White Paper

An ISO 26262 Automotive Semiconductor Safety Primer

Understanding the implications of the ISO 26262 standard on electronic systems development

Version: 191125
November 2019

Optima-DA Confidential

Abstract

The ISO 26262 Automotive Safety Standard provides critical guidance on the development of electronic hardware with a tolerable level of risk for modern electric and autonomous vehicles. However, the standard, and derivative works tend to be filled with jargon, acronyms, and preconceived knowledge requirements that make understanding the subject difficult and time consuming. This paper attempts to provide a clear, concise description of the key elements of the standard for developers, such that a working knowledge of its methodology requirements may be achieved.

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement no. 850104

Optima Design Automation Inc.
www.optima-da.com

Introduction

Ensuring that semiconductors used in applications where safety is paramount will not fail is clearly very important. However, in many applications such as the aerospace industries, this has meant that silicon technologies, including development environments, require extensive proof and this can require many years of operational experience. This is unacceptable in a fast-moving, highly competitive environment such as the automotive industry.

With Advanced Driver Assistance Systems (ADAS) literally hitting the streets, automotive devices represent some of the most complex electronic systems being devised today. The rationale behind the ISO 26262 standard is, therefore, to maximize device safety while still enabling a technology excellence. This dichotomy leads to a requirement for new methodologies and tools that can facilitate both safe and advanced automotive Integrated Circuit (IC) development.

Electronic Automotive Devices Today

The modern vehicle contains a plethora of electronic systems, as shown in Figure 1. Some of these are obvious, such as radar, entertainment, and navigation, however, most are hidden from the driver, but are integral to the vehicle's basic operation. The next-generation of autonomous vehicles will require advanced computer-based artificial intelligence.

A vehicle in 2018 typically contained 100-300 micro-controllers or processors, 50+ complex electronic control units, between 5 and 20 million lines of software code, and miles of wire harness to connect them.

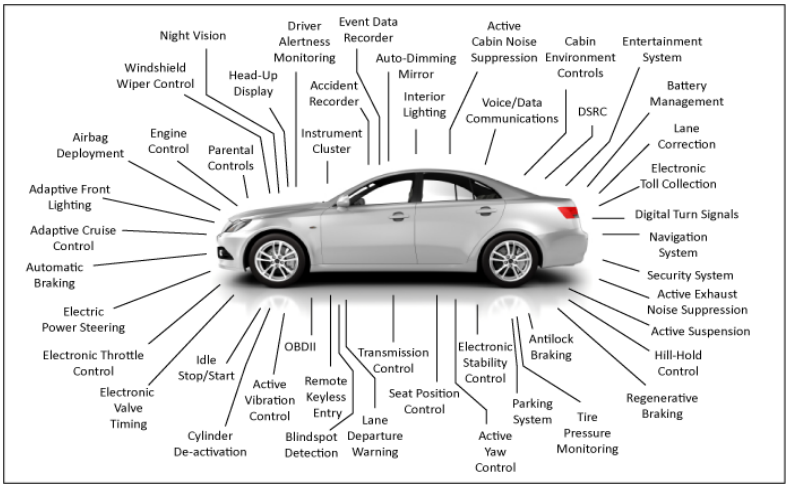

Figure 1: Electronic Components in a Modern Vehicle
Courtesy: Vehicular Electronics Laboratory, Clemson University

All of these electronic devices must be considered "safe" to a specified tolerable level of risk. In the past, ensuring safety was the purview of the vehicle manufacturer and systems providers. Today, the entire automotive value chain treats safety as a primary goal, and this is the purpose of the automotive safety standards. The ISO 26262 [1] provides a definition of functional safety as: "the absence of unreasonable risk due to hazards caused by malfunctioning behavior of electrical/electronic systems."

An Introduction to ISO 26262

Figure 2 shows a general hierarchy of active safety standards for different applications. The quality requirements for automotive hardware components are documented in the ISO (International Organization for Standardization) 26262 standard, named "Road vehicles – Functional safety", ratified in 2011. It is an adaptation of the IEC (International Electro-technical Commission) standard 61508, named "Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems", and borrows many aspects of this generalized standard. A second edition of the standard, ISO 26262-1:2018, was issued in December 2018.

Functional Safety Standards						
DO 254	IEC 61508					
	ISO 26262	IEC 61511	IEC 62061	IEC 61513 IEC 62138	IEC 62404 ISO 13485	EN 50128
Areospace Defense	Automotive	Industrial Controls	Machine Tooling	Nuclear Power	Medical Devices	Railway Transport

Figure 2: Hierarchy of Safety Standards

ISO 26262 notes two failure types that must be considered during electronic system and semiconductor development.

- Systematic Failures
  These failures are related to the development, manufacturing, etc., of devices and dictate the use of effective processes to minimizing the risk of issues introduced during automobile production.

- Random Failures
  Random failures are issues that occur during the operation of a device due to various reasons including environmental effects, device malfunction, etc. Minimizing these risks involve the analysis of the device for the effect of operational faults, and building systems into the device to protect against these.

Risk Categorization

A key aspect of the standard is the determination of the risk level associated with these failures. The Automotive Safety Integrity Level (ASIL) is defined as the level of risk reduction needed to achieve a tolerable exposure, and four ASIL levels from A to D, where D is the highest, are defined by considering the following:

- Exposure: the probability of the operational conditions that lead to injury?
- Severity: what is the level of harm caused by the risk?
- Controllability: what is the extent of the driver's ability to contain the damage?

These are categorized as shown in the following table:

Exposure		Severity		Controllability	
E1	Extremely low probability	S1	No injuries	C1	Easy control
E2	Low probability <1%	S2	Light injury	C2	Normal control
E3	Medium probability 1-10%	S3	Life threatening	C3	Uncontrollable
E4	High probability >10%				

These categories are combined as shown in this table to rate the device ASIL:

	S1				S2				S3			
	E1	E2	E3	E4	E1	E2	E3	E4	E1	E2	E3	E4
C1	QM	QM	QM	QM	QM	QM	QM	A	QM	QM	A	B
C2	QM	QM	QM	A	QM	QM	A	B	QM	A	B	C
C3	QM	QM	A	B	QM	A	B	C	A	B	C	D

As an example, an anti-lock brake controller would be categorized S3 because a malfunction can be life threatening, C3 because a driver has no way to control it if an error occurs and E4 because it has a high probability of occurring, rendering a risk tolerance of ASIL-D.

Systematic Failure Management

Although the majority of this paper is focused on the subject of Random Failures, we also provide a brief discussion on Systematic Failure management. In general, the semiconductor industry is well prepared to handle systematic faults using its existing verification and design-for-testing methodologies and technologies. It is the Random Failures that present greater difficulties and require more focus.

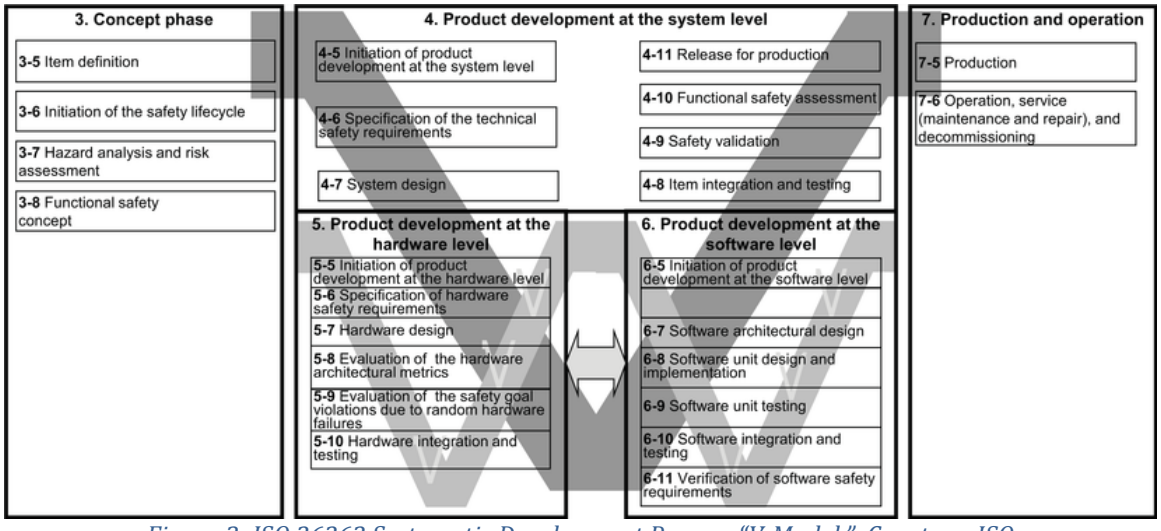

Figure 3: ISO 26262 Systematic Development Process "V-Model." Courtesy: ISO

For development environments, minimizing the risk of systematic failures involves the careful specific of requirements, rigorous design practices to meet those requirements, highly effective verification techniques, and the measurement of coverage to close the loop. These processes are depicted in the well-known "V-Model," as shown in Figure 3.

The V-Model leads to a development process that involves the careful specific of individual requirements, followed by each requirement being provided its own implementation plan describing specific features. Each feature is given a verification goal and a plan for achieving these goals. Functional coverage is measured during verification regardless of the tools used, and these coverage metrics fed back to the original plans to ensure the goals are met. Figure 4 shows an example of a development process in use.

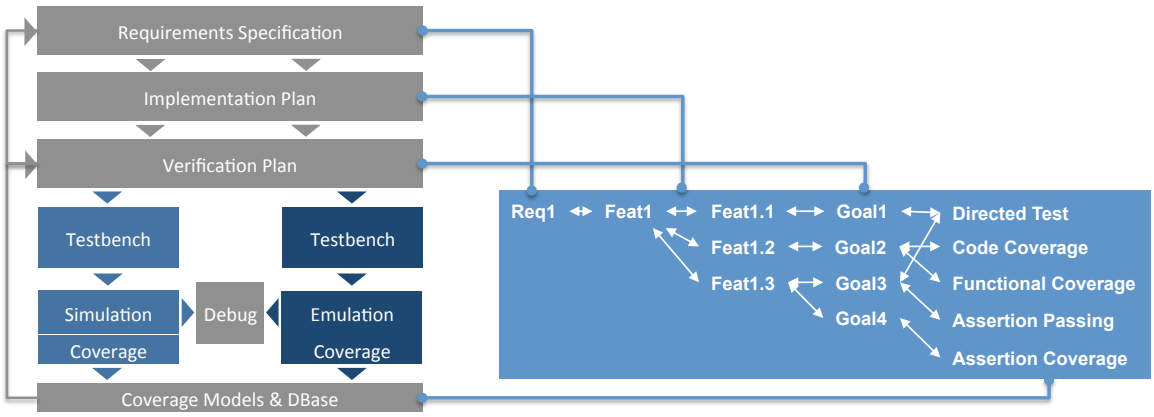

Figure 4: Decomposition of requirements into development processes. Source: TV&S

Handling Random Failures

A random failure is a fault that occurs in the system during its operation, caused by environmental effects, for example, radiation from the sun flipping a memory bit or heat exposure causing a change of state in a key register. Potential faults must be measured to understand their contribution towards the random failure risk and where possible eliminated, such that a certain level of overall fault tolerance is achieved.

Some of these faults will not exist in a dangerous area of the design or will not cause an issue even if they do occur due to some artifact of the design logic operation. Other faults will create an operational problem and their effect needs to be mitigated. This is performed using a Safety Mechanism (SM), which detects and, if possible, corrects the fault. The various fault types will be discussed below, and shown in Figure 6.

Examples of Safety Mechanisms include:

- Error Correcting Codes (ECC)
  Commonly used for memory protection, ECC involves coding the data input into the memory or logic using Hamming or some other form of coding, see Figure 5. When the data is read, it is decoded. Hamming codes will error correct single bit errors, and detect multi-bit errors.

- TMR / DMR
  Dual and Triple Modular Redundancy refers to the general scheme of doubling or tripling a design, providing them the same inputs and comparing the outputs. In the triple modular case if two outputs are the same and one different, the mechanism uses the two common outputs, thereby correcting random errors. In the dual case, if the outputs are different, an error is detected. TMR and DMR can be used at different levels, including the flip-flop level, module-level, unit level, chip-level and even system level in some cases.

- Lock Step
  One particular version of DMR, where a master circuit element and an identical diagnostic element are run lock step, clock cycle by cycle, together and the outputs compared for issues. This is common for processor subsystem checking.

- DMR Hardened Flip Flops
  Another version of DMR, where the redundancy mechanism is implemented at the flip-flop level, see Figure 5. This is particularly effective for trapping transient faults (aka soft-errors), in a so-called “Design Hardening” process.

- Built-In Self Test (BIST)
  In this mechanism, a test sequence is run over a section of the device, either on start up or other intervals. The BIST may be applied at the inputs or over a scan path, and the resulting output coded into a signature. If the signature is incorrect, a fault exists.

- Cyclic Redundancy Checking (CRC)
  A common fault correction capability used in communications devices, CRC has been applied to fault tolerant circuits as well. A serial sequence of bits is encoded and the code transmitted with those bits. If the code is incorrect, then one of more of the bits is incorrect.

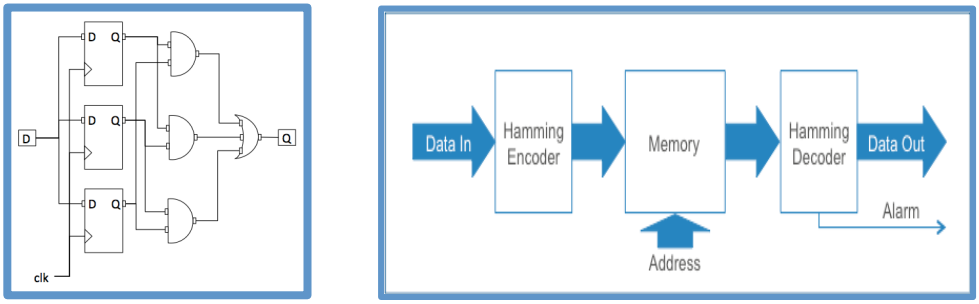

Figure 5: DMR Hardened Flip-Flop and Error Correcting Code (ECC) Safety Mechanisms

Fault Types or Models

Faults may be broadly translated into Permanent Faults, also known as "Hard Errors," and Transient Faults, also known as "Soft Errors." As the names suggest permanent faults are faults that, once manifested, do not go away. These may consist of Stuck at 1 or Stuck at 0 faults, bridging faults (two signals stuck together), and tri-state faults (signal going to an un-driven, or Z, state). Transient faults are faults that exist for a short amount of time before the signal returns to regular operation. These two fault types require handling using different methods, but both are important for risk analysis.

Finally, the timing of the occurrence of a fault and its resolution is also important. The system must be able to detect a fault and transition to a safe state within the "Fault Tolerant Time Interval" or FTTI. The FTTI varies for different types of faults, for example many of the above safety mechanisms the correction is within one or two clock cycles. For a processor, the FTTI can be measured in a few milliseconds.

Fault Classification and ASIL Estimation

To understand how these faults affect the ASIL category, we must first have an understanding of the various fault types. There are various fault types defined in ISO 26262, as shown in Figure 6.

Note the naming of these fault types may be considered somewhat ambiguous, so it is worth reviewing them:

- Safe Faults: Are located in part of the logic not relevant for the safety of the device, or they do not impact the safety of the device.
- Single Point Faults: These faults impact the safety of the device and are not handled by a safety mechanism. As such, these are dangerous faults.
- Residual Faults: Occurs in an area monitored by a safety mechanism, but cannot be handled by the safety mechanism.
- Multipoint Faults: These faults are handled by the safety mechanism.

The Multipoint Faults are sub-classified into:

- Detected: Detected and corrected by the safety mechanism

- Latent: Corrected by the safety mechanism, but with no indication they existed
- Perceived: Not detected and have some effect on the driving experience (note: this last category rarely applies to digital ICs)

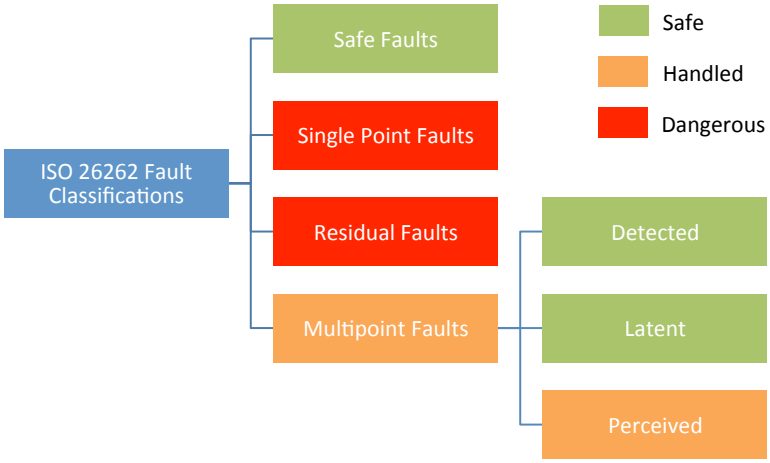

Figure 6: ISO 26262 Fault Classifications

The faults that need to be measured are the Single Point Faults and the Residual Latent Faults. Three metrics are established to provide a measurement of these faults:

- Failure-in-Time (FiT) metric is defined as one failure in 1 billion hours
- SPFM is the Single Point Fault (SPF) Metric – measurement of SPF coverage
- LFM is the Latent Fault Metric – a measurement of latent fault coverage

Using these metrics, ISO 26262 defines a level of fault tolerance for different ASIL ratings as follows:

ASIL	FiT	SPFM	LFM
A	Irrelevant	Irrelevant	Irrelevant
B	< 1000	> 90%	> 60%
C	< 100	> 97%	> 80%
D	< 10	> 99%	> 90%

Functional Safety Analysis

Understanding the risk level associated with an automotive device is a complex and time consuming business. The Failure Mode Effect and Diagnostic Analysis (FMEDA) process involves either a quantitative operation where potential faults are inspected and their effect calculated, or a qualitative inspection where safety mechanisms are described and their effect noted. Many automotive device suppliers have derived their own FMEDA methodology based on a combination of the above, combined with statistical measures, to present an abstract derivative of the device ASIL rating.

The reason for this seemingly arbitrary approach is the cost and time required to perform a fully quantitative analysis. Traditional fault simulation, intended for manufacturing test analysis, is often used. Fault simulation involves running a simulator that exercises design functionality, and then rerunning the same simulation while injecting a fault into the design to check for a change in operation. This is repeated for all the potential faults in the design. Although optimizations are applied to this process, a design of say 1M gates might have many 100,000s of faults. The run time of a fault simulation relates to the product of these two numbers, and can be measured in weeks. As such, comprehensive fault analysis can occupy several months at the end of the development process, prohibitive in terms of time.

This time constraint is being mitigated by new technology. Optima's advanced and specially designed Fault Injection Engine (FIE™) technology platform, which in benchmarks has demonstrated performance improvement greater than 1000 times the nearest alternative simulator, is one such new approach. Other approaches can also aid with this issue, including the use of formal verification, SM synthesis approaches, etc. The use of FIE and derivative "Apps" will be the subject of a second paper in this series.

In addition to Hardware Random Fault Analysis, the standard also requires Dependent Failure Analysis (DFA) to be performed. DFA consists of an analysis of common faults between different elements in a design, for example clocking schemes, reset schemes, faults introduced by test elements such as scan paths, etc. Countermeasures for these kinds of faults must be introduced and these also need to be detailed.

Summary

This paper is designed to provide a clear and concise description of the key elements of the ISO 26262 standard for Hardware IC development. It is hoped that some aspects of the standard and automotive safety solutions in general have been demystified by making the paper an easy read while trying to limit the jargon that is often used.

The reader has been provided a description of automotive issues today and the ISO standard that was created to provide protection for manufacturers and end-user alike. The issues associated with device development and operation were discussed, together with basic methods used to mitigate these issues. Finally the analysis of the risk level associated with a device, and methods used to assess this were discussed.

It may be observed that this subject is complex and is still in a state of evolution, where new technologies and methodologies are often being proposed. The standard itself is not standing still, with new issues, such as device security, now under consideration. This moving target means that the industry must continue to develop and innovate.

Part 2 of this paper series will look at a new technology for fault analysis that shows great promise in providing a qualitative analytical methodology for fault analysis that can be executed in a sensible time frame, and methods by which this is applied to different fault types.

References

[1] The International Organization for Standardization (ISO) 26262 Standard, Parts 4,5,8 https://www.iso.org/obp/ui/#iso:std:43464:en