP tool name, 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 the 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 ESP
User manual and other usage guide documents	To access click the ESP User Guide and ESP Online Help on SolvNet or type `esphelp` on the UNIX command prompt.	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 the ESP Master List of Safety-Related Issues in Section 7 of this document. As of ESP version N-2017.12, there are no outstanding safety-related bugs.	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 ESP 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-ESP-001	User shall not continue until all log files errors/warnings have been reviewed and corrective action is taken.
CoU-ESP-002	User shall only use the Tcl script-based batch mode (and not the interactive command line mode) for the final ESP signoff run. The ESP Tcl scripts, user-generated summary files and tool logs shall be retained as design signoff records.
CoU-ESP-003	User shall follow the commands and syntax specified in the latest released ESP User Guide and ESP Online Help to avoid potential obsolete syntax or tool settings.
CoU-ESP-004	User shall not use ESP to functionally verify unsupported and non-standard circuit design styles, as described in Chapter 7 of the ESP Functional Safety Manual.
CoU-ESP-005	User shall only use schematic based SPICE netlists for ESP. The ESP tool does not support layout-extracted SPICE netlists with back-annotated RC parasitics.
CoU-ESP-006	User shall conduct a signoff review of all custom constraints applied to input pins, internal nets, device instances and voltage supplies as well as custom modifications of default tool parameters or internal algorithm settings.
CoU-ESP-007	User shall neither run Memory Redundancy Verification nor Power Integrity Verification auxiliary modes until basic functional equivalence has been verified with Verilog versus Schematic mode.

The complete list of Assumption of Use (AoU) ESP 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-ESP-001	User shall review the cell log.tb and user summary log files for error messages, warnings, completeness (no missing sections), integrity (no corruption), logic functionality correctness and logic functionality coverage.
AoU-ESP-002	User shall verify whether all the library cells are analyzed.
AoU-ESP-003	User shall check that all output files are generated with an up-to-date timestamp.
AoU-ESP-004	User shall ensure the run logs contain the status `“SUCCEEDED”` and the coverage report metrics meet signoff criteria.
AoU-ESP-005	User shall cross check ESP functional verification results against the results from post-layout simulation with extracted netlists and binary input vectors (with a tool such as CustomSim).
AoU-ESP-006	User shall cross check ESP power integrity verification results against results from another post-layout netlist electrical rule checker (such as CustomSim Circuit Check).