WHITE PAPER

SYNOPSYS®

Meeting ISO 26262 Guidelines with the Synopsys Software Integrity Portfolio

Meeting ISO 26262 Guidelines with the Synopsys Software Integrity Portfolio

The average car is expected to contain 300 million lines of code in the next decade, up from 100 million lines of code in today's cars. And software and electronics are expected to account for 90% of automobile innovation. Software controls everything from safety critical systems like brakes and power steering, to basic vehicle controls like doors and windows, V2V, V2I, and sophisticated infotainment systems and telematics. However, with the exponential growth of software comes a dramatic increase in software defects. The average car is expected to contain up to 150,000 bugs, many of which could damage the brand, hurt customer satisfaction and, in the most extreme case, lead to a catastrophic failure. Jaguar, a leading brand in luxury cars, was forced to recall nearly 18,000 X-type cars after it discovered a major software fault which meant drivers might not be able to turn off cruise control, which could put the vehicles' occupants in jeopardy.

Introduction to ISO 26262

To help address vehicle safety, the International Organization for Standardization (ISO) put forth ISO 26262 for road vehicle functional safety. The standard was created to provide guidance to avoid the risk of systematic failures and random hardware failures through feasible requirements and processes. ISO 26262 is the adaptation of IEC 61508 to comply with needs specific to the application sector of electric and or electronic elements such as power supplies, sensors and other input devices, data highway, and other communication paths, actuators, and other output devices. The purpose of this paper is to discuss how the Synopsys Software Integrity Portfolio can be used to help meet the guidelines set forth in ISO 26262.

The standard is comprised of 10 parts that span the breadth of the automotive safety lifecycle including management, development, production, operation service and decommissioning. The Synopsys Software Integrity Portfolio products, Synopsys Static Analysis (Coverity) and Synopsys Software Test Optimization (Test Advisor), apply most directly to Part 6: Product Development: Software Level.

Introduction to the Synopsys Software Integrity Portfolio

Development testing is designed to help developers, management, and the business easily find and fix quality and security problems early in the software development lifecycle, as the code is being written, without impacting time-to-market, cost or customer satisfaction.

Development testing augments traditional testing, including Quality Assurance (QA) functional and performance testing and security audits, providing development teams with a quick and easy way to test their code for defects and to ensure critical code has been properly tested in a non-intrusive manner. This enables development to stay focused on innovation, management to get visibility into problems early in the cycle to make better decisions, and the business to continue to deliver high-quality products to market for competitive advantage.

Synopsys provides the industry's leading development testing portfolio with tailored solutions for development and management teams that can assist organizations with achieving ISO 26262 compliance.

In addition, Synopsys Static Analysis and Synopsys Software Test Optimization are certified by TUV SUD Product Service GmBH according to the applicable requirements of the standard IEC 61508 and ISO 26262 for developing and testing safety critical software.

Synopsys Static Analysis – Synopsys delivers the industry's most accurate static analysis solution. It is used by developers around the world to improve the quality of their code by enabling them to find and fix defects in C/C++, Java and C# code faster which results in lower overall costs. Organizations can create customized analysis rules to support their unique requirements through the Synopsys Static Analysis Software Development Kit (SDK). Static analysis is included in ISO 26262 as a formal verification method for adherence to the coding guidelines and can be used for reviewing pieces of code that access memory locations containing safety-related data as specified in ISO 26262 Annex D, freedom from interference by software partitioning.

Coverity Architecture Analysis – This solution enables organizations to visualize and control the complexity in their software architecture, which is a critical component of ISO 26262. In addition to controlling architectural complexity, it helps software architects and developers reduce risk by visualizing the code structure to identify dependency conflicts and interface violations, eliminating excess code complexities and security vulnerabilities through policy enforcement, and providing function call graphs to reduce debugging time.

Coverity Connect – This solution provides a centralized defect management workflow that enables developers and managers to quickly view defects in the source code and take the appropriate action to resolve them. Developers and managers can identify defects associated with a particular Automotive Safety Integrity Level (ASIL) and find where defects occur across various code branches. This capability is a critical time saver for development teams as code reuse is prevalent in the automotive industry.

Coverity Policy Manager – This solution enables organizations to establish and enforce consistent policies tied to safety requirements defined in ISO 26262, by ASIL level. It enables users to define clear and comprehensible policies to meet the key requirements for this standard. Once the policies have been established, organizations can test against them with the Synopsys Static Analysis development testing solutions, and quickly visualize areas of risk in the project by component and ASIL level. Managers and executives get a hierarchical view of risk, can understand the relative effort required to address the defect, and can drill down to details to pinpoint the specific issues or verify that specific safety requirements have been satisfied.

Synopsys Software Test Optimization – As more companies find themselves in the business of software, including those in the automotive segment, they are increasingly realizing they need to test their code as they develop it. For many, this can be overwhelming as developers don't know where to start or when they are finished with their testing efforts. Synopsys Software Test Optimization brings intelligence to the unit testing process by enabling developers to focus their efforts on the most critical components of the code, such as changed code or code impacted by change versus chasing a naïve coverage number that treats all lines of code equally. Synopsys Software Test Optimization enables organizations to establish and enforce a consistent policy for their automated testing and provides a uniform workflow which integrates seamlessly into the development process.

Applying the Synopsys Software Integrity Portfolio to ISO 26262 Requirements

Automotive Safety Integrity Levels (ASILs)

ISO 26262 uses one of four ASILs (A, B, C or D) to specify the item's or element's necessary safety requirements for achieving an acceptable level of risk, with D representing the most stringent and A the least stringent level.

Elements with an ASIL of D are expected to be tested with the greatest level of rigor.

The Synopsys Software Integrity Portfolio enables organizations to annotate and visualize ASIL levels at the software component level. With Coverity Policy Manager, users can establish and enforce consistent policies for the coding guidelines and design principles for software unit design and implementation required by ASIL level. Managers can view policy compliance in a hierarchical view and by ASIL level so teams can quickly address areas of risk in their projects.

With Synopsys Static Analysis, teams can test against the required policies from within their IDE or as part of the central build process. Static Analysis testing is one of the required methods for the verification of adherence to software coding guidelines and design and implementation principles. Within Coverity Connect, the centralized developer workflow for defect management, developers can quickly search for defects based on specific ASIL classifications. For example, a user may search on all recursion defects with an ASIL classification of D or all defects associated with ASIL level of C.

Synopsys Software Test Optimization enables development teams to ensure that their critical code has been covered by a unit test as required by clause 6-9 of the standard. Testing inadequacies are automatically identified and routed to the appropriate developer for resolution as part of the standard development workflow. Organizations can establish and enforce stage gate criteria that all critical code must be covered by an automated test before it is released to QA.

Software Modeling and Coding Guidelines

As part of the initiation of the product development phase at the software level, ISO 26262 created a set of coding and modeling guidelines which are published in Table 1 of the Software Development Module. The Synopsys Software Integrity Portfolio supports these guidelines in the following manner:

Table Legend	
SSA = Synopsys Static Analysis	++ indicates highly recommended
CAA = Coverity Architecture Analysis	+ indicates recommended
CPM = Coverity Policy Manager	o indicates not required
SSTO = Synopsys Software Test Optimization	

ISO 26262 Table 1: Topics to be covered by modeling and coding guidelines					
		ASIL			
Topic	Synopsys Software Integrity Portfolio Support	A	B	C	D
Enforcement of low complexity (1a)	SSA will analyze code and compute cyclomatic complexity and Halstead metrics. CPM can then be used to establish and enforce policies for low code complexity. CAA further enables users to reduce complexity by identifying circular dependencies and excessive complexity.	++	++	++	++
Use of language subsets (1b)	SSA detects non-standard language constructs and flags them as actionable defects to the developer. Additional checks can be created with the Coverity SDK.	++	++	++	++

ISO 26262 Table 1: Topics to be covered by modeling and coding guidelines					
Topic	Synopsys Software Integrity Portfolio Support	ASIL			
		A	B	C	D
Enforcement of strong typing (1c)	C and C++ are considered less strongly typed than other languages such as Java because of their support for implicit and explicit casting. SSA will automatically find unsafe casting and flag the occurrence as a defect. Additional checks can be created through the Coverity SDK. For example if casting is disallowed, a custom checker could be created to create a defect for every cast operation.	++	++	++	++
Use of defensive implementation techniques (1d)	SSA enforces defensive programming by highlighting as an error failure to check return value of any function; not just checking for null but verifying or testing returned value for possible error conditions.	O	+	++	++
Use of established design principles (1e)	CAA enables architects to establish design principles and provides them with the visibility and control needed to ensure the design specifications do not degrade over time and unwanted dependencies are not introduced during development. Through the Coverity SDK custom analysis rules can be created to test for specific violations of select design principles such as the use of global variables.	+	+	+	++
Use of unambiguous graphical representation (1f)	This guideline is not applicable to C/C++ , Java or C.	+	++	++	++
Use of style guides (1g)	The extensible Synopsys Software Integrity Portfolio enables organizations to integrate third-party tools, such as style guides, into the platform and manage violations of the guide in a common user interface.	+	++	++	++
Use of naming conventions (1h)	The Coverity SDK can be used to create a custom check for naming convention violations.	++	++	++	++

Software Architectural Design

Once the initiation of the product design phase has commenced, the next step in the guideline is to address software architectural design. The objective of this phase is to develop an architectural design that realizes the software safety requirements. The architectural design represents all software components and their interactions with one another in a hierarchical structure. Static aspects such as interfaces and data paths of all software components, as well as dynamic aspects such as process sequences and timing behavior, need to be described. The Coverity Architecture Analysis and Coverity Policy Manager can help achieve the desired objectives of modularity, encapsulation and minimal complexity.

ISO 26262 Table 3: Principles for software architectural design					
Topic	Synopsys Software Integrity Portfolio Support	ASIL			
		A	B	C	D
Hierarchical structure of software components (1a)	Synopsys Static Analysis provides a graphical representation of the hierarchical structure of software components through both CAA and CPM.	++	++	++	++
Restricted size of software components (1b)	CPM provides a graphical representation into the size of software components and custom checkers could be created to identify modules that have exceeded the maximum size allowed.	++	++	++	++
Restricted size of interfaces (1c)	CAA can be used to visualize the size of interfaces.	+	+	+	+
High cohesion within each software component (1d)	A third-party tool focused on software cohesion could be integrated into the extensible Synopsys Static Analysis.	+	++	++	++
Restricted coupling between software components (1e)	CAA can be used to view relationships and dependencies between components.	+	++	++	++
Appropriate scheduling properties (1f)	A third-party tool could be integrated into the extensible Synopsys Static Analysis platform to address scheduling properties.	++	++	++	++
Restricted use of interrupts (1g)	The Coverity SDK can be used to create a custom check for behavior such as interrupts to the system hardware.	+	+	+	++

Software Unit Design and Implementation

Once the architectural design is complete, the next stage in the ISO 26262 standard is software unit design and implementation.

The standard supplies numerous guidelines for software design and implementation to ensure the correct order of execution, consistency of interfaces, correctness of data flow and control flow, simplicity, readability and comprehensibility and robustness.

ISO 26262 Table 8: Design principles for software unit design and implementation						
Topic	Synopsys Software Integrity Portfolio Support	Rule Mapping	ASIL			
			A	B	C	D
One entry and one exit point in subprograms and functions (1a)	SSA automatically analyzes return statements to determine if more than one entry or exit points exists in a component or function.	MISRA C 2004 Rules 14.4 and 14.7, MISRA C 2012 Rules 15.1 and 15.5	++	++	++	++
No dynamic objects or variables, or else online testing during their creation (1b)	SSA automatically analyzes the code to identify the use of dynamic objects are properly tested during their creation. For example, users can analyze the code to ensure that if malloc() is used, the return must be checked. Through the Coverity SDK, customer checkers could be created to identify heap allocation calls.		+	++	++	++
Initialization of variables (1c)	SSA automatically tests the code for uninitialized variables.	MISRA C 2004 Rule 9.1, MISRA C 2012 Rules 9.1 and 9.4	++	++	++	++
No multiple use of variable names (1d)	SSA automatically creates parse warnings which appear as actionable defects to the developer for such issues as local hiding local, local hiding parameter and linkage conflict issues.	MISRA C 2004 Rule 5.5, MISRA C 2012 Rules 5.8 and 5.9	+	++	++	++
Avoid global variables or else justify their usage (1e)	The Coverity SDK can be used to create a custom checker for the use of global variables.		+	+	++	++
Limited use of pointers (1f)	The Coverity SDK can be used to create a custom checker for imposing limitations on pointer arithmetic.		O	+	+	++
No implicit type conversions (1g)	The Coverity SDK can be used to create custom checks for additional implicit type conversions.		+	++	++	++
No hidden data flow or control flow (1h)	The Coverity SDK can be used to create a custom checker to ensure no hidden data flow or control flow exists. For example, the checker could analyze the code for goto pointers and flag those as potential defects.		+	++	++	++

ISO 26262 Table 8: Design principles for software unit design and implementation						
Topic	Synopsys Software Integrity Portfolio Support	Rule Mapping	ASIL			
			A	B	C	D
No unconditional jumps (1i)	SSA automatically checks for use of gotos and other unconditional jumps	MISRA C 2004 Rule 14.4, MISRA C 2012 Rule 15.1	++	++	++	++
No recursions (1j)	SSA automatically checks for recursions and flags occurrences as actionable defects.	MISRA C 2004 Rule 16.2, MISRA C 2012 Rule 17.2	+	+	++	++

ISO 26262 Table 9: Methods for the verification of software unit design and implementation					
Topic	Synopsys Software Integrity Portfolio Support	ASIL			
		A	B	C	D
Control flow analysis (1e)	SSA in depth analysis covers all paths of the control flows	+	+	++	++
Data flow analysis (1f)	SSA in depth analysis covers all paths of the data flows	+	+	++	++
Static code analysis (1g)	SSA automatically performs static code analysis	+	++	++	++
Semantic code analysis (1h)	SSA automatically performs semantic code analysis	+	+	+	+

Software Unit Testing

Software unit testing is an important requirement in the ISO26262 standard. Software unit tests must be planned, specified and executed. Synopsys Software Test Optimization provides statement coverage and can be used to ensure that safety critical code and risky code such as changed code or legacy code impacted by change has a unit test associated with it. Testing inadequacies can be automatically managed and routed to the appropriate developer for immediate resolution.

ISO 26262 Table 15: Structural coverage metrics at the software unit level					
Topic	Synopsys Software Integrity Portfolio Support	ASIL			
		A	B	C	D
1a	Synopsys Software Test Optimization provides statement coverage	++	++	+	+

Summary

The Synopsys Software Integrity Portfolio can help organizations comply with the coding guidelines, architectural design and unit testing requirements set forth in the ISO 26262 standard to ensure the functional safety of the automobiles they produce and better manage the increased code complexity.

THE SYNOPSYS DIFFERENCE

Synopsys offers the most comprehensive solution for integrating security and quality into your SDLC and supply chain. Whether you're well-versed in software security or just starting out, we provide the tools you need to ensure the integrity of the applications that power your business. Our holistic approach to software security combines best-in-breed products, industry-leading experts, and a broad portfolio of managed and professional services that work together to improve the accuracy of findings, speed up the delivery of results, and provide solutions for addressing unique application security challenges. We don't stop when the test is over. Our experts also provide remediation guidance, program design services, and training that empower you to build and maintain secure software.

For more information go to www.synopsys.com/software

SYNOPSYS®
185 Berry Street, Suite 6500
San Francisco, CA 94107 USA

U.S. Sales: (800) 873-8193
International Sales: +1 (415) 321-5237
Email: software-integrity-sales@synopsys.com