

Understanding How to Assess Tool Confidence Levels for ISO 26262

October 2017

Overview

ISO 26262, the automotive functional safety standard, requires the assessment of software tool confidence levels. Synopsys is a leading provider of EDA software tools for automotive system-on-chip (SoC) designers, and our customers frequently ask for insight on assessing Tool Confidence Levels (TCLs) that are consistent with ISO 26262 requirements as documented in clause 11 of part 8 titled, “Confidence in the Use of Software Tools.” This white paper explains the intent of this clause and provides guidance on how to effectively meet the standard’s requirements.

Intent

Part 8:2011, clause 11.4.5 of the ISO 26262 standard provides a methodology with guidance for software tool classification and qualification. It applies to software tools used for the development of safety-critical designs where it is essential that each tool’s function operates correctly. The methodology provides guidance for cases where the relevant output of the tools can’t be fully examined or verified.

Clause 11 has two main objectives:

1. “Provide criteria to determine the required level of confidence in a software tool when applicable”
2. “Provide means for the qualification of the software tool when applicable, in order to create evidence that the software tool is suitable to be used to tailor the activities or tasks required by ISO 26262 (i.e. the user can rely on the correct functioning of a software tool for those activities or tasks required by ISO 26262)”

In general, the intent of this clause is to provide a means to assess, describe, and document your software tool confidence level in your development flow. In this case, “confidence” is defined as potential risk of systematic faults that can be introduced in your design due to a possible malfunction of the software tools in your flow. ISO 26262 defines two types of faults: systematic and random. Systematic faults are due to human error while random faults are due to physical causes.

Tool Confidence Level

Two criteria are used to determine TCL: Tool Impact (TI) and Tool Error Detection (TD). TI evaluates whether the software tool being evaluated has direct safety-related impact on the design under development. TI1 means that there is no direct impact on the design (e.g. use of Microsoft Excel), whereas TI2 signifies that there is a direct impact to the design (e.g. use of Synopsys Design Compiler® synthesis). TD establishes the level of confidence you have that there are measures to prevent and/or detect tool malfunctions. TD1 means a high degree of confidence; TD2 represents a medium degree of confidence; and TD3 indicates a lower degree of confidence.

The following table is then used to establish the TCL:

ISO 26262-8 Table 3 Determination of the Tool Confidence Level (TCL)		Tool Error Detection		
		TD1	TD2	TD3
Tool Impact	T11	TCL1	TCL1	TCL1
	T12	TCL1	TCL2	TCL3

Figure 1 – Determination of Tool Confidence Level

The TCL and corresponding Automotive Safety Integrity Level (ASIL), a risk classification scheme also defined in ISO 26262, of the item or element being developed determines what type of tool qualification (or if any) needs to be completed.

TCL1 means that no further qualification is required. It is important to note:

The goal should not be to classify all tools with a TCL1 because this is not realistic, but rather to classify the tools accurately for your specific situation and use case. You must perform additional tool qualification for those tools you have not classified as TCL1 for your use case.

If the level is TCL2 or TCL3, the qualification methods are outlined in ISO 26262-8:2011, tables 4 and 5. Figure 2 illustrates the TCL assessment process.

Software Tool Classification and Qualification Procedure

Assumption of customer use cases

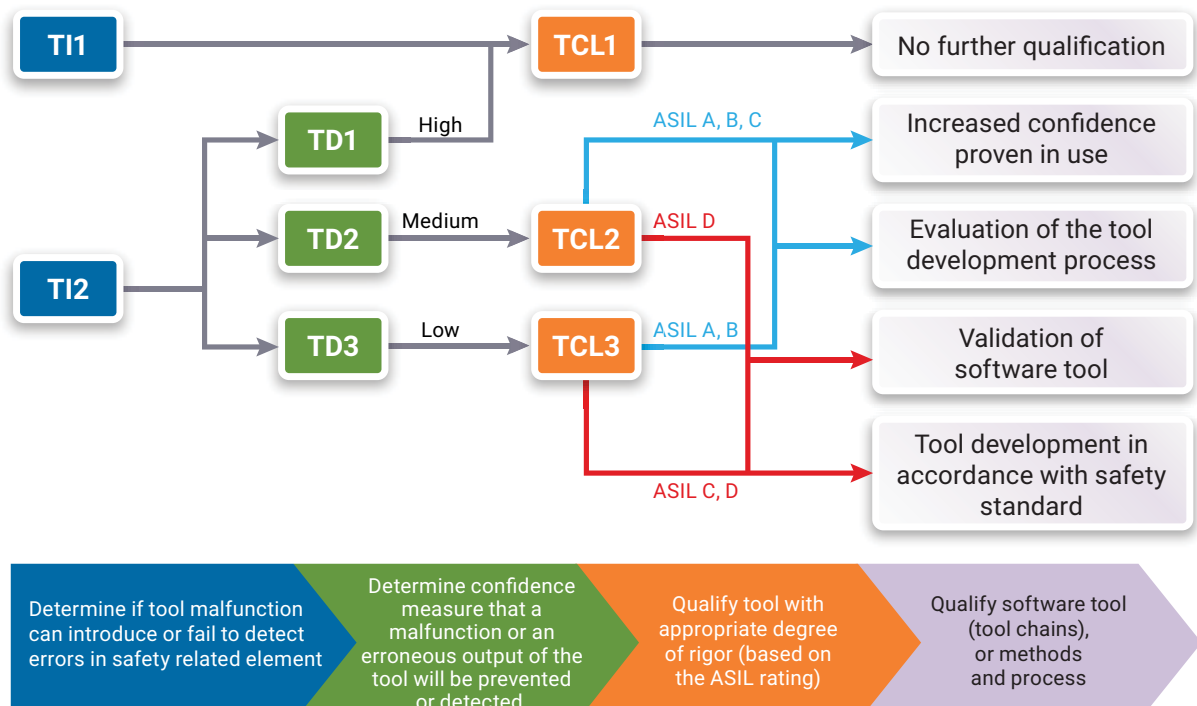


Figure 2 – TCL Assessment Flow Diagram

Who Is Responsible for TCL Assessment?

The engineering team and the company developing a safety-critical item or element is required to complete the TCL assessment and necessary tool qualification, if applicable, for the tools in their design flow and use case. Each use case needs to be assessed independently. Some software tool vendors provide sample pre-determined TCL assessments for generic use cases. While these sample assessments can be useful as a starting point for determining TCL, ultimately you must do an independent assessment with your specific use case to meet the standard's intent.

Most design flows and use cases have tools and scripts from a variety of vendors and sources—all of them must be assessed in the context of your specific use case. In the case of a pre-determined TCL, the company developing the automotive element is required to confirm the validity of the pre-determined TCL, the use case utilized, and tool qualification. The user of the software tool is responsible for the TCL value assessment, so if a pre-determined value is used, it is important that you fully agree with that assessment. It is completely acceptable (and possibly likely) for you to disagree with a pre-determined value for your specific use case; this just means that additional qualification of the software tool may be required.

TCL Assessment

As mentioned in the previous section, the engineering team using a software tool is responsible for assessing, then assigning the TCL value. The criteria for assigning a TCL1 is subjective and is up to you to define. There are many ways to support high confidence in using software tools, and one common method is to do a thorough incoming inspection and test on each tool before it is used on a safety-critical design. This testing should include running similar designs with known-good outputs, such as the extensive regression testing Synopsys performs on all tools using both internal and customer test cases and designs. To assist with tool confidence assessments, Synopsys encourages all our customers to provide their designs for incorporation into the regression suite.

Foundries also do thorough qualification testing of tools used for design implementation and signoff since they guarantee correct silicon if all rule parameters are met. These tools can include static timing analyzers, place and route, SPICE simulators, LVS/DRC physical verification, and others that design with or check foundry parametric values and rules. Foundry qualification for specific processes is a valid proxy for customer tool qualification.

Example of TCL Assessment Using a Tool Chain

Sometimes it is difficult to assess the TCL of a point tool in a development flow. One way to increase the confidence level of a point tool is to determine the TCL of a tool chain, instead. It is noted in ISO 26262-8:2011, clause 11.2, the understanding of “software tool” can vary from a separately used standalone software tool to a set of software tools integrated into a tool chain. Any user-created scripts are also part of the tool chain and its TCL assessment.

For example, the output of a synthesis tool may be assigned a TCL2 or TCL3 because the user of a synthesis tool cannot be sure that the tool, in isolation, did not insert incorrect logic. To increase the confidence level that the synthesis tool output is correct, the user could perform an independent formal verification comparison of the input RTL or netlist and the output netlist. Static timing analysis and dynamic gate-level simulations could also be used to increase confidence in the output of a synthesis tool. When a tool chain is used to determine an overall TCL, that chain, inclusive of all tools and scripts, must be documented as the use case that determined the TCL. The tool user and the safety manager must decide to what extent these comparisons/checks will be carried out and what quantitative values, if applicable, must be met. All of this should be captured in the tool chain documentation.

An example of using a tool chain to determine overall TCL is shown in Figure 3. In this example, the synthesis output (Netlist A) is checked by formal equivalence and static timing analysis tools. After successful completion of these additional steps, the output of the synthesis tool chain could be raised to TCL1. The required tool chain would be documented as Synthesis+Equivalence Checking+Static Timing Analysis=Netlist A.

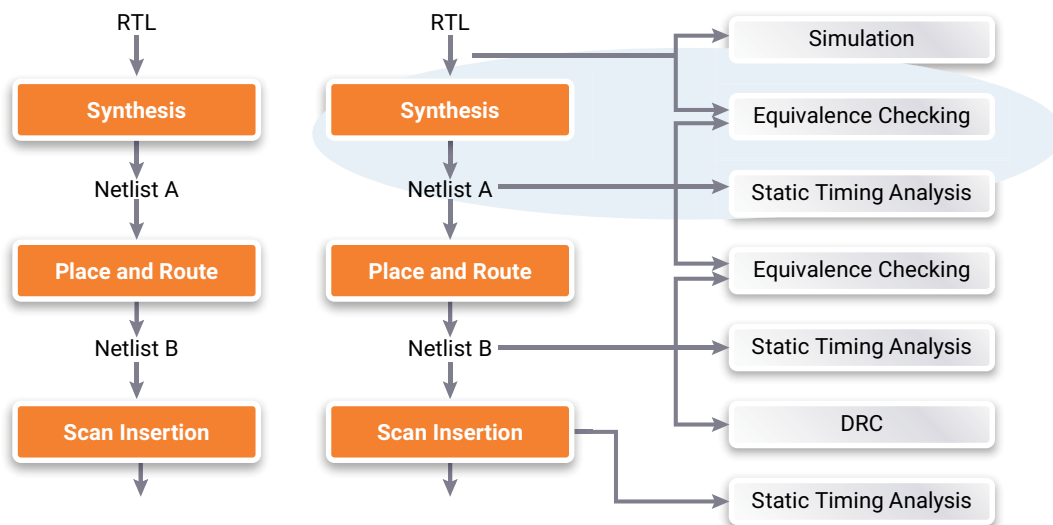


Figure 3 – Using a Tool Chain to Increase TCL

Should All Tools Be TCL1?

There is a common misconception that the goal of ISO 26262 part 8, clause 11—confidence in the use of software tools—is to assess all tools in an SoC development flow as TCL1. This is incorrect.

The actual goal is to construct and describe a development flow in which you have high confidence that the software tools you’re using do not introduce, or fail to detect, errors. It is not always possible to assess a tool with TCL1, in which case you must perform a qualification of that software tool.

In fact, it is acceptable to use software tools that are assessed at TCL2 or TCL3 as long as you perform additional qualifications. The acceptable qualification methods are listed in ISO 26262-8:2011, table 4 for tools classified as TCL3 and table 5 for tools classified as TCL2. The methods are the same for either a TCL2 or a TCL3 assessment, but different qualification methods are preferred for different ASIL classifications. Figure 4 shows a consolidation of the tables with the preferred methods highlighted and descriptions follow.

ISO 26262-8 Tables 4 and 5		TCL1	TCL2				TCL3			
Methods	ISO 26262-8 Reference	Needs no qualification methods	ASIL				ASIL			
			A	B	C	D	A	B	C	D
1a Increased confidence from use	11.4.7		++	++	++	+	++	++	+	+
1b Evaluation of the tool development process	11.4.8		++	++	++	+	++	++	+	+
1c Validation of the software tool	11.4.9		+	+	+	++	+	+	++	++
1d Development in accordance with a safety standard	*		+	+	+	++	+	+	++	++

++ Indicates that the method is highly recommended for the identified ASIL

+ Indicates that the method is recommended for the identified ASIL

NOTE: * No safety standard is fully applicable to the development of software tools.

Instead, a relevant subset of requirements of the safety standard can be selected.

Example: Development of the software tool in accordance with ISO 26262, IEC 61508 or RTCA DO-178

Figure 4 – Methods for Qualifying Software Tools

1a Increased confidence from use

Companies with experience developing automotive SoCs may have extensively documented histories of EDA tools used for similar design types. In this case, it may be possible to qualify a tool as having increased confidence from use. Method 1a requires evidence that the tool has been used for the same purpose with comparable use cases, comparable operating environment and configuration, and that the specification of the tool is unchanged. It also requires that the occurrence of malfunctions and the corresponding erroneous output(s) of the software tool acquired during previous development periods or releases be documented in a systematic way. If you are using an EDA tool for the first time on an entirely new design, or if the tool has substantially changed since you last used it, this qualification option may not be appropriate.

1b Evaluation of the tool development process

Method 1b requires access to the tool supplier's internal development process. The evaluation of the tool development process is usually performed by an assessment based on an appropriate national or international standard certification organization. Synopsys has worked with SGS TÜV Saar, a leading inspection, verification, testing and certification company, to do an independent assessment of many of our tool development processes. We provide our customers with the results of these assessments to help simplify their tool qualification efforts.

1c Validation of the software tool

For most EDA tools, the recommended qualification method to use is 1c, validation of the software tool. In this method, your design or CAD team should validate that the tool complies with your company's specified requirements in the context of its safety requirements and related use cases. You should analyze potential tool malfunctions and corresponding erroneous outputs, as well as the mitigation measures to prevent or detect them. In addition, you must consider the reaction of the tool to anomalous operating conditions (e.g., incomplete input data, incomplete installation, use of configuration, settings that are not considered "safe", or settings that are undocumented). At most large-to-medium-sized SoC design companies, CAD teams evaluate incoming new tools or tool releases using company-specific test cases, and for consistency, often run these in the form of regression tests where known-good outputs are compared with the outputs of a new tool version.

It is important that your test cases include the use cases that are specified for the safety requirements of your target design. If you do not include the use cases, you will need to develop and validate additional test cases with the tool before deploying it. In addition, complete design test cases may be used to flush an entire EDA tool flow to check for malfunctions related to tool interactions. Other sources of data for this type of qualification are the tool's user guide or release notes, and the supplier's online support center.

You may use software tools that have been assessed at TCL2 or TCL3, along with a 1c validation, for any ASIL level as specified in ISO 26262-8:2011, tables 4 and 5.

1d Development in accordance with a safety standard

The 1d method is less commonly used for EDA tools because they are usually developed and used for a wide variety of industries and applications. No safety standard is fully applicable to the development of EDA software tools for safety-critical automotive designs. Instead, you can select a subset of requirements from an existing safety standard that is relevant to safety-critical automotive designs.

The following example is listed in the "Note" below the ISO 26262-8:2011, tables 4 and 5: EXAMPLE: Development of the software tool in accordance with ISO 26262, IEC 61508, or RTCA DO-178.

What Documentation Is Required?

ISO 26262-8:2011 clause 11.4.6.2 states that the qualification of the software tool must be documented and include the following information:

- ▶ The unique identification and version number of the software tool
- ▶ The maximum Tool Confidence Level for which the software tool is classified together with a reference to its evaluation analysis
- ▶ The pre-determined maximum ASIL, or specific ASIL, of any safety requirement that might be violated if the software tool malfunctions and produces erroneous output
- ▶ The configuration and environment for which the software tool is qualified
- ▶ The person or organization who performed the qualification
- ▶ The methods applied for the tool's qualification in accordance with clause 11.4.6.2
- ▶ The results of the methods applied to qualify the software tool
- ▶ The usage constraints and malfunctions identified during the qualification, if applicable

Note: For software tool chains, all unique identifiers, version numbers, and required scripts must be documented and these define the software tool chain. The same documentation described above for individual tools is required for software tool chains.

Required Confirmation Measures

ISO 26262 requires the results of the tool confidence evaluation and related qualification results to be confirmed. The level of independency required depends on the ASIL (see ISO 26262-2 clause 6.4.7, table 1 – required confirmation measures, including the required level of independency).

How Synopsys Can Help

Synopsys provides an example automotive design flow with the respective Synopsys tools listed therein. Individual TCL assessments for each tool, along with a tool chain TCL assessment, if applicable, to increase the TCL are also available. For tools that Synopsys cannot assess at a TCL1 individually or in a tool chain, Synopsys provides the ISO 26262 required report that confirms certification of completion of method 1b evaluation or method 1c validation.

Synopsys has determined that a limited number of its EDA tools (such as TetraMAX® automatic test pattern generation, Certitude® functional verification, and Z01X™ fault simulation tools) are difficult to assess at TCL1 and, therefore, in accordance with the ISO 26262 requirements, have been independently audited and “certified” for use in safety-related designs. For these tools, Synopsys provides an Independent Assessment Report from an accredited ISO 26262 certification company, such as SGS-TÜV Saar. These certified products include the required software tool criteria evaluation report and a software tool qualification report that you can use to quickly confirm qualification, per methods 1b and 1c, of software tools classified as TCL2 or TCL3.

The software tool criteria evaluation report includes the following information:

- Tool name and version (e.g. TetraMAX Automotive, L-2016.03)
- Description (e.g. features, functions, and properties of the tool)
- Intended purpose (e.g. automatic test pattern generation)
- Documentation (e.g. user manual, safety guide)
- Environment (e.g. resources, infrastructure, or runtime environment needed to execute the tool).
Note that this may also require the specification of other processes that occur before and/or after the tool is run, such as input file checking or output file verification
- Configuration (e.g. switches, pragmas, options)
- Restrictions (e.g. incorrect combinations of tool switches, an operating environment not complying with installation or user manual) and expected tool behavior if restrictions are not followed (e.g. are there any error messages in the tool log file?)
- Errata (e.g. known software bugs and appropriate safeguards or workarounds, if applicable)

- Error prevention or detection methods (e.g. how does the tool check for errors in its results, or does it rely on another tool? What does the other tool do and can it also be relied on to operate correctly?)
- Use cases (e.g. generate optimized netlist from RTL source)
- Maximum ASIL classification of all safety requirements of the target design
- Tool Error Impact (TI1 or TI2)
- Tool Error Detection (TD1, TD2, or TD3)
- Required Tool Confidence Level (TCL1, TCL2, or TCL3) – may need multiple as per use cases
- Qualification method to be used, if applicable (1a, 1b, 1c, or 1d)

The software tool qualification report includes the following information:

- Tool name and version
- Maximum TCL for which the tool is classified and a reference to its evaluation analysis
- For the considered use cases, the maximum ASIL classification for any safety requirements in the design that may be violated if the tool malfunctions and produces erroneous output
- Identified malfunctions (e.g. tool errors or bugs found during qualification)
- Person or organization who carried out the qualification
- Method(s) used for qualification
- Results of the qualification method applied (e.g. test results). Note: if method 1d validation of the software tool is used, evidence that potential errors either do not occur or will be detected must be provided
- Analysis of results and any constraints and malfunctions identified during the qualification; descriptions of restrictions or workarounds, if necessary

Conclusion

To comply with ISO 26262, the company developing a safety-critical automotive SoC design or subsystem is responsible for assessing and documenting the TCL value for all software tools in their design flows and use cases. The goal is not to have all tools classified as TCL1, but rather to accurately classify the tools for the specific use case and to do further tool qualification as needed for those tools not classified as TCL1.

Synopsys understands that it is unlikely that all tools in design flows and use cases can be assessed at TCL1. For those tools that typically need further tool qualification, Synopsys provides an independent tool qualification assessment from an accredited ISO 26262 certification body to help you quickly and definitively meet your tool confidence level qualification needs.

**For more information on how Synopsys can help you assess ISO 26262 TCLs,
email: iso26262-tcl@synopsys.com.**