StarRC™
Functional Safety Manual

July 2021, Revision 2.0

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
0.9	Initial draft	05-Oct-2017
1.0	First release of the document submitted for review.	22-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.	01-Mar-2018
1.3	Added AoU-SRC-006 and AoU-SRC-007 based on certification review	09-Mar-2018
2.0	Updated for ISO 26262 re-certification	12-Jul-2021

Contents

1
Customer Support

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

Accessing SolvNetPlus

The SolvNetPlus 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 SolvNetPlus site:

1. Go to the web page at https://solvnetplus.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 SolvNetPlus 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 *StarRC Functional Safety Manual* describes the proper use of the StarRC tool in safety-related applications according to the ISO 26262 standard, and is intended to confirm the compliance of the StarRC tool to the standard when used in the context of a tool chain.

The StarRC tool extracts parasitics such as resistors, capacitors, and inductors from databases that represent integrated circuit layout designs. You can use the StarRC tool to generate parasitic netlists for many types of analysis, such as timing, noise, and electromigration.

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 StarRC 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 StarRC 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 *StarRC 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.

Back-annotation	The procedure of attaching extracted parasitic devices to the original design schematic, which effectively creates a revised design that more accurately represents the circuit behavior after manufacturing. Also referred to as annotation.
Capacitance	A parameter that represents the ability of a structure to store an electric charge. Parasitic capacitance is capacitance that is not part of the original design, but instead is introduced from the physical layout.
Command file	A file that contains StarRC commands to specify conditions for the parasitic extraction operation. Also referred to as a cmd file.
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 *StarRC 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.
DCLS	Dual Core Lock Step A safety mechanism that can be implemented in functional safety that have the processors operating in parallel (e.g. in lock step). It is used for error detection of the cores of a single event upset (such as a soft error).
Defect	Product nonconformance.
DMR	Dual Modular Redundancy This is a safety mechanism that can be implemented for functional safety. In most cases, it is used in the context of safety registers for error detection of a single event upset (such as a soft error).
DRC	Design Rule Checking. An analysis that verifies whether a circuit layout follows all the rules necessary to make the design manufacturable by the selected process. An example of a design rule is the minimum allowable spacing between adjacent metal lines.
DSPF	Detailed Standard Parasitic Format. A format used to store circuit parasitic information for back-annotation to the design. This format contains detailed parasitics, which includes all extracted parasitics before reduction (simplification) operations.
EM	Electromigration.

	An analysis of current density and other electrical performance factors that affect integrated circuit reliability.
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).
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.
FFSM	Failsafe Finite State Machine State machine encoding that is used as a safety mechanism.
Field solver	An analysis mode of the StarRC tool that directly analyzes the parasitics of a structure in the design layout by using three-dimensional electrostatic field analysis. This is the highest accuracy analysis mode, but takes more analysis time than the standard mode of analyis, which matches layout structures to structures in the foundry-supplied process modeling file.
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.
Foundry	A company that manufactures integrated circuits on a contract basis. A foundry provides guidelines and source files that enable circuit designers to design circuits that can be manufactured successfully using the foundry's process. Source files include command files and process modeling files.
FuSa	Functional Safety
Inductance	A parameter that represents the tendency of electrical behavior of a circuit to be affected by neighboring circuits. Parasitic inductance is inductance that is not part of the original design, but instead is introduced from the physical layout.
LEF/DEF	A circuit design database format.
LVS	Layout Versus Schematic.

	An analysis that provides the mapping between the design schematic (which uses high-level device representations) and the design layout (which shows the physical layout of those devices).
Milkyway	A circuit design database format.
NDM	A circuit design database format.
OASIS	A circuit design database format.
Process	The specific patterning and chemical steps used to manufacture an integrated circuit.
Process modeling file	A binary file provided by a foundry that contains information about how physical structures in the circuit layout result in parasitic devices associated with those layout structures after manufacturing. This file is unique to a specific foundry manufacturing process.
Resistance	A parameter that represents the opposition to the flow of electrical current in a material. Parasitic resistance is resistance that is not part of the original design, but instead is introduced from the physical layout.
Signoff	The final cycle of testing a design before submitting it for manufacturing. Signoff includes extracting parasitics from the final layout and performing timing analysis on the final design with its annotated parasitics.
Software / software tool	The StarRC 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.
SolvNetPlus	Synopsys customer support site.
SPEF	Standard Parasitic Exchange Format. A format used to store circuit parasitic information for back-annotation to the design. This format is defined in IEEE standard 1481-2009.
SRC	StarRC
SSF	Safety Specification Format
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 internal defect database. Only Synopsys employees can access the internal defect database. However, limited STAR information is available from SolvNetPlus 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.
StarRC	The Synopsys parasitic extraction tool.
STA	Static Timing Analysis. An analysis that validates the timing performance of a design by checking all paths for timing violations.
SVF	Synopsys Verification File This file is used as guidance for the Formality product.
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.
TD	Tool error detection, as defined in ISO 26262-8, clause 11.
TI	Tool impact, as defined in ISO 26262-8, clause 11.
TMR	Triple Modular Redundancy This is a safety mechanism that can be implemented for functional safety. In most cases, it is used in the context of safety registers for error correction of a single event upset (such as a soft error).
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.

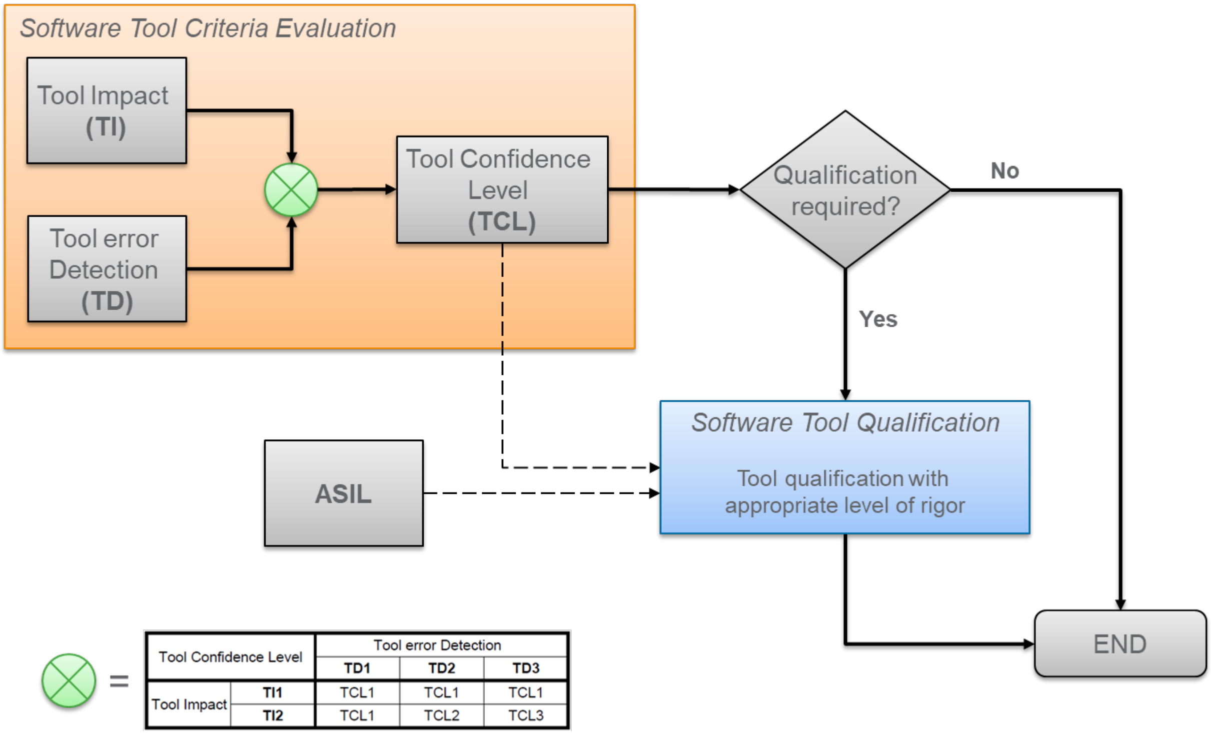
According to ISO 26262, to determine the required level of confidence in a software tool that is used in the development of a safety-related automotive product, the following criteria are evaluated:

- 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.

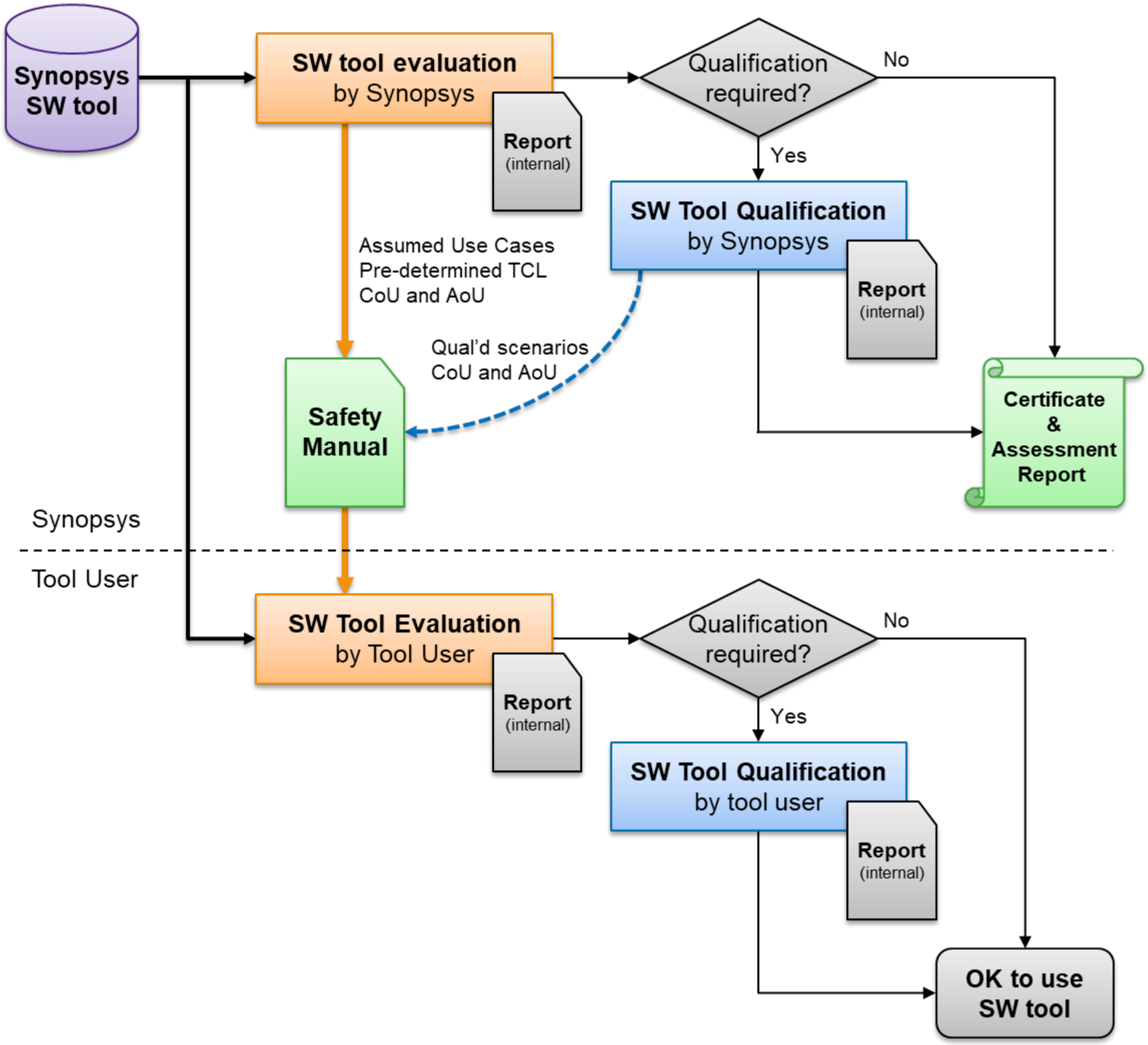


Figure 2: Work Split Between Synopsys and Tool Users

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 *StarRC 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 StarRC 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 StarRC Description

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

Coverage

The *StarRC Functional Safety Manual* is intended to be used starting with the version N-2017.12 and later versions of the StarRC 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 StarRC 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 *StarRC Functional Safety Manual*.

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

Product Documentation and Support

Comprehensive documentation for using the StarRC tool is provided on SolvNetPlus. The latest documentation for the StarRC tool can be accessed at StarRC Online Help on SolvNetPlus.

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 *StarRC Functional Safety Manual*.

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

Installation and Supported Platforms

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

Users are required to download the tool executable and INSTALL_README from the SolvNetPlus 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 StarRC 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 StarRC 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 StarRC tool, such as the User Guide, at StarRC Online Help on SolvNetPlus.

- This Functional Safety Manual
- The published list of safety-related defects for the StarRC tool available at StarRC Safety-Related Issues Master List
- Applicability of the StarRC tool in the overall tool chain

Managing Known Safety-Related Defects

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

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

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

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

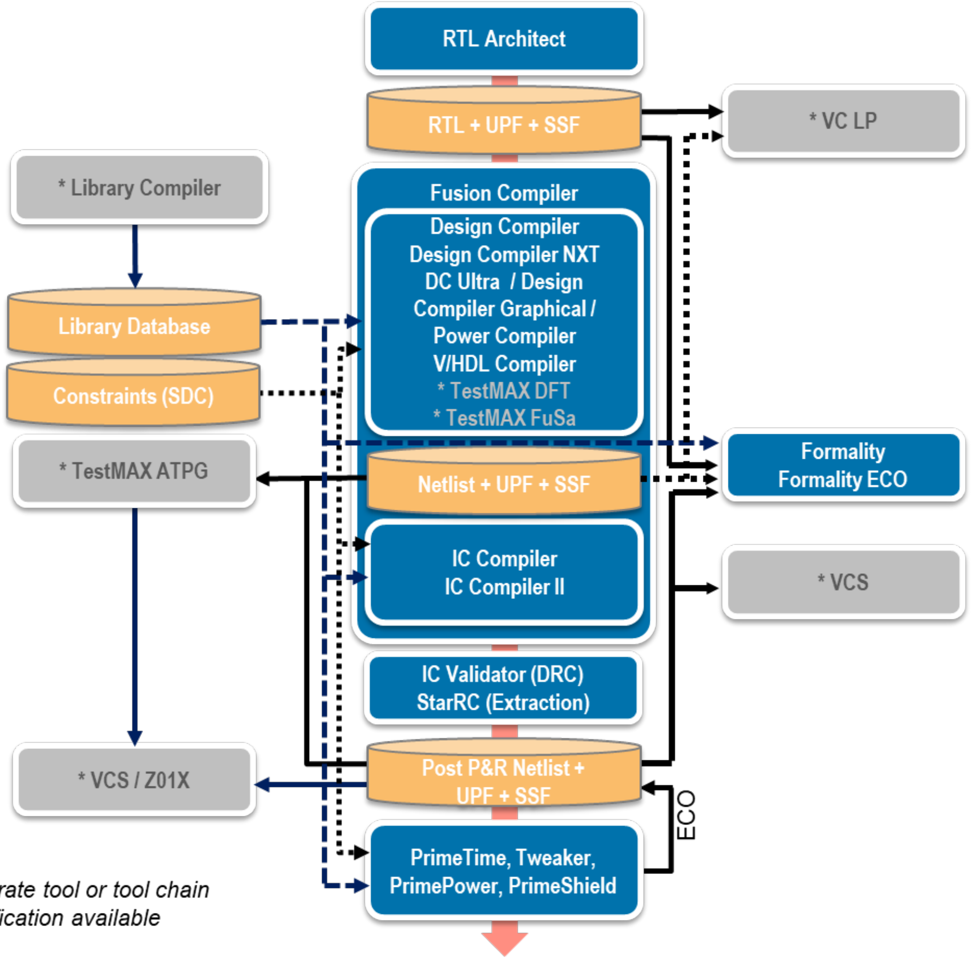
In addition, users must refer to the latest version of this document, the *StarRC Functional Safety Manual*, available with the product release contents.

5
Synopsys Digital and Analog Tool Chain

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



* *Separate tool or tool chain certification available*

Figure 3: Synopsys Digital Tool Chain

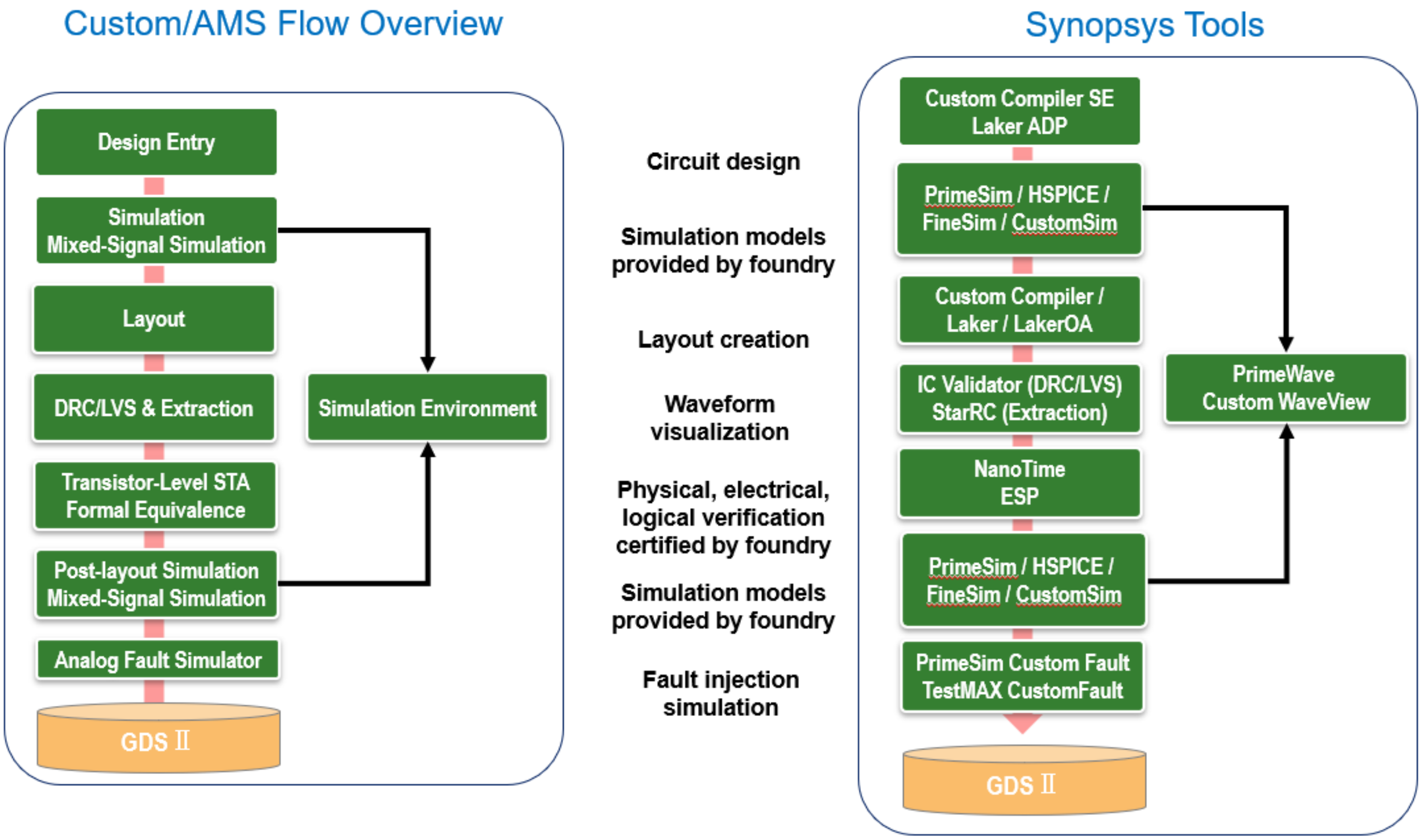

Custom/AMS Flow Overview
Synopsys Tools
Design Entry
Simulation Mixed-Signal Simulation
Layout
DRC/LVS & Extraction
Simulation Environment
Transistor-Level STA Formal Equivalence
Post-layout Simulation Mixed-Signal Simulation
Analog Fault Simulator
GDS II
Circuit design
Simulation models provided by foundry
Layout creation
Waveform visualization
Physical, electrical, logical verification certified by foundry
Simulation models provided by foundry
Fault injection simulation
Custom Compiler SE Laker ADP
PrimeSim / HSPICE / FineSim / CustomSim
Custom Compiler / Laker / LakerOA
IC Validator (DRC/LVS) StarRC (Extraction)
NanoTime ESP
PrimeSim / HSPICE / FineSim / CustomSim
PrimeSim Custom Fault TestMAX CustomFault
PrimeWave Custom WaveView
GDS II

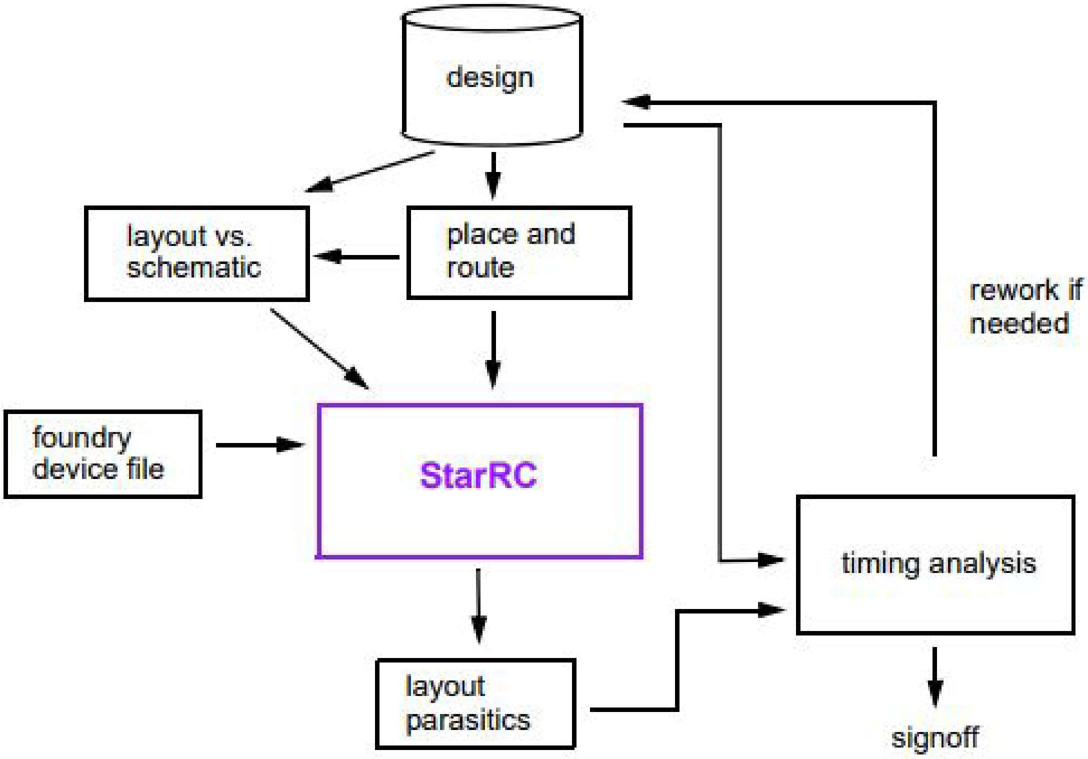
Figure 4: Synopsys Analog/Mixed-Signal Tool Chain

6
Use Cases

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

The StarRC tool extracts parasitics such as resistors, capacitors, and inductors from databases that represent integrated circuit layout designs. You can use the StarRC tool to generate netlists for many types of analysis, such as timing, noise, and electromigration.

The StarRC tool can be used for both digital and analog design flows. The following diagram shows on a high level how the StarRC tool fits in the design flow.



Use Case 1: Parasitic RC extraction

In this use case, the goal is to extract the RC (resistance and capacitance) parasitics of the design. The RC parasitics are then stored in an output netlist which is used by other software tools such as circuit simulation, STA (static timing analysis), and EM (electromigration) analysis tools. This use case represents the most common extraction procedure for circuit design and is the procedure used for signoff.

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

- Inputs:
  - Design database
    - Digital flow: NDM, LEF/DEF, or Milkyway format design database
    - Analog flow: Files generated by an LVS (layout versus schematic) tool
  - Foundry process modeling files
  - StarRC command file (cmd file)
- Expected outputs:
  - Parasitic netlist (SPEF or DSPF format file)
  - RC view for use by the Custom Compiler tool
  - Reports
  - Log files

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

- CoU-SRC-001: User shall review all error and warning messages and take appropriate action.
- CoU-SRC-002: User shall follow the StarRC Reference Methodology or use equivalent scripts to generate required command files to run the tool. If the command file is provided by the foundry, then the user shall use that file.
- CoU-SRC-003: For the final run, command file based batch mode execution shall be used, without interactive command line entry or GUI manual command entry. Command files and log files shall be retained as design signoff records.
- CoU-SRC-004: User shall provide an input layout that is DRC and LVS clean.
- CoU-SRC-005: User shall always use the foundry qualified process modeling file.

For this use case of *StarRC*, 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-SRC-001: User shall verify back-annotation results in downstream tools including timing analysis tools (such as the PrimeTime or NanoTime tool) and SPICE simulation tools (such as PrimeSim HSPICE / HSPICE, PrimeSim SPICE / FineSim, or PrimeSim XA / CustomSim tools).
- AoU-SRC-002: User shall review the results files for completeness and shall review the log files for error, warning, and information messages.
- AoU-SRC-003: User shall check that all output files are generated with an up-to-date timestamp.

- AoU-SRC-004: For digital or gate level flows, user shall compare extraction values from the StarRC tool to in-design extraction values in a physical implementation tool (such as the IC Compiler II or Fusion Compiler tool). If a significant discrepancy is found, based on user judgment, user shall run StarRC field solver (FS) extraction on those nets.
- AoU-SRC-005: For analog or transistor level flows, user shall review extraction results against expected results, then based on user judgment, run the field solver (FS) for verification of nets of concern.
- AoU-SRC-006: User shall check the netlist file for R, L, C components to ensure expected extraction mode (R, C, RLC, RC) is performed or not.
- AoU-SRC-007: User shall turn on special options in StarRC (NETLIST_TAIL_COMMENTS: YES) and check netlist file to ensure all expected layers are extracted.

All analyzed failure modes and prevention, detection and mitigation measures (including conditions and assumptions of use listed above) are independent of the exact StarRC 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 *StarRC* Use Case 1: Parasitic RC extraction

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

Use Case 2: Parasitic RLC extraction

In this use case, the goal is to extract the RLC (resistance, inductance, and capacitance) parasitics of the design. The RLC are then stored in an output netlist which is used by other software tools such as circuit simulation, STA (static timing analysis), and EM (electromigration) analysis tools. This use case represents extraction performed for analysis of clock networks. For final signoff analysis of a complete design, standard RC extraction is performed (*Use Case 1: Parasitic RC extraction*

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

- Inputs:
  - Design database
    - Digital flow: NDM, LEF/DEF, or Milkyway format design database
    - Analog flow: Files generated by an LVS (layout versus schematic) tool

    - Foundry process modeling files
    - StarRC command file (cmd file)
- Expected outputs:
    - Parasitic netlist (DSPF format file)
    - Reports
    - Log files

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

- CoU-SRC-001: User shall review all error and warning messages and take appropriate action.
- CoU-SRC-002: User shall follow the StarRC Reference Methodology or use equivalent scripts to generate required command files to run the tool. If the command file is provided by the foundry, then the user shall use that file.
- CoU-SRC-003: For the final run, command file based batch mode execution shall be used, without interactive command line entry or GUI manual command entry. Command files and log files shall be retained as design signoff records.
- CoU-SRC-004: User shall provide an input layout that is DRC and LVS clean.
- CoU-SRC-005: User shall always use the foundry qualified process modeling file.

For this use case of *StarRC*, 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-SRC-001: User shall verify back-annotation results in downstream tools including timing analysis tools (such as the PrimeTime or NanoTime tool) and SPICE simulation tools (such as PrimeSim HSPICE / HSPICE, PrimeSim SPICE / FineSim, or PrimeSim XA / CustomSim tools).
- AoU-SRC-002: User shall review the results files for completeness and shall review the log files for error, warning, and information messages.
- AoU-SRC-003: User shall check that all output files are generated with an up-to-date timestamp.
- AoU-SRC-004: For digital or gate level flows, user shall compare extraction values from the StarRC tool to in-design extraction values in a physical implementation tool (such as the IC Compiler II or Fusion Compiler tool). If a significant discrepancy is found, based on user judgment, user shall run StarRC field solver (FS) extraction on those nets.
- AoU-SRC-005: For analog or transistor level flows, user shall review extraction results against expected results, then based on user judgment, run the field solver (FS) for verification of nets of concern.

- AoU-SRC-006: User shall check the netlist file for R, L, C components to ensure expected extraction mode (R, C, RLC, RC) is performed or not.
- AoU-SRC-007: User shall turn on special options in StarRC (NETLIST_TAIL_COMMENTS: YES) and check netlist file to ensure all expected layers are extracted.

All analyzed failure modes and prevention, detection and mitigation measures (including conditions and assumptions of use listed above) are independent of the exact StarRC 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 *StarRC* Use Case 2: Parasitic RLC extraction

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 StarRC tool are listed in the StarRC Safety-Related Issues Master List available on SolvNetPlus.

LIM-1: LCA Features

Each release of the StarRC 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 StarRC software tool, which is needed by the tool user for performing his/her software tool criteria evaluation.

The following information about StarRC 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	StarRC N-2017.12 and later versions	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 StarRC Installation Notes at https://www.synopsys.com/install. Click the StarRC 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 StarRC
User manual and other usage guide documents	See StarRC Online Help and Reference Methodology on SolvNetPlus.	ISO 26262-8, 11.4.4.2.a – d Tool user to include a link to these documents (Synopsys SolvNetPlus 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. For the published list of safety-related defects for the StarRC tool, see StarRC Safety-Related Issues Master List on SolvNetPlus	ISO 26262-8, 11.4.4.2.e Tool user to include a link to these documents (Synopsys SolvNetPlus 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 StarRC is shown in the table below. CoU define 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-SRC-001	User shall review all error and warning messages and take appropriate action.
CoU-SRC-002	User shall follow the StarRC Reference Methodology or use equivalent scripts to generate required command files to run the tool. If the command file is provided by the foundry, then the user shall use that file.
CoU-SRC-003	For the final run, command file based batch mode execution shall be used, without interactive command line entry or GUI manual command entry. Command files and log files shall be retained as design signoff records.
CoU-SRC-004	User shall provide an input layout that is DRC and LVS clean.
CoU-SRC-005	User shall always use the foundry qualified process modeling file.

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

ID	Description
AoU-SRC-001	User shall verify back-annotation results in downstream tools including timing analysis tools (such as the PrimeTime or NanoTime tool) and SPICE simulation tools (such as PrimeSim HSPICE / HSPICE, PrimeSim SPICE / FineSim, or PrimeSim XA / CustomSim tools).
AoU-SRC-002	User shall review the results files for completeness and shall review the log files for error, warning, and information messages.
AoU-SRC-003	User shall check that all output files are generated with an up-to-date timestamp.
AoU-SRC-004	For digital or gate level flows, user shall compare extraction values from the StarRC tool to in-design extraction values in a physical implementation tool (such as the IC Compiler II or Fusion Compiler tool). If a significant discrepancy is found, based on user judgment, user shall run StarRC field solver (FS) extraction on those nets.

ID	Description
AoU-SRC-005	For analog or transistor level flows, user shall review extraction results against expected results, then based on user judgment, run the field solver (FS) for verification of nets of concern.
AoU-SRC-006	User shall check the netlist file for R, L, C components to ensure expected extraction mode (R, C, RLC, RC) is performed or not.
AoU-SRC-007	User shall turn on special options in StarRC (NETLIST_TAIL_COMMENTS: YES) and check netlist file to ensure all expected layers are extracted.