FINAL
DRAFT

INTERNATIONAL
STANDARD

**ISO/FDIS
26262-5**

ISO/TC **22**/SC **32**

Secretariat: JISC

Voting begins on:
2018-07-19

Voting terminates on:
2018-09-13

Road vehicles — Functional safety —

Part 5:
Product development at the hardware level

Véhicules routiers — Sécurité fonctionnelle —

Partie 5: Développement du produit au niveau du matériel

Reference number
ISO/FDIS 26262-5:2018(E)

COPYRIGHT PROTECTED DOCUMENT

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see the following URL: www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 22, *Road vehicles*, Subcommittee SC 32, *Electrical and electronic components and general system aspects*.

This edition of ISO 26262 series of standards cancels and replaces the edition ISO 26262:2011 series of standards, which has been technically revised and includes the following main changes:

— requirements for trucks, buses, trailers and semi-trailers;

— extension of the vocabulary;

— more detailed objectives;

— objective oriented confirmation measures;

— management of safety anomalies;

— references to cyber security;

— updated target values for hardware architecture metrics;

— guidance on model based development and software safety analysis;

— evaluation of hardware elements;

— additional guidance on dependent failure analysis;

— guidance on fault tolerance, safety related special characteristics and software tools;

— guidance for semiconductors;

— requirements for motorcycles;

— general restructuring of all parts for improved clarity.

A list of all parts in the ISO 26262 series can be found on the ISO website.

Introduction

The ISO 26262 series of standards is the adaptation of IEC 61508 series of standards to address the sector specific needs of electrical and/or electronic (E/E) systems within road vehicles.

This adaptation applies to all activities during the safety lifecycle of safety-related systems comprised of electrical, electronic and software components.

Safety is one of the key issues in the development of road vehicles. Development and integration of automotive functionalities strengthen the need for functional safety and the need to provide evidence that functional safety objectives are satisfied.

With the trend of increasing technological complexity, software content and mechatronic implementation, there are increasing risks from systematic failures and random hardware failures, these being considered within the scope of functional safety. ISO 26262 series of standards includes guidance to mitigate these risks by providing appropriate requirements and processes.

To achieve functional safety, the ISO 26262 series of standards:

a) provides a reference for the automotive safety lifecycle and supports the tailoring of the activities to be performed during the lifecycle phases, i.e., development, production, operation, service and decommissioning;

b) provides an automotive-specific risk-based approach to determine integrity levels [Automotive Safety Integrity Levels (ASILs)];

c) uses ASILs to specify which of the requirements of ISO 26262 are applicable to avoid unreasonable residual risk;

d) provides requirements for functional safety management, design, implementation, verification, validation and confirmation measures; and

e) provides requirements for relations between customers and suppliers.

The ISO 26262 series of standards is concerned with functional safety of E/E systems that is achieved through safety measures including safety mechanisms. It also provides a framework within which safety-related systems based on other technologies (e.g. mechanical, hydraulic and pneumatic) can be considered.

The achievement of functional safety is influenced by the development process (including such activities as requirements specification, design, implementation, integration, verification, validation and configuration), the production and service processes and the management processes.

Safety is intertwined with common function-oriented and quality-oriented activities and work products. The ISO 26262 series of standards addresses the safety-related aspects of these activities and work products.

Figure 1 shows the overall structure of the ISO 26262 series of standards. The ISO 26262 series of standards is based upon a V-model as a reference process model for the different phases of product development. Within the figure:

— the shaded “V”s represent the interconnection among ISO 26262-3, ISO 26262-4, ISO 26262-5, ISO 26262-6 and ISO 26262-7;

— for motorcycles:

  — ISO 26262-12:2018, Clause 8 supports ISO 26262-3,

  — ISO 26262-12:2018, Clauses 9 and 10 support ISO 26262-4;

— the specific clauses are indicated in the following manner: “m-n”, where “m” represents the number of the particular part and “n” indicates the number of the clause within that part.

EXAMPLE “2-6” represents ISO 26262-2, Clause 6.

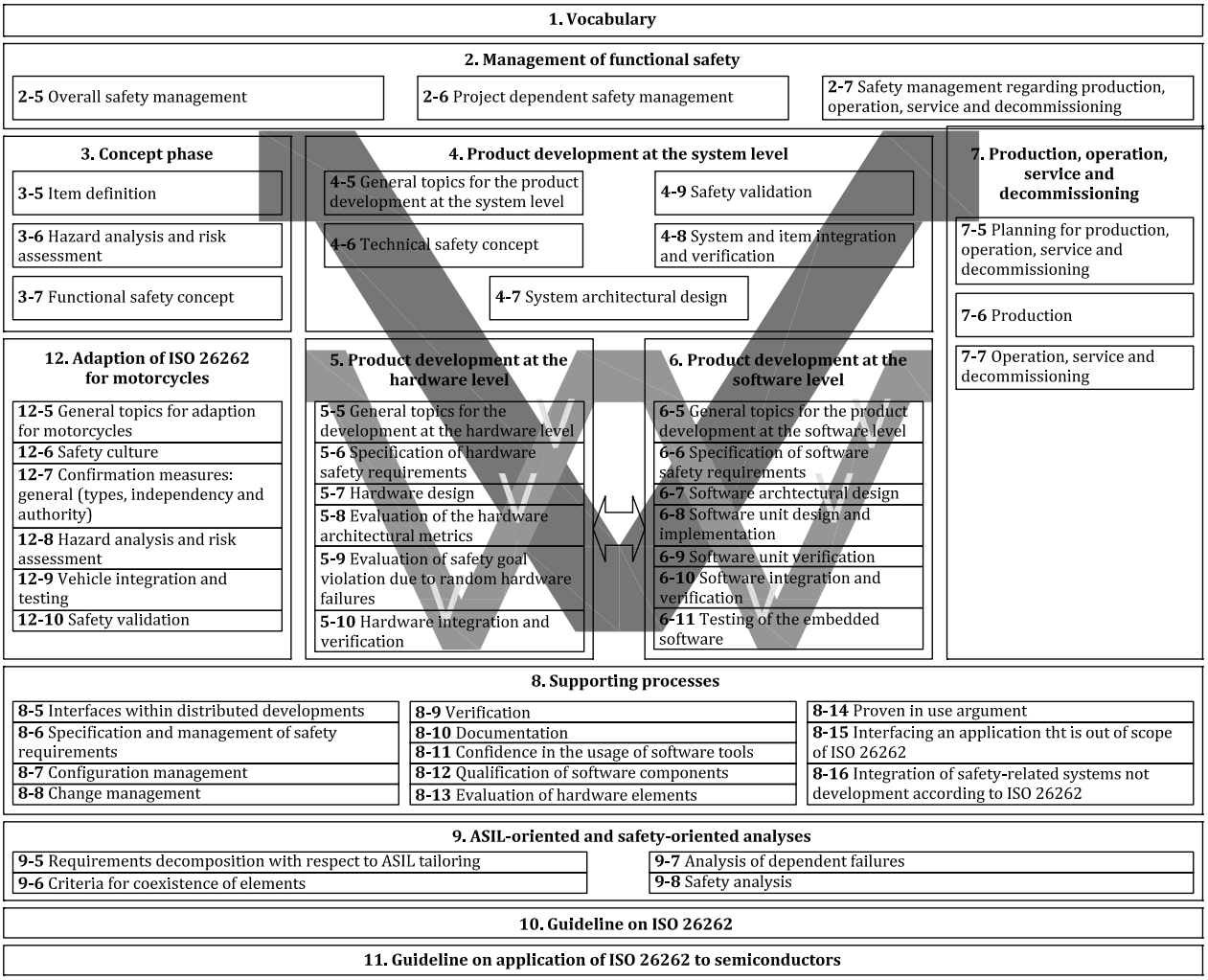


Figure 1 — Overview of the ISO 26262 series of standards

Road vehicles — Functional safety —

Part 5:
Product development at the hardware level

1 Scope

This document is intended to be applied to safety-related systems that include one or more electrical and/or electronic (E/E) systems and that are installed in series production road vehicles, excluding mopeds. This document does not address unique E/E systems in special vehicles such as E/E systems designed for drivers with disabilities.

NOTE Other dedicated application-specific safety standards exist and can complement the ISO 26262 series of standards or vice versa.

Systems and their components released for production, or systems and their components already under development prior to the publication date of this document, are exempted from the scope of this edition. This document addresses alterations to existing systems and their components released for production prior to the publication of this document by tailoring the safety lifecycle depending on the alteration. This document addresses integration of existing systems not developed according to this document and systems developed according to this document by tailoring the safety lifecycle.

This document addresses possible hazards caused by malfunctioning behaviour of safety-related E/E systems, including interaction of these systems. It does not address hazards related to electric shock, fire, smoke, heat, radiation, toxicity, flammability, reactivity, corrosion, release of energy and similar hazards, unless directly caused by malfunctioning behaviour of safety-related E/E systems.

This document describes a framework for functional safety to assist the development of safety-related E/E systems. This framework is intended to be used to integrate functional safety activities into a company-specific development framework. Some requirements have a clear technical focus to implement functional safety into a product; others address the development process and can therefore be seen as process requirements in order to demonstrate the capability of an organization with respect to functional safety.

This document does not address the nominal performance of E/E-Systems.

This document specifies the requirements for product development at the hardware level for automotive applications, including the following:

— general topics for the development at the hardware level;

— specification of hardware safety requirements;

— hardware design;

— evaluation of the hardware architectural metrics;

— evaluation of safety goal violations due to random hardware failures; and

— hardware integration and verification.

The requirements of this document for hardware elements are applicable to both, non-programmable and programmable elements, such as ASIC, FPGA and PLD. Further guidelines can be found in ISO 26262-10:2018 and ISO 26262-11:2018.

Annex A provides an overview on objectives, prerequisites and work products of the supporting processes.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 26262-1:2018, *Road vehicles — Functional safety — Part 1: Vocabulary*

ISO 26262-2:2018, *Road vehicles — Functional safety — Part 2: Management of functional safety*

ISO 26262-4:2018, *Road vehicles — Functional safety — Part 4: Product development at the system level*

ISO 26262-6:2018, *Road vehicles — Functional safety — Part 6: Product development at the software level*

ISO 26262-7:2018, *Road vehicles — Functional safety — Part 7: Production and operation*

ISO 26262-8:2018, *Road vehicles — Functional safety — Part 8: Supporting processes*

ISO 26262-9:2018, *Road vehicles — Functional safety — Part 9: Automotive Safety Integrity Level (ASIL)-oriented and safety-oriented analyses*

3 Terms and definitions

For the purposes of this document, the terms, definitions and abbreviated terms given in ISO 26262-1:2018 apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— IEC Electropedia: available at http://www.electropedia.org/

— ISO Online browsing platform: available at https://www.iso.org/obp

4 Requirements for compliance

4.1 Purpose

This clause describes how:

a) to achieve compliance with the ISO 26262 series of standards;

b) to interpret the tables used in the ISO 26262 series of standards; and

c) to interpret the applicability of each clause, depending on the relevant ASIL(s).

4.2 General requirements

When claiming compliance with the ISO 26262 series of standards, each requirement shall be met, unless one of the following applies:

a) tailoring of the safety activities in accordance with ISO 26262-2 has been performed that shows that the requirement does not apply, or

b) a rationale is available that the non-compliance is acceptable and the rationale has been evaluated in accordance with ISO 26262-2.

Informative content, including notes and examples, is only for guidance in understanding, or for clarification of the associated requirement, and shall not be interpreted as a requirement itself or as complete or exhaustive.

The results of safety activities are given as work products. “Prerequisites” are information which shall be available as work products of a previous phase. Given that certain requirements of a clause are ASIL-dependent or may be tailored, certain work products may not be needed as prerequisites.

“Further supporting information” is information that can be considered, but which in some cases is not required by ISO 26262 series of standards as a work product of a previous phase and which may be made available by external sources that are different from the persons or organizations responsible for the functional safety activities.

4.3 Interpretations of tables

Tables are normative or informative depending on their context. The different methods listed in a table contribute to the level of confidence in achieving compliance with the corresponding requirement. Each method in a table is either:

a) a consecutive entry (marked by a sequence number in the leftmost column, e.g. 1, 2, 3), or

b) an alternative entry (marked by a number followed by a letter in the leftmost column, e.g. 2a, 2b, 2c).

For consecutive entries, all listed highly recommended and recommended methods in accordance with the ASIL apply. It is allowed to substitute a highly recommended or recommended method by others not listed in the table. In this case, a rationale shall be given describing why these comply with the corresponding requirement. If a rationale can be given to comply with the corresponding requirement without choosing all entries, a further rationale for omitted methods is not necessary.

For alternative entries, an appropriate combination of methods shall be applied in accordance with the ASIL indicated, independent of whether they are listed in the table or not. If methods are listed with different degrees of recommendation for an ASIL, the methods with the higher recommendation should be preferred. A rationale shall be given that the selected combination of methods or even a selected single method complies with the corresponding requirement.

NOTE A rationale based on the methods listed in the table is sufficient. However, this does not imply a bias for or against methods not listed in the table.

For each method, the degree of recommendation to use the corresponding method depends on the ASIL and is categorized as follows:

— “++” indicates that the method is highly recommended for the identified ASIL;

— “+” indicates that the method is recommended for the identified ASIL; and

— “o” indicates that the method has no recommendation for or against its usage for the identified ASIL.

4.4 ASIL-dependent requirements and recommendations

The requirements or recommendations of each sub-clause shall be met for ASIL A, B, C and D, if not stated otherwise. These requirements and recommendations refer to the ASIL of the safety goal. If ASIL decomposition has been performed at an earlier stage of development, in accordance with ISO 26262-9:2018, Clause 5, the ASIL resulting from the decomposition shall be met.

If an ASIL is given in parentheses in the ISO 26262 series of standards, the corresponding sub-clause shall be considered as a recommendation rather than a requirement for this ASIL. This has no link with the parenthesis notation related to ASIL decomposition.

4.5 Adaptation for motorcycles

For items or elements of motorcycles for which requirements of ISO 26262-12:2018 are applicable, the requirements of ISO 26262-12:2018 supersede the corresponding requirements in this document. Requirements of ISO 26262-2:2018 that are superseded by ISO 26262-12:2018 are defined in Part 12.

4.6 Adaptation for Trucks, Buses, Trailers and Semitrailers

Content that is intended to be unique for Trucks, Buses, Trailers and Semitrailers (T&B) is indicated as such.

5 General topics for the development at the hardware level

5.1 Objectives

The objective of this clause is to describe the functional safety activities during the individual sub-phases of hardware development.

5.2 General

The necessary activities and processes needed to develop hardware that meets the safety requirements are planned according to ISO 26262-2:2018, 6.4.6.

Figure 2 illustrates the hardware level product development process steps in order to comply with the requirements of this document, and the integration of these steps within the ISO 26262 framework.

The necessary activities and processes for the product development at the hardware level include:

— the hardware implementation of the technical safety concept;

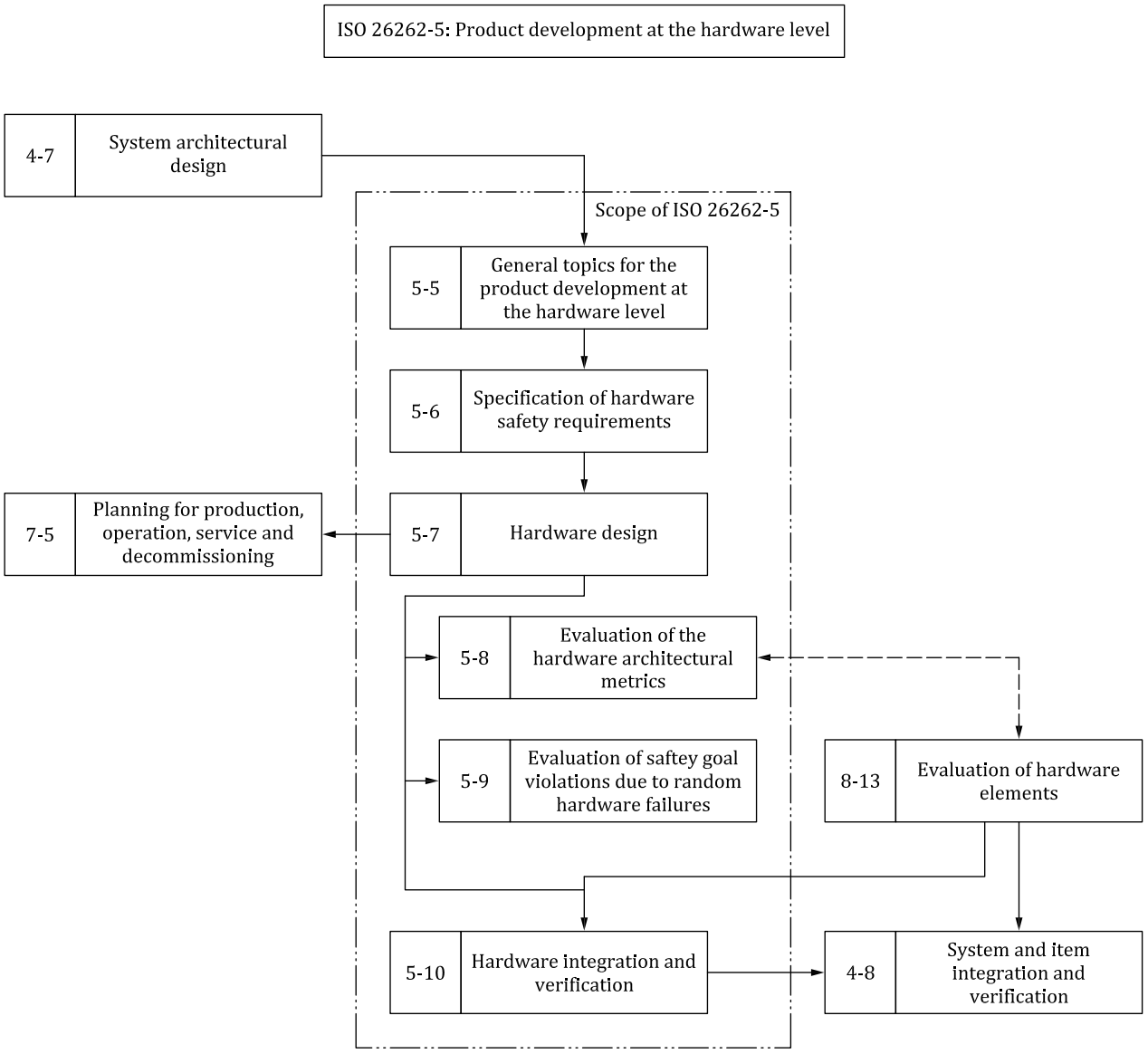
— the analysis of potential hardware faults and their effects; and

— the coordination with software development.

In contrast to the software development sub-phases, this document contains two clauses describing quantitative evaluations of the overall hardware architecture of the item.

Clause 8 describes two metrics to evaluate the effectiveness of the hardware architecture of the item and the implemented safety mechanisms to cope with random hardware failures.

As a complement to Clause 8, Clause 9 describes two alternative methods to evaluate whether the residual risk of safety goal violations is sufficiently low, either by using a global probabilistic approach (see 9.4.2, PMHF method) or by using a cut-set analysis (see 9.4.3, EEC method) to study the impact of each identified fault of a hardware element upon the violation of the safety goals.



NOTE Within the figure, the specific clauses of each part of ISO 26262 are indicated in the following manner: "m-n", where "m" represents the number of the part and "n" indicates the number of the clause, e.g. "4-7" represents ISO 26262-4:2018, Clause 7.

Figure 2 — Reference phase model for the product development at the hardware level

6 Specification of hardware safety requirements

6.1 Objectives

The objectives of this clause are:

a) to specify the hardware safety requirements. They are derived from the technical safety concept and the system architectural design specification;

b) to refine the hardware-software interface (HSI) specification initiated in ISO 26262-4:2018, 6.4.7; and

c) to verify that the hardware safety requirements and the hardware-software interface (HSI) specification are consistent with the technical safety concept and the system architectural design specification.

6.2 General

The technical safety requirements are allocated to hardware and software. The requirements that are allocated to both are further partitioned to yield hardware-only safety requirements. The hardware safety requirements are further detailed considering design constraints and the impact of these design constraints on the hardware.

6.3 Inputs to this clause

6.3.1 Prerequisites

The following information shall be available:

— technical safety concept in accordance with ISO 26262-4:2018, 6.5.2;

— system architectural design specification in accordance with ISO 26262-4:2018, 6.5.3; and

— hardware-software interface (HSI) specification in accordance with ISO 26262-4:2018, 6.5.4.

6.3.2 Further supporting information

The following information can be considered:

— software safety requirements specification (see ISO 26262-6:2018, 6.5.1); and

— hardware specifications (from an external source).

6.4 Requirements and recommendations

6.4.1 A hardware safety requirements specification for the hardware elements of the item shall be derived from the technical safety requirements allocated to hardware (resulting from ISO 26262-4:2018, 6.5.2).

6.4.2 The hardware safety requirements specification shall include each hardware requirement that relates to functional safety, including the following:

a) the hardware safety requirements and relevant properties of safety mechanisms to control internal failures of the hardware of the element. These includes internal safety mechanisms to cover transient faults when shown to be relevant due, for instance, to the technology used;

EXAMPLE 1 Properties can include the timing and detection abilities of a watchdog.

b) the hardware safety requirements and relevant properties of safety mechanisms to control or tolerate failures external to the element;

EXAMPLE 2 The functional behaviour required for an ECU in the event of an external failure, such as an open-circuit on an input of the ECU.

c) the hardware safety requirements and relevant properties of safety mechanisms to comply with the safety requirements of other elements;

EXAMPLE 3 Diagnosis of sensors or actuators.

d) the hardware safety requirements and relevant properties of safety mechanisms to detect and signal internal or external failures; and

NOTE 1 The hardware safety requirements described in bullet d) include safety mechanisms to prevent faults from being latent.

EXAMPLE 4 The specified fault reaction time interval for the hardware part of a safety mechanism, so as to be consistent with the fault tolerant time interval.

e) the hardware safety requirements not specifying safety mechanisms.

EXAMPLE 5 Examples are:

— requirements on the hardware elements to meet the target values for random hardware failures as described in 6.4.3 and 6.4.4;

— requirements for the avoidance of a specific behaviour (for instance, “a particular sensor shall not produce an unstable output signal”);

— requirements allocated to hardware elements implementing the intended functionality; and

— requirements specifying design measures on harnesses or connectors.

NOTE 2 Safety mechanisms can be implemented in hardware, in software or as a combination of both.

6.4.3 This requirement applies to ASIL (B), C, and D of the safety goal. The target values specified to comply with ISO 26262-4:2018, 6.4.5, for the metrics of Clause 8 of this document shall be considered when deriving values for the hardware elements of the item.

6.4.4 This requirement applies to ASIL (B), C, and D of the safety goal. The target values specified to comply with ISO 26262-4:2018, 6.4.5, for the procedures of Clause 9 of this document shall be considered when deriving values for the hardware elements of the item.

NOTE This activity can include an apportionment of PMHF target values in the case of a distributed development as given in ISO 26262-8:2018, Clause 5, unless the use of EEC of 9.4.3 is agreed.

6.4.5 The hardware safety requirements shall be specified in accordance with ISO 26262-8:2018, Clause 6.

6.4.6 The criteria for design verification of the hardware elements of the item shall be specified, including environmental conditions (temperature, vibration, EMI, etc.), specific operational environment (supply voltage, mission profile, etc.) and component specific requirements:

a) for verification by evaluation of hardware elements, the criteria shall meet the needs of ISO 26262-8:2018, Clause 13; and

b) for verification by testing, the criteria shall meet the needs of Clause 10 of this document.

6.4.7 The hardware safety requirements shall comply with the fault tolerant time interval, or with the maximum fault handling time interval, for safety mechanisms as specified in ISO 26262-4:2018, 6.4.2.

NOTE A mechanism able to control a fault, but not able to comply with the fault tolerant time interval or with the maximum fault handling time interval, can be specified in the HW design. In such a case, it cannot be considered within the metrics of Clause 8 and Clause 9 of this document and it cannot be considered in an ASIL decomposition scheme.

6.4.8 The hardware safety requirements shall comply with the multiple-point fault detection interval as specified in ISO 26262-4:2018, 6.4.2.

NOTE 1 In the case of ASIL C and D safety goals, and if the corresponding safety concept does not prescribe specific values, the multiple-point fault detection intervals can be specified to be equal to or lower than the item's “power-up to power-down” cycle.

NOTE 2 Appropriate multiple-point fault detection intervals can also be justified by the quantitative analysis of the occurrence of random hardware failures (see Clause 9).

6.4.9 The hardware safety requirements shall be verified in accordance with ISO 26262-8:2018, Clause 9, in order to provide evidence of their:

a) consistency with the technical safety concept, the system design specification and the hardware specifications;

b) completeness with respect to the technical safety requirements allocated to the hardware element;

c) consistency with the relevant software safety requirements; and

d) correctness and accuracy.

6.4.10 The HSI specification initiated in ISO 26262-4:2018, 6.4.7, shall be refined sufficiently to allow for the correct control and usage of the hardware by the software and shall describe each safety-related dependency between hardware and software.

6.4.11 The persons responsible for hardware and software development shall be jointly responsible for the verification of the adequacy of the refined HSI specification.

6.5 Work products

6.5.1 Hardware safety requirements specification (including test and evaluation criteria) resulting from requirements 6.4.1 to 6.4.8.

6.5.2 Hardware-software interface specification (HSI) (refined) resulting from requirement 6.4.10.

NOTE This work product refers to the same work product as given in ISO 26262-6:2018, 6.5.2.

6.5.3 Hardware safety requirements verification report resulting from requirements 6.4.9 and 6.4.11.

7 Hardware design

7.1 Objectives

The objectives of this clause are:

a) to create a hardware design that:

   — supports the safety-oriented analyses;

   — considers the results of the safety-oriented analyses;

   — fulfils the hardware safety requirements;

   — fulfils the hardware-software interface (HSI) specification;

   — is consistent with system architectural design specification; and

   — satisfies the required hardware design properties; and

b) to specify requirements and to provide information regarding functional safety of the hardware during production, operation, service and decommissioning; and

c) to verify:

— that the hardware design can fulfil the hardware safety requirements and the hardware-software interface (HSI) specification;

— the validity of the assumptions used to develop each SEooC integrated in the developed hardware; and

— the suitability of the safety-related special characteristics to achieve functional safety during production and service.

7.2 General

Hardware design includes hardware architectural design and hardware detailed design. Hardware architectural design represents all hardware components and their interactions with one another. Hardware detailed design is at the level of electrical/electronic schematics representing the interconnections between hardware parts composing the hardware components.

In order to develop a single hardware design, both hardware safety requirements as well as all non-safety requirements shall be complied with. Hence, in this sub-phase, safety and non-safety requirements are handled within one development process.

7.3 Inputs to this clause

7.3.1 Prerequisites

The following information shall be available:

— hardware safety requirements specification in accordance with 6.5.1;

— hardware-software interface specification (HSI) (refined) in accordance with 6.5.2; and

— system architectural design specification in accordance with ISO 26262-4:2018, 6.5.3.

7.3.2 Further supporting information

The following information can be considered:

— software safety requirements specification (see ISO 26262-6:2018, 6.5.1); and

— specification of non-safety-related requirements of the hardware (from an external source).

7.4 Requirements and recommendations

7.4.1 Hardware architectural design

7.4.1.1 The hardware architecture shall implement the hardware safety requirements defined in Clause 6.

7.4.1.2 The hardware safety requirements shall be allocated to the hardware elements. As a result, each hardware element shall be developed in compliance with the highest ASIL of any of the requirements allocated to it.

NOTE Each characteristic of the hardware element will inherit the highest ASIL of the hardware safety requirements that it implements.

7.4.1.3 If ASIL decomposition is applied to the hardware safety requirements during hardware architectural design, it shall be applied in accordance with ISO 26262-9:2018, Clause 5.

7.4.1.4 If a hardware element is made of sub-elements that have an ASIL lower than the ASIL of the element or no ASIL assigned, then each of these shall be treated in accordance with the highest ASIL, unless the criteria for coexistence in accordance with ISO 26262-9:2018, Clause 6 are met.

7.4.1.5 The traceability between hardware safety requirements and hardware architectural design elements shall be established down to the lowest level of hardware components.

NOTE The traceability of hardware safety requirements is not required down to the hardware detailed design. No hardware safety requirements are allocated to hardware parts that cannot be divided into sub-parts. For example, it is neither meaningful nor beneficial to try to establish hardware traceability down to each capacitor and resistor, etc.

7.4.1.6 In order to avoid systematic faults the hardware architectural design shall exhibit the following properties by use of the principles listed in Table 1:

a) modularity;

NOTE 1 Modularity enables the re-use of the design of HW elements without modification (e.g. temperature detection circuit block, ECC block in microcontroller).

b) adequate level of granularity; and

NOTE 2 The intent is that the architectural representation provides the necessary information at the necessary level of detail to show the effectiveness of the safety mechanisms.

c) simplicity.

Table 1 — Properties of hardware architectural design

Properties		ASIL			
		A	B	C	D
1	Hierarchical design	+	+	+	+
2	Precisely defined interfaces of safety-related hardware components	++	++	++	++
3	Avoidance of unnecessary complexity of interfaces	+	+	+	+
4	Avoidance of unnecessary complexity of hardware components	+	+	+	+
5	Maintainability (service)	+	+	++	++
6	Testability[a]	+	+	++	++
[a] Testability includes testability during development, production, service and operation.					

7.4.1.7 Non-functional causes for failure of a safety-related hardware component shall be considered during hardware architectural design, including the following influences, if applicable: temperature, vibrations, water, dust, EMI, noise factor, cross-talk originating either from other hardware components of the hardware architecture or from its environment.

7.4.2 Hardware detailed design

7.4.2.1 In order to avoid common design faults, relevant lessons learned shall be applied in accordance with ISO 26262-2:2018, 5.4.2.6.

7.4.2.2 Non-functional causes for failure of a safety-related hardware part shall be considered during hardware detailed design, including the following influences, if applicable: temperature, vibrations, water, dust, EMI, noise factor, cross-talk originating either from other hardware parts of the hardware component or from its environment.

7.4.2.3 The mission profile and the operating conditions of the HW part or HW component, according to the HW detailed design, shall be considered to ensure that the HW part or HW component is operated within its specification in order to avoid failures due its intended use.

7.4.2.4 Robust design principles should be considered.

NOTE Robust design principles can be shown by use of HW design guidelines.

EXAMPLE Conservative specification of components regarding their robustness against environmental and operational stress factors.

7.4.3 Safety analyses

7.4.3.1 Safety analyses of the hardware design to identify the causes of failures and the effects of faults shall be applied in accordance with Table 2 and ISO 26262-9:2018, Clause 8.

NOTE 1 The initial purpose of the safety analyses is to support the specification of the hardware design. Subsequently, the safety analyses can be used for verification of the hardware design (see 7.4.4).

NOTE 2 In order to support the specifications of the hardware design, qualitative analysis is appropriate and can be sufficient.

Table 2 — Hardware design safety analysis

Methods		ASIL			
		A	B	C	D
1	Deductive analysis	o	+	++	++
2	Inductive analysis	++	++	++	++
NOTE The level of detail of the analysis is commensurate with the level of detail of the design. Both methods can, in certain cases, be carried out at different levels of detail. EXAMPLE An FMEA is done on hardware component level and provides the basic events of an FTA conducted at a higher abstraction level.					

7.4.3.2 This requirement applies to ASIL (B), C, and D of the safety goal. For each safety-related hardware component or part, the safety analyses shall identify the following for the safety goal under consideration:

a) safe faults;

b) single-point faults or residual faults; and

c) multiple-point faults (either perceived, detected or latent).

NOTE 1 The intention of the identification of multiple-point faults is not to require a systematic analysis of every possible combination of hardware faults but, as a minimum, to consider combinations that derive from the technical safety concept (for instance the combination of two faults where one fault affects a safety-related element and another fault affects the corresponding safety mechanism intended to achieve or maintain a safe state).

NOTE 2 In most of the cases, the analysis can be limited to dual-point faults. However, multiple-point faults of a higher order than two can be shown to be relevant in the technical safety concept (e.g. when implementing redundant safety mechanisms).

7.4.3.3 This requirement applies to ASIL (A), (B), C, and D of the safety goal. Evidence of the effectiveness of implemented safety mechanisms to prevent faults from leading to single-point failures or to reduce residual faults shall be made available.

For that purpose:

a) evidence of the ability of the safety mechanisms to achieve and maintain a safe state shall be made available (in particular, appropriate failure mitigation ability within the fault tolerant time interval and within the maximum fault handling time interval); and

b) the diagnostic coverage, with respect to residual faults that is achieved by the safety mechanisms, shall be evaluated.

NOTE 1 A fault that can occur at any time (e.g. not only at power-up) cannot be considered as being effectively covered if the fault detection interval plus the fault reaction time interval, of the associated safety mechanism, is longer than the relevant fault tolerant time interval or the specified maximum fault handling time interval.

NOTE 2 If it can be demonstrated that a particular failure mode can only occur at power-up and that its probability of occurrence during the vehicle trip is negligible, then a test for those failure modes at start-up only is acceptable.

NOTE 3 An analysis such as FMEA or FTA can be used to structure the rationale.

NOTE 4 Depending on the knowledge of the failure modes of the hardware elements and their consequences at higher levels, the evaluation can be either an overall diagnostic coverage of the hardware element, or a more detailed failure mode coverage evaluation.

NOTE 5 Annex D can be used as a starting point for determining the diagnostic coverage provided by the planned safety mechanisms. The claimed DC is supported with a proper rationale (examples can be found in the clause concerning the evaluation of residual failure rate of ISO 26262-10:2018 and in ISO 26262-11:2018, Annex A).

NOTE 6 This requirement applies to safety mechanisms implemented in hardware, software or a combination of both.

7.4.3.4 This requirement applies to ASIL (A), (B), C, and D of the safety goal. Evidence of the effectiveness of implemented safety mechanisms to prevent a fault from being latent shall be made available.

For that purpose:

a) evidence of the ability of the safety mechanism to detect failures and to achieve/maintain a safe state or to notify them to the driver, within the acceptable multiple-point fault detection interval for latent faults, shall be made available in order to determine which faults will remain latent and which faults are detectable; and

b) the diagnostic coverage with respect to latent faults achieved by the safety mechanisms shall be evaluated.

NOTE 1 A fault cannot be considered covered if the fault handling time interval of the associated safety mechanism is longer than the relevant multiple-point fault detection interval for latent faults.

NOTE 2 An analysis such as FMEA or FTA can be used to structure the rationale.

NOTE 3 Depending on the knowledge of the failure modes of the hardware elements and their consequences at higher levels, the evaluation can be either an overall diagnostic coverage of the hardware element or a more detailed failure mode coverage evaluation.

NOTE 4 Annex D can be used as a starting point for determining the diagnostic coverage provided by the planned safety mechanisms. The claimed DC is supported with a proper rationale (examples can be found in the clause concerning the evaluation of residual failure rate of ISO 26262-10:2018 and in ISO 26262-11:2018, Annex A.

NOTE 5 This requirement applies to safety mechanisms implemented in hardware, software or a combination of both.

7.4.3.5 Evidence that the hardware elements in the design are compliant with their requirements for independence shall be provided, if applicable, based on an analysis of dependent failures in accordance with ISO 26262-9:2018, Clause 7.

NOTE 1 See also ISO 26262-9:2018, Annex C.

NOTE 2 See also ISO 26262-11:2018, 4.7.

7.4.3.6 If new hazards introduced by the hardware design are not already covered by an existing HARA report, they shall be introduced and evaluated in accordance with the change management process in ISO 26262-8:2018, Clause 8.

NOTE Newly identified hazards not already covered by an existing safety goal are usually non-functional hazards. Non-functional hazards are outside the scope of ISO 26262, but they can be annotated in the hazard analysis and risk assessment with the following statement: "No ASIL is assigned to this hazard as it is not within the scope of ISO 26262". However, an ASIL can be assigned for reference purposes.

7.4.4 Verification of hardware design

7.4.4.1 The hardware design shall be verified in accordance with ISO 26262-8:2018, Clause 9, and by using the hardware design verification methods listed in Table 3 to provide evidence for the following:

a) that it fulfils the hardware safety requirements;

b) that it is compatible with the hardware-software-interface specification; and

c) the suitability of the safety-related special characteristics to achieve functional safety during production and service.

Table 3 — Hardware design verification

Methods		ASIL			
		A	B	C	D
1a	Hardware design walk-through[a]	++	++	o	o
1b	Hardware design inspection[a]	+	+	++	++
2	Safety analyses	In accordance with 7.4.3			
3a	Simulation[b]	o	+	+	+
3b	Development by hardware prototyping[b]	o	+	+	+

NOTE The scope of this verification review is technical correctness and completeness with regards to the HW safety requirements.

[a] Methods 1a and 1b serve as a check of the complete and correct implementation of the hardware safety requirements in the hardware design.

[b] Methods 3a and 3b serve as a check of particular points of the hardware design (e.g. fault injection as described in ISO 26262-11:2018, 4.8) for which analytical methods 1 and 2 are not considered to be sufficient.

7.4.4.2 If it is discovered, during hardware design, that the implementation of any hardware safety requirement is not feasible, a request for change shall be issued in accordance with the change management process in ISO 26262-8:2018, Clause 8.

7.4.4.3 The validity of the assumptions used to develop a SEooC that is integrated into the hardware shall be verified against the hardware safety requirements and the hardware design specification.

7.4.5 Production, operation, service and decommissioning

7.4.5.1 Safety-related special characteristics shall be specified if safety analysis has shown them to be relevant. Specification of safety-related special characteristics shall include:

a) the verification measures for production and operation; and

b) the acceptance criteria for these measures.

EXAMPLE A safety analysis of hardware design that relies on new sensor technologies (e.g., camera or radar sensors) can reveal the relevance of special installation procedures for these sensors. In such a case, additional verification measures for these components can be necessary during the production phase.

7.4.5.2 If incorrect assembly, disassembly or decommissioning of safety-related hardware elements could have adverse effects on achieving or maintaining functional safety, then the information necessary to avoid incorrect execution shall be directed to the persons responsible for production, operation, service and decommissioning appointed in accordance with ISO 26262-2:2018, Clause 7.

7.4.5.3 The safety-related hardware elements shall be traceable, in accordance with ISO 26262-7:2018, 5.4.1.2 and 5.4.3.3 in order to:

a) enable effective field monitoring according to ISO 26262-2:2018, 7.4.2.3 and ISO 26262-7:2018, 7.4.1.1; and

b) enable recall or replacement management.

NOTE This can include adequate labelling or other identification of hardware elements to indicate that they are safety-related.

7.4.5.4 If incorrect servicing could have adverse effects on achieving or maintaining functional safety, then the information necessary to avoid such affects shall be directed to the persons responsible for production, operation, service and decommissioning appointed in accordance with ISO 26262-2:2018, Clause 7.

7.4.5.5 The requirements for production, operation, service and decommissioning of the hardware elements arising during hardware design shall be directed to the persons responsible for production, operation, service and decommissioning appointed in accordance with ISO 26262-2:2018, Clause 7.

7.5 Work products

7.5.1 Hardware design specification resulting from requirements in 7.4.1 and 7.4.2.

7.5.2 Hardware safety analysis report resulting from requirements in 7.4.3.

7.5.3 Hardware design verification report resulting from requirements in 7.4.4.

7.5.4 Specification of requirements related to production, operation, service and decommissioning resulting from requirements in 7.4.5.

8 Evaluation of the hardware architectural metrics

8.1 Objectives

The objective of this clause is to provide evidence based on the hardware architectural metrics for the suitability of the hardware architectural design of the item with respect to detection and control of safety-related random hardware failures.

8.2 General

This clause describes two hardware architectural metrics for the evaluation of the effectiveness of the architecture of the item to cope with random hardware failures.

These metrics and associated target values are evaluated on item level for the hardware elements of the item and are complementary to the evaluation of safety goal violations due to random hardware failures described in Clause 9.

The random hardware failures addressed by these metrics are limited to some of the item's safety-related electrical and electronic hardware parts, namely those that can significantly contribute to the violation or the achievement of the safety goal, and to the single-point, residual and latent faults of those parts. For electromechanical hardware parts, only the electrical failure modes and failure rates are considered.

NOTE 1 Hardware elements whose faults are multiple-point faults with a higher order than two can be omitted from the calculations unless they are shown to be relevant in the technical safety concept.

The hardware architectural metrics can be applied iteratively during the hardware architectural design and the hardware detailed design.

The hardware architectural metrics are dependent upon the whole hardware of the item. Compliance with the target figures prescribed for the hardware architectural metrics shall be achieved for each safety goal in which the item is involved.

These hardware architectural metrics are intended to achieve the following objectives:

— be objectively assessable: metrics are a comprehensible means to differentiate between different architectures;

— support evaluation of the final design (i.e. calculations based on the selected detailed hardware design);

— make available ASIL dependent pass/fail criteria to evaluate the adequacy of the hardware architecture;

— reveal whether or not the coverage by the safety mechanisms, to prevent risk from single-point or residual faults in the hardware architecture, is sufficient (single-point fault metric);

— reveal whether or not the coverage by the safety mechanisms, to prevent risk from latent faults in the hardware architecture, is sufficient (latent-fault metric);

— address single-point faults, residual faults and latent faults;

— ensure robustness concerning uncertainty of hardware failure rates;

— be limited to safety-related elements; and

— support usage on different element levels, e.g. target values can be assigned to suppliers' hardware elements.

EXAMPLE To facilitate distributed development, derived target values can be assigned to Integrated Circuits or ECUs.

NOTE 2 Items with safety-related availability requirements (i.e. the loss of a certain functionality can lead to a hazardous event) are subject to the same requirements and targets for hardware architectural metrics as items without safety related availability requirements.

8.3 Inputs of this clause

8.3.1 Prerequisites

The following information shall be available:

— hardware safety requirements specification in accordance with 6.5.1;

— hardware design specification in accordance with 7.5.1; and

— hardware safety analysis report in accordance with 7.5.2.

8.3.2 Further supporting information

The following information can be considered:

— technical safety concept (see ISO 26262-4:2018, 6.5.2); and

— system architectural design specification (see ISO 26262-4:2018, 6.5.3).

8.4 Requirements and recommendations

8.4.1 This requirement applies to ASIL (B), C, and D of the safety goal. The concepts of diagnostic coverage, single-point fault metric and latent-fault metric, in accordance with Annex C, shall apply to requirements 8.4.2 to 8.4.9.

8.4.2 This requirement applies to ASIL (B), C, and D of the safety goal. The diagnostic coverage of safety-related hardware elements by safety mechanisms shall be estimated with respect to residual faults and with respect to relevant latent faults.

NOTE 1 Annex D can be used as a starting point for determining the diagnostic coverage provided by the planned safety mechanisms. The claimed DC is supported with a proper rationale (examples can be found in clause concerning the evaluation of residual failure rate of ISO 26262-10:2018 and in ISO 26262-11:2018, Annex A.

NOTE 2 Depending on the knowledge of the failure modes of the hardware elements and their consequences at a higher level, the evaluation can be either a diagnostic coverage of the hardware element or a more detailed failure mode coverage evaluation.

8.4.3 This requirement applies to ASIL (B), C, and D of the safety goal. The estimated failure rates for hardware parts used in the analyses shall be determined:

a) using hardware part failure rate data from a recognised industry source, or

EXAMPLE 1 Commonly recognised industry sources to determine the hardware part failure rates and the failure mode distributions include SN 29500, IEC 61709, MIL HDBK 217 F notice 2, RIAC HDBK 217 Plus, UTE C80-811, NPRD-2016, EN 50129:2003, Annex C, RIAC FMD-2016, MIL HDBK 338 and FIDES 2009 EdA. The failure mode distributions e.g. those defined by “Alessandro Birolini - Reliability Engineering” can be used.

NOTE 1 The failure rate values given in these databases are generally considered to be conservative.

NOTE 2 In applying a selected industry source the following considerations are appropriate to avoid artificial reduction of the calculated base failure rate:

— the mission profile;

— the applicability of the failure modes with respect to the operating conditions; and

— the failure rate unit (per operating hour or per calendar hour).

b) using statistics based on field returns. In this case, the estimated failure rate should have a comparable confidence level of at least 70 %, or

NOTE 3 If the confidence level for the failure rates of different HW parts used in the SPFM and LFM evaluation is significantly different, the metrics will be biased.

NOTE 4 It may still be necessary to scale these statistics based data from field returns, before using them together with values from other data sources with different confidence levels. See also Note 7.

NOTE 5 Failure rates based on field returns can be calculated as described in ISO 26262-8:2018, Clause 14 (Proven in use).

c) using expert judgement founded on an engineering approach based on quantitative and qualitative arguments. Expert judgement shall be exercised in accordance with structured criteria as a basis for this judgement. These criteria shall be set before the estimation of failure rates is made.

NOTE 6 The criteria for expert judgment can include a combination of heuristic information supported by a combination of field data, testing, reliability analysis and physics-of-failure based simulation approaches while considering the novelty of the design.

NOTE 7 Informative references from international reliability expert bodies can be used: SAE J1211 "Robustness Validation" – Analysis, Modelling and Simulation provides physics-of-failure (PoF) based failure mechanism models, JEDEC-JESD89, JEDEC-JESD91, JEDEC-JESD94, JEDEC-JEP143, JEDEC-JESD148.

NOTE 8 If failure rates from multiple data sources (as listed in 8.4.3) are combined, e.g. in the case the failure rates of different parts are not available from the same source, the failure rates can be scaled using a scaling factor such that the quality of prediction of the different failures rates is equivalent. This scaling can be used if a rationale for the scaling factor between two failure rate sources is available.

EXAMPLE 2 An element failure rate is found only in one source whereas a similar element is available in that and another source. The scaling factor is the ratio of the failure rate from these two sources utilising the same mission profile.

EXAMPLE 3 Failure rates from data handbooks are generally considered to be conservative. If a random hardware failure target value consistent with the use of handbook data is chosen, failure rate derived from field returns can be used by applying an appropriate scaling factor (corresponding to a confidence level more conservative than usual, for example).

NOTE 9 If a suitable scaling factor is not available, separate target values compliant with the SPFM and LFM requirements can be assigned to the different elements under consideration (analoguous to 8.4.4).

NOTE 10 For semiconductors, see ISO 26262-11:2018, 4.6 for details.

8.4.4 This requirement applies to ASIL (B), C, and D of the safety goal. If the evidence that can be made available for the hardware part failure rate is insufficient, then alternative means shall be proposed (e.g. add safety mechanisms to detect and control this fault). If the alternative means consist solely of added safety mechanisms:

a) the diagnostic coverage of the hardware part with respect to residual faults shall be equal to or higher than the item SPFM target value; and

b) the diagnostic coverage of the hardware part with respect to latent faults shall be equal to or higher than the item LFM target value.

NOTE 1 Sufficient evidence means, for instance, that evidence is given that the failure rate has been determined using one of the methods listed in 8.4.3.

NOTE 2 The proportion of safe faults of the hardware part can be considered when determining the coverage of the safety mechanisms. In this case, the calculation of the coverage is done in a similar way to the calculation of the single point fault metric or the latent fault metric, at the hardware part level instead of at the item level.

8.4.5 This requirement applies to ASIL (B), C, and D of the safety goal. For each safety goal, a quantitative target value for the "single-point fault metric" as required in ISO 26262-4:2018, 6.4.5, shall be based on one of the following sources of reference target values:

a) derived from the hardware architectural metric calculation applied on a similar well-trusted design, or

NOTE 1 Two similar designs have similar functionalities and similar safety goals with the same assigned ASIL.

b) derived from Table 4.

Table 4 — Possible source for the derivation of the target "single-point fault metric" value

	ASIL B	ASIL C	ASIL D
Single-point fault metric	≥90 %	≥97 %	≥99 %

NOTE 2 This quantitative target is intended to provide:

— design guidance; and

— evidence that the design complies with the safety goals.

8.4.6 This requirement applies to ASIL (B), (C), and D of the safety goal. For each safety goal, a quantitative target value for "latent-fault metric" as required in ISO 26262-4:2018, 6.4.5 shall be based on one of the following sources of reference target values:

a) derived from the hardware architectural metrics calculation applied on similar well-trusted design; or

NOTE 1 Two similar designs have similar functionalities and similar safety goals with the same assigned ASIL.

b) derived from Table 5.

Table 5 — Possible source for the derivation of the target "latent-fault metric" value

	ASIL B	ASIL C	ASIL D
Latent-fault metric	≥60 %	≥80 %	≥90 %

NOTE 2 This quantitative target is intended to provide:

— design guidance; and

— evidence that the design complies with the safety goals.

8.4.7 This requirement applies to ASIL (B), C, and D of the safety goal. For each safety goal, the whole hardware of the item shall comply with one of the following alternatives:

a) to meet the target "single-point fault metric" value, as described in 8.4.5, or

b) to meet the appropriate targets prescribed at the hardware element level which are sufficient to comply with the single-point fault metric's target value assigned to the whole hardware of the item as described in requirement 8.4.5 and to provide the rationale for compliance with these targets at the hardware element level.

NOTE 1 If an item contains different kinds of hardware elements with significantly different failure rate levels, the risk exists that compliance with the hardware architectural metrics only focuses on the kind of hardware elements with the highest magnitude of failure rates. One example where this can occur is for the single-point fault metric for which compliance can be achieved by considering the failure rates for failures of wires / fuses / connectors, while disregarding the failure rates of hardware parts with significantly lower failure rates. The prescription of appropriate metric target values for each kind of hardware helps to avoid this side effect.

NOTE 2 Transient faults are considered when shown to be relevant due, for instance, to the technology used. They can be addressed either by specifying and verifying a dedicated target "single-point fault metric" value to them (as explained in NOTE 1) or by a qualitative rationale based on the verification of the effectiveness of the internal safety mechanisms implemented to cover these transient faults.

NOTE 3 If the target is not met, the rationale for how the safety goal is achieved will be assessed as given in 4.2.

NOTE 4 Some or all of the applicable safety goals can be considered together for the determination of the single-point fault metric; but in this case the metric's target to be considered is that of the safety goal with the highest ASIL.

EXAMPLE If an item consists out of three hardware elements A, B and C with a failure rate λ_A, λ_B, λ_C and $\lambda_{total} = \lambda_A + \lambda_B + \lambda_C$, any combination of element SPFM target values $M_{SPFM,\,Element}$ fulfilling the following equation is acceptable: $\left(\frac{\lambda_A}{\lambda_{total}} \times M_{SPFM,A} + \frac{\lambda_B}{\lambda_{total}} \times M_{SPFM,B} + \frac{\lambda_C}{\lambda_{total}} \times M_{SPFM,C} \right) \geq M_{SPFM,Itemtarget}$.

8.4.8 This requirement applies to ASIL (B), (C), and D of the safety goal. For each safety goal, the whole hardware of the item shall comply with one of the following alternatives:

a) to meet the target "latent-fault metric" value, as described in 8.4.6;

b) to meet the appropriate targets prescribed at the hardware element level which are sufficient to comply with the latent-fault metric's target value assigned to the whole hardware of the item as described in requirement 8.4.6 and to provide the rationale for compliance with these targets at the hardware element level; or

c) to meet the target values for the diagnostic coverage, with respect to latent faults, identical to the target value given in reference 8.4.6 for the latent-fault metric (treated as a diagnostic coverage), for each hardware element with faults that can lead to the unavailability of a safety mechanism (to prevent a fault from violating the safety goal). This alternative applies when each safety mechanism, whose unavailability can contribute to the violation of the safety goal, is based on fault detection.

NOTE 1 Alternative c) is limited to the cases where each relevant safety mechanism is based on fault detection. It is supposed that in this case the potentially latent faults of the intended functionality are alerted through the detection of these safety mechanisms. In other cases, this alternative cannot be applied and alternatives a) and b) are the only possibilities.

EXAMPLE 1 Annex H provides examples of different types of safety mechanisms considering latent fault handling.

NOTE 2 In the case of c), a metric is not calculated. Only the coverage of the hardware elements by safety mechanisms with respect to latent faults is evaluated.

NOTE 3 If an item contains different kinds of hardware elements with significantly different failure rate levels, the risk exists that compliance with the hardware architectural metrics only focuses on the kind of hardware elements with the highest magnitude of failure rates. One example where this can occur is for the latent fault metric for which compliance could be achieved by considering the failure rates for failures of wires / fuses / connectors, while disregarding the failure rates of hardware parts with significantly lower failure rates. The prescription of appropriate metric target values for each kind of hardware helps to avoid this side effect.

NOTE 4 If the target is not met, the rationale for how the safety goal is achieved will be assessed as given in 4.2.

NOTE 5 Some or all of the applicable safety goals can be considered together for the determination of the latent-fault metric; but in this case the metric's target to be considered is that of the safety goal with the highest ASIL.

EXAMPLE 2 If an item consists out of three hardware elements A, B and C with a failure rate λ_A, λ_B, λ_C and $\lambda_{total} = \lambda_A + \lambda_B + \lambda_C$, any combination of element LFM target values $M_{LFM,\ Element}$ fulfilling the following equation is acceptable: $\left(\frac{\lambda_A}{\lambda_{total}} \times M_{LFM,A} + \frac{\lambda_B}{\lambda_{total}} \times M_{LFM,B} + \frac{\lambda_C}{\lambda_{total}} \times M_{LFM,C} \right) \geq M_{LFM,Itemtarget}$.

8.4.9 This requirement applies to ASIL (B), C, and D of the safety goal. A verification review of the result of the applied methods in 8.4.7 and 8.4.8 shall be performed in order to provide evidence of its technical correctness and completeness in accordance with ISO 26262-8:2018, Clause 9.

NOTE Verification of the single-point fault metric ensures that only failure rates of safety-related hardware elements are taken into consideration, so that the metric is not inappropriately skewed by unnecessary safety-related hardware elements without the potential for having single-point faults or residual faults (e.g. by adding unnecessary hardware elements to a safety mechanism).

8.5 Work products

8.5.1 Analysis of the effectiveness of the architecture of the item to cope with the random hardware failures resulting from requirements 8.4.1 to 8.4.8.

8.5.2 Verification review report of evaluation of the effectiveness of the architecture of the item to cope with the random hardware failures resulting from requirement 8.4.9.

9 Evaluation of safety goal violations due to random hardware failures

9.1 Objectives

The objective of this clause is to provide evidence that the residual risk of a safety goal violation, due to random hardware failures of the item, is sufficiently low.

NOTE "Sufficiently low" means "comparable to residual risks on items already in use and known to be safe".

9.2 General

Two alternative methods (see 9.4) are proposed to evaluate whether the residual risk of safety goal violations is sufficiently low.

Both methods evaluate the residual risk of violating a safety goal due to single-point faults, residual faults, and plausible dual-point faults. Multiple-point faults can also be considered if shown to be relevant to the safety concept. In this analysis, coverage of safety mechanisms will be considered for residual and dual-point faults, and exposure duration will be considered as well for dual-point faults.

The requirements for the first method are given in 9.4.2. The "Probabilistic Metric for random Hardware Failures" (PMHF) represents a quantitative analysis which evaluates the violation of the considered safety goal by random failures of the hardware elements. The quantitative result of the analysis is compared with a target value.

The requirements for the second method are given in 9.4.3. The "Evaluation of Each Cause of safety goal violation" (EEC) is based on the individual evaluation of each hardware part and its contribution to the violation of the considered safety goal with respect to residual, single-point and plausible dual-point failures.

The chosen method can be applied iteratively during the hardware architectural design and the hardware detailed design.

The scope of this clause is limited to the random hardware failures of the item. The parts considered in the analyses are the electrical and electronic hardware parts. For electromechanical hardware parts, only the electrical failure modes and failure rates are considered.

9.3 Inputs to this clause

9.3.1 Prerequisites

The following information shall be available:

— hardware safety requirements specification in accordance with 6.5.1;

— hardware design specification in accordance with 7.5.1; and

— hardware safety analysis report in accordance with 7.5.2.

9.3.2 Further supporting information

The following information can be considered:

— technical safety concept (see ISO 26262-4:2018, 6.5.2); and

— system architectural design specification (see ISO 26262-4:2018, 6.5.3).

9.4 Requirements and recommendations

9.4.1 General

9.4.1.1 This requirement applies to ASIL (B), C and D of the safety goal. The item shall comply with either 9.4.2 or 9.4.3.

9.4.1.2 This requirement applies to ASIL C and D of the safety goal. A hardware part single-point fault shall only be considered acceptable if an argument for its sufficiently low probability of occurrence is provided by one of the following options:

a) dedicated measures are taken, or

b) for a safety goal ASIL D the following criteria are satisfied:

   — a conservative data source is used;

   — only a small portion of the failure rate (e.g. one particular failure mode) can violate the safety goal; and

   — the resulting single-point failure rate is smaller than one tenth of the value corresponding to failure rate class 1 (according to 9.4.3.3);

c) for a safety goal ASIL C the following criteria are satisfied:

   — a conservative data source is used;

   — only a small portion of the failure rate (e.g. one particular failure mode) can violate the safety goal; and

   — the resulting single-point failure rate is smaller than one tenth of the value corresponding to failure rate class 2 (according to 9.4.3.3).

NOTE 1 Regarding this requirement, a microcontroller, an ASIC or similar SoC can be treated as hardware parts.

NOTE 2 Dedicated measures can include:

a) design features such as hardware part over-design (e.g. electrical or thermal stress rating) or physical separation (e.g. spacing of contacts on a printed circuit board);

b) a special sample test of incoming material to reduce the risk of occurrence of this failure mode;

c) a burn-in test;

d) a dedicated control set as part of the control plan; and

e) assignment of safety-related special characteristics.

9.4.1.3 This requirement applies to ASIL C and D of the safety goal. A hardware part residual fault, with a diagnostic coverage (with respect to residual faults) lower than 90 %, shall only be considered acceptable if an argument for its sufficiently low probability of occurrence is provided by one of the following options:

a) dedicated measures are taken (the NOTE 2 in 9.4.1.2 lists examples of dedicated measures), or

b) for a safety goal ASIL D the following criteria are satisfied:

— a conservative data source is used;

— only a small portion of the failure rate (e.g. one particular failure mode) can violate the safety goal; and

— the resulting residual failure rate is smaller than one tenth of the value corresponding to failure rate class 1 (according to 9.4.3.3);

c) for a safety goal ASIL C the following criteria are satisfied:

— a conservative data source is used;

— only a small portion of the failure rate (e.g. one particular failure mode) can violate the safety goal; and

— the resulting residual failure rate is smaller than one tenth of the value corresponding to failure rate class 2 (according to 9.4.3.3).

NOTE 1 Regarding this requirement, a microcontroller, an ASIC or similar SoC can be treated as hardware parts.

NOTE 2 The proportion of safe faults of the hardware part can be considered when determining the coverage of the safety mechanisms. In this case the calculation of the coverage is done analogously to the calculation of the single-point fault metric, but at the hardware part level instead of at the item level.

9.4.1.4 This requirement applies to ASIL (B), C, and D of the safety goal. The failure rates for hardware parts used in the analyses shall be estimated in accordance with 8.4.3.

9.4.2 Evaluation of Probabilistic Metric for random Hardware Failures (PMHF)

9.4.2.1 This requirement applies to ASIL (B), C, and D of the safety goal. Quantitative target values of requirements 9.4.2.2 or 9.4.2.3 shall be expressed in terms of average probability per hour over the operational lifetime of the item.

NOTE 1 Failure rate and average probability of failure per hour over the operational lifetime of the item are different values even if they share the same unit.

NOTE 2 Operational lifetime only includes the operating hours.

9.4.2.2 This requirement applies to ASIL (B), C, and D of the safety goal. Quantitative target values for the maximum probability of the violation of each safety goal at item level due to random hardware failures as required in ISO 26262-4:2018, 6.4.5, shall be defined using one of the sources a), b) or c) of reference target values, as outlined below:

a) derived from Table 6,

b) derived from field data from a similar well-trusted design, or

c) derived from quantitative analysis techniques applied to a similar well-trusted design using failure rates in accordance with 8.4.3.

NOTE 1 These quantitative target values derived from sources a), b) or c) do not have an absolute significance but are useful for comparing a new design with existing ones. They are intended to make available design guidance as described in 9.1 and to make available evidence that the design complies with the safety goals.

NOTE 2 Two similar designs have similar functionalities and similar safety goals with the same assigned ASIL.

NOTE 3 Table 6 is typically chosen when no other source is available to determine the random hardware failure target value.

NOTE 4 The values in Table 6 are intended for items composed of a single system (e.g. Engine Management System, Electronic Stability Control System, Electric Power Assisted Steering System, Airbag Restraint System).

NOTE 5 The target values given in Table 6 are consistent with the use of handbook data, the latter are recognised as being conservative. If the evaluation of safety goal violations due to random hardware failures is done based on statistical data (e.g. from the field), the target values given in Table 6 can be adapted to avoid an artificial simplification in achieving the target values.

Table 6 — Possible source for the derivation of the random hardware failure target values

ASIL	Random hardware failure target values
D	$<10^{-8}$ h^{-1}
C	$<10^{-7}$ h^{-1}
B	$<10^{-7}$ h^{-1}
NOTE The quantitative target values described in this table can be tailored as specified in 4.2 to fit specific uses of the item (e.g. if the item is able to violate the safety goal for durations longer than the typical use of a passenger car).	

9.4.2.3 This requirement applies to ASIL (B), C and D of the safety goal. When an item consists of several systems, the derived target value of requirement 9.4.2.2 may be directly allocated to each system composing the item. This can be applied, as long as each of these systems has the potential to violate the same safety goal and the corresponding item target value is not increased by more than one order of magnitude.

NOTE 1 The possibility described in requirement 9.4.2.3 can, for example, be used for legacy systems, that are involved in a new higher level functionality (e.g. new ADAS using Engine Management System, Electronic Stability Control System, Electric Power Assisted Steering System or Airbag Restraint System), and that had achieved the same safety goal in previous developments.

EXAMPLE If an ASIL D safety goal is allocated to an item comprised of several systems (up to ten), each of which has the potential to violate that safety goal, the target value of 10^{-8}/h can be allocated to each system.

NOTE 2 An example of a PMHF budget assignment, for an item consisting of two systems, is given in Annex G.

9.4.2.4 This requirement applies to ASIL (B), C, and D of the safety goal. A quantitative analysis of the hardware architecture with respect to the single-point, residual and multiple-point faults shall provide evidence that target values of requirements 9.4.2.2 or 9.4.2.3 have been achieved. This quantitative analysis shall consider:

a) the architecture of the item;

b) the estimated failure rate for the failure modes of each hardware part that would cause a single-point fault or a residual fault;

c) the estimated failure rate for the failure modes of each hardware part that would cause a multiple-point fault;

d) the diagnostic coverage of safety-related hardware elements by safety mechanisms; and

e) the exposure duration in the case of multiple-point faults.

NOTE 1 Failure modes of hardware elements that can cause a failure of a safety-related hardware element and its safety mechanism simultaneously are considered in the quantitative analysis. They can be single-point faults, residual faults or multiple-point faults.

NOTE 2 Exposure duration starts as soon as the fault can occur and includes:

— the multiple-point fault detection interval associated with each safety mechanism, or the lifetime of the vehicle if the fault is not indicated to the driver (latent fault);

— the maximum duration of a trip (in the case where the driver is requested to stop in a safe manner); and

— the average time interval until the vehicle is at the workshop for repair (in the case where the driver is alerted to have the vehicle repaired).

Therefore, exposure duration depends on the type of monitoring involved (e.g. continuous monitoring, periodic self-tests, driver monitoring, no monitoring) and the kind of reaction when the fault has been detected. It can be as short as a few milliseconds in the case of a continuous monitoring triggering a transition to a safe state. It can be as long as the car lifetime when there is no monitoring.

Example of assumptions on the average time to vehicle repair, depending on the fault type:

— 200 vehicle trips for reduction of comfort features;

— 50 vehicle trips for reduction of driving support features;

— 20 vehicle trips for amber warning lights or impacts on driving behaviour; and

— one vehicle trip for red warning lights.

The time taken for repair is usually not considered (except to evaluate hazards that can expose maintenance personnel).

The mean duration of a vehicle trip can be considered as being equal to:

— 1 h for passenger cars; and

— 10 h for Trucks and Buses.

NOTE 3 In most cases, multiple-point failures of a higher order than two have a negligible contribution with respect to the quantitative target values. However, in some particular cases (very high failure rate or poor diagnostic coverage), it can be necessary to provide two redundant safety mechanisms to reach the target. When the technical safety concept is based on redundant safety mechanisms, multiple-point failures of a higher order than two are considered in the analysis.

NOTE 4 Annex D can be used as a starting point for determining the diagnostic coverage provided by the planned safety mechanisms. The claimed DC is supported with a proper rationale (examples can be found in clause concerning the evaluation of residual failure rate of ISO 26262-10:2018 and in ISO 26262-11:2018, Annex A).

NOTE 5 As pointed out in 9.4.2.2 NOTE 1, the PMHF values do not have an absolute significance but are useful for comparing a new design with existing ones.

NOTE 6 If the target value defined in 9.4.2.2 or 9.4.2.3 is not met, the rationale for how the safety goal is achieved will be assessed as given in 4.2. This rationale can be based on:

— the identification of the top contributors to the PMHF value and the failure modes which have a low coverage; and

— the review of these contributors, considering amongst other criteria the failure rate, reliability investigations, diagnostic coverage, failure mode coverage, field experience, verification measures, state of the art and dedicated measures (the NOTE 2 in 9.4.1.2 lists examples of dedicated measures).

An example of such rationale is given in Annex F.

NOTE 7 Depending on the knowledge of the failure modes of the hardware elements and their consequences at a higher level, the evaluation can be either a diagnostic coverage of the hardware element, or a more detailed failure mode coverage evaluation.

NOTE 8 Due to the uncertainties in the derivation of the PMHF value of the item (e.g. derivation of the failure rate, failure modes, failure mode distribution, diagnostic coverage, ratio of safe faults) the calculated value can vary significantly and special care is taken when interpreting it.

9.4.3 Evaluation of Each Cause of safety goal violation (EEC)

9.4.3.1 A method for evaluation of each cause of a safety goal violation due to random hardware failures is illustrated by flowcharts in Figure 3 and in Figure 4. Each single-point fault is evaluated using criteria on the occurrence of the fault. Each residual fault is evaluated using criteria combining the occurrence of the fault and the efficiency of the safety mechanism.

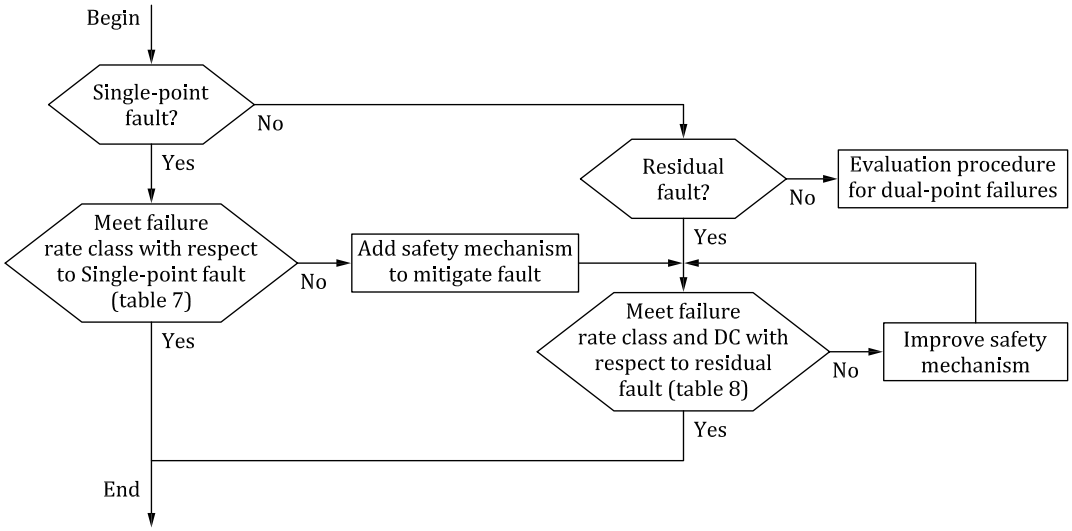


Figure 3 — Evaluation procedure for single-point and residual faults

The procedure to be applied for dual-point failures is illustrated by the flowchart in Figure 4. Each dual-point failure is first evaluated regarding its plausibility. A dual-point failure is considered implausible if both faults leading to the failure are detected or perceived in a sufficiently short time with sufficient coverage. If the dual-point failure is plausible, the faults causing it are then evaluated using criteria combining occurrence of the fault and coverage of the safety mechanisms.

If a fault evaluation fails to comply with the criterion combining occurrence of the fault and coverage of the safety mechanisms, the corresponding dual point failure can then be evaluated against a criterion of occurrence.

The evaluation procedures described in Figure 3 and 4 apply to the hardware parts (resistors, capacitors, CPUs, etc.) level.

NOTE The probability of occurrence of the dual point failure is evaluated by a quantitative analysis technique (e.g. FTA, Markov analysis), according to the description given in 9.4.2.4.

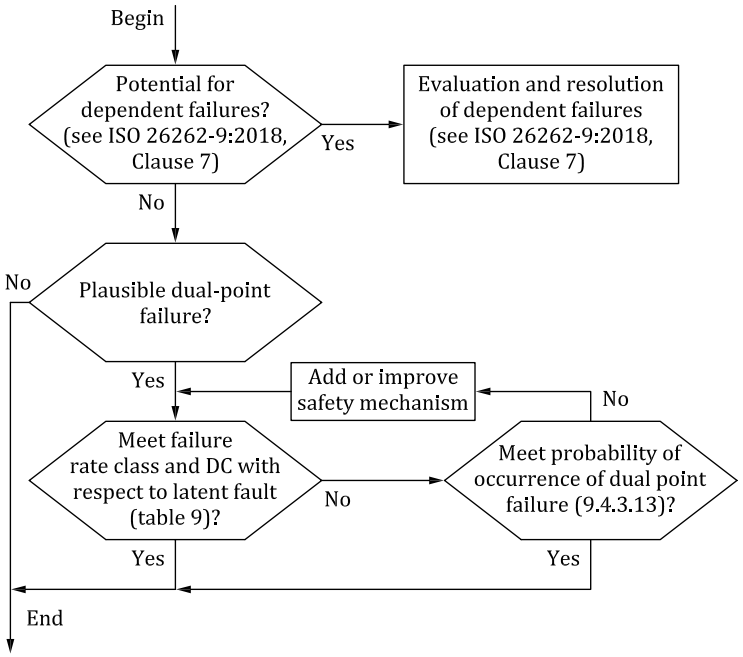


Figure 4 — Evaluation procedure for dual-point failures

9.4.3.2 This requirement applies to ASIL (B), C, and D of the safety goal. An individual evaluation of each single-point fault, residual fault and dual-point failure violating the considered safety goal shall be performed at the hardware part level. This evaluation shall provide evidence that each single-point fault, residual fault and dual-point failure violating the considered safety goal is acceptable in accordance with requirements 9.4.3.3 to 9.4.3.13.

NOTE 1 This analysis can be viewed as a review of cut sets where absence or incompleteness of coverage is treated as a fault.

NOTE 2 In most cases, multiple-point failures of a higher order than two are negligible. However, in some particular cases (very high failure rate or poor diagnostic coverage), it can be necessary to provide two redundant safety mechanisms. Therefore, it is necessary to consider multiple-point failures of a higher order than two in the analysis when the technical safety concept is based on redundant safety mechanisms.

NOTE 3 When analysis is performed at subsystem level, this analysis can consider system safety mechanisms implemented in other subsystems.

NOTE 4 If the evidence is not provided that a single-point fault, a residual fault or a dual-point failure is acceptable in accordance with requirements 9.4.3.3 to 9.4.3.13, the rationale for how the safety goal is achieved will be assessed as given in 4.2. This rationale can be based on the review of these faults/failures, considering amongst other criteria the failure rate, reliability investigations, diagnostic coverage, failure mode coverage, field experience, verification measures, state of the art and dedicated measures (the NOTE 2 in 9.4.1.2 lists examples of dedicated measures).

An example of such rationale is given in Annex F.

9.4.3.3 This requirement applies to ASIL (B), C, and D of the safety goal. The failure rate class ranking for a hardware part failure rate shall be determined as follows:

NOTE 1 The failure rate classes 1, 2 and 3 are introduced to address the failure occurrence rates. These classes are analogous to the occurrence levels 1, 2 and 3, respectively, used in an FMEA, where a 1 is assigned to failure modes which have the lowest occurrence rate.

a) the failure rate corresponding to failure rate class 1 shall be less than the target for ASIL D divided by 100; unless 9.4.3.4 is applied;

NOTE 2 The target values given in Table 6 can be used.

b) the failure rate corresponding to failure rate class 2 shall be less than or equal to 10 times the failure rate corresponding to failure rate class 1;

c) the failure rate corresponding to failure rate class 3 shall be less than or equal to 100 times the failure rate corresponding to failure rate class 1; and

d) the failure rate corresponding to failure rate class i, i > 3 shall be less than or equal to $10^{(i-1)}$ times the failure rate corresponding to failure rate class 1.

NOTE 3 The failure rate class assignment is based upon the hardware part failure rate not considering the effectiveness of the safety mechanisms.

NOTE 4 For the case where a small number of parts (such as inside a semiconductor component) have failure rates higher than the failure rate class i upper limit, then these parts can be assigned a class i occurrence if the resulting average failure rate of parts assigned class i is lower than failure rate class i's upper limit.

9.4.3.4 This requirement applies to ASIL (B), C, and D of the safety goal. The failure rate class ranking may be divided by a number different than 100, for which a rationale is provided. In this case, it shall be ensured that a correct ranking is maintained considering the single-point faults, residual faults and higher degree cut-sets.

EXAMPLE The rationale can be based on the number of minimal cut-sets or the number of safety-related hardware elements.

9.4.3.5 This requirement applies to ASIL (B), C and D of the safety goal. A single-point fault occurring in a hardware part shall only be considered as acceptable if the corresponding hardware part failure rate class ranking complies with the targets given in Table 7.

Table 7 — Targets of failure rate classes of hardware parts regarding single-point faults

ASIL of the safety goal	Failure rate class
D	Failure rate class 1 + dedicated measures[a]
C	Failure rate class 2 + dedicated measures[a] or Failure rate class 1
B	Failure rate class 2 or Failure rate class 1

[a] The NOTE 2 in requirement 9.4.1.2 gives examples of dedicated measures.

NOTE When assessing the failure rate class, the proportion of safe faults of the hardware part can be considered.

9.4.3.6 This requirement applies to ASIL (B), C, and D of the safety goal. A residual fault occurring in a hardware part shall be considered acceptable if the failure rate class ranking complies with the targets given in Table 8 for the diagnostic coverage (with respect to residual faults) of the corresponding hardware part.

NOTE 1 The considered failure rate is the hardware part failure rate not considering the effectiveness of the safety mechanisms.

Table 8 — Maximum failure rate classes for a given diagnostic coverage of the hardware part – residual faults

ASIL of the safety goal	Diagnostic coverage with respect to residual faults			
	≥99,9 %	≥99 %	≥90 %	<90 %
D	Failure rate class 4	Failure rate class 3	Failure rate class 2	Failure rate class 1 + dedicated measures[a]
C	Failure rate class 5	Failure rate class 4	Failure rate class 3	Failure rate class 2 + dedicated measures[a]
B	Failure rate class 5	Failure rate class 4	Failure rate class 3	Failure rate class 2

[a] The NOTE 2 in requirement 9.4.1.2 gives examples of dedicated measures.

NOTE 2 Table 8 specifies the connection between maximum failure rate class allowed given the target ASIL and the diagnostic coverage. Lower failure rate classes are acceptable but not required.

NOTE 3 "Lower failure rate classes" means failure rate classes with a lower number. For example, "lower failure rate classes" with respect to failure rate class 3 means failure rate classes 2 and 1.

NOTE 4 The proportion of safe faults of the hardware part can be considered when determining the coverage of the safety mechanisms. In this case, the calculation of the coverage is done analogously to the calculation of the single-point fault metric, but at the hardware part level instead of at the item level.

9.4.3.7 This requirement applies for ASIL (B), C and D of the safety goal. For failure rate classes i, $i > 3$, a residual fault shall be considered as acceptable if the diagnostic coverage is greater than or equal to $[100 - 10^{(3-i)}]$ % for ASIL D or greater than or equal to $[100 - 10^{(4-i)}]$ % for ASIL B and C.

NOTE 1 The considered failure rate is the hardware part failure rate, and does not take into account the effectiveness of the safety mechanisms.

NOTE 2 The proportion of safe faults of the hardware part can be considered when determining the coverage of the safety mechanisms. In this case the calculation of the coverage is done analogously to the calculation of the single-point fault metric, but at the hardware part level instead of at the item level.

9.4.3.8 This requirement applies to ASIL D of the safety goal. A dual-point failure shall be considered plausible if:

a) at least one of both hardware parts involved has a diagnostic coverage (with respect to the latent faults) of less than 90 %; or

b) one of the dual-point faults causing the dual-point failure remains latent for a time longer than the multiple-point fault detection interval as specified in requirement 6.4.8.

NOTE The proportion of safe faults of the hardware part can be considered when determining the coverage of the safety mechanisms. In this case the calculation of the coverage is done analogously to the calculation of the latent fault metric, but at the hardware part level instead of at the item level.

9.4.3.9 This requirement applies to ASIL C of the safety goal. A dual-point failure shall be considered plausible if:

a) at least one of both hardware parts involved has a diagnostic coverage (with respect to the latent faults) of less than 80 %; or

b) one of the dual-point faults causing the dual-point failure remains latent for a time longer than the multiple-point fault detection interval as specified in requirement 6.4.8.

NOTE The proportion of safe faults of the hardware part can be considered when determining the coverage of the safety mechanisms. In this case the calculation of the coverage is done analogously to the calculation of the latent fault metric, but at the hardware part level instead of at the item level.

9.4.3.10 This requirement applies to ASIL C and D of the safety goal. A dual-point failure that is implausible shall be considered compatible with the safety goal target and thus acceptable.

9.4.3.11 This requirement applies to ASIL C and D of the safety goal. A dual-point fault occurring in a hardware part and contributing to a plausible dual-point failure shall be considered acceptable if the corresponding hardware part complies with the targets for the failure rate class ranking and diagnostic coverage (with respect to latent faults) given in Table 9.

NOTE 1 The considered failure rate is the hardware part failure rate. Therefore, it does not consider the effectiveness of safety mechanisms.

Table 9 — Targets of failure rate class and coverage of hardware part regarding dual-point faults

ASIL of safety goal	Diagnostic coverage with respect to latent faults		
	≥99 %	≥90 %	<90 %
D	Failure rate class 4	Failure rate class 3	Failure rate class 2
C	Failure rate class 5	Failure rate class 4	Failure rate class 3

NOTE 2 Table 9 specifies the maximum failure rate class allowed given the target ASIL and the level of diagnostic coverage achieved. Lower failure rate classes are acceptable but not required.

NOTE 3 “Lower failure rate classes” means failure rate classes with a lower number. For example, “lower failure rate classes” with respect to failure rate class 3 means failure rate classes 2 and 1.

NOTE 4 The proportion of safe faults of the hardware part can be considered when determining the coverage of the safety mechanisms. In this case the calculation of the coverage is done analogously to the calculation of the latent fault metric, but at the hardware part level instead of at the item level.

NOTE 5 This requirement is still applicable if both dual-point faults contributing to the plausible dual-point failure can occur in the same hardware part.

EXAMPLE The safety mechanism "parity" lies within the hardware part "RAM". Therefore, both dual-point faults contributing to the dual-point failure "fault in RAM cell and fault in parity safety mechanism" lie within the same hardware part "RAM". For both dual-point faults to be considered acceptable the hardware part "RAM" needs to fulfil the targets of failure rate class and diagnostic coverage stated in Table 9.

9.4.3.12 This requirement applies to ASIL C and D of the safety goal. For failure rate classes i, $i > 3$, a dual-point fault contributing to a plausible dual-point failure shall be considered acceptable if the diagnostic coverage is greater than or equal to $[100 - 10^{(4-i)}]$ % for ASIL D or greater than or equal to $[100 - 10^{(5-i)}]$ % for ASIL C.

9.4.3.13 This requirement applies to ASIL C and D of the safety goal. If the requirements of 9.4.3.11 or 9.4.3.12 could not be met then a plausible dual-point failure shall be considered as acceptable if its probability of occurrence, expressed in terms of average probability per hour over the operational lifetime of the item, is less than or equal to:

a) one tenth of the value corresponding to failure rate class 1, for a safety goal ASIL D; and

b) one tenth of the value corresponding to failure rate class 2, for a safety goal ASIL C.

9.4.4 Verification review

This requirement applies to ASIL (B), C, and D of the safety goal. A verification review of the analysis resulting from the set of requirements 9.4.2 or 9.4.3 shall be performed in order to provide evidence of its technical correctness and completeness in accordance with ISO 26262-8:2018, Clause 9.

9.5 Work products

9.5.1 Analysis of safety goal violations due to random hardware failures resulting from requirements in 9.4.2 or in 9.4.3.

9.5.2 Specification of dedicated measures for hardware, if needed, including the rationale regarding the effectiveness of the dedicated measures resulting from requirements 9.4.1.2 and 9.4.1.3.

9.5.3 Verification review report of evaluation of safety goal violations due to random hardware failures resulting from requirement 9.4.4.

10 Hardware integration and verification

10.1 Objectives

The objective of this clause is to ensure the compliance of the developed hardware with the hardware safety requirements.

10.2 General

The activities described in this clause aim at integrating hardware elements and verifying the compliance of the hardware design with the hardware safety requirements in accordance with the appropriate ASIL.

Hardware integration and verification differs from the evaluation of hardware elements activity of ISO 26262-8:2018, Clause 13, which gives evidence of the suitability of hardware elements for their use as parts of items, systems or elements developed in compliance with ISO 26262.

10.3 Inputs of this clause

10.3.1 Prerequisites

The following information shall be available:

— hardware safety requirements specification in accordance with 6.5.1; and

— hardware design specification in accordance with 7.5.1.

10.3.2 Further supporting information

The following information can be considered:

— hardware safety analysis report (see 7.5.2).

10.4 Requirements and recommendations

10.4.1 Hardware integration and verification activities shall be performed in accordance with ISO 26262-8:2018, Clause 9.

10.4.2 Hardware integration and verification shall be coordinated with specification of integration and test strategy specified in ISO 26262-4:2018, 7.4.1.

NOTE If ASIL decomposition is applied, as defined in ISO 26262-9:2018, Clause 5, the corresponding integration activities of the decomposed elements, and the subsequent activities, are applied at the ASIL before decomposition.

10.4.3 The safety-related hardware parts shall be qualified according to well-established procedures based on worldwide quality standards or equivalent company standards.

EXAMPLE Qualification in accordance with ISO 16750, or with AEC-Q100 or AEC-Q200 standards, for electronic parts.

10.4.4 To provide evidence that appropriate test cases for the selected hardware integration tests are specified, test cases shall be derived using an appropriate combination of methods as listed in Table 10.

Table 10 — Methods for deriving test cases for hardware integration testing

Methods		ASIL			
		A	B	C	D
1a	Analysis of requirements	++	++	++	++
1b	Analysis of internal and external interfaces	+	++	++	++
1c	Generation and analysis of equivalence classes[a]	+	+	++	++
1d	Analysis of boundary values[b]	+	+	++	++
1e	Knowledge or experience based error guessing[c]	++	++	++	++
1f	Analysis of functional dependencies	+	+	++	++
1g	Analysis of common limit conditions, sequences and sources of dependent failures	+	+	++	++
1h	Analysis of environmental conditions and operational use cases	+	++	++	++
1i	Standards if existing[d]	+	+	+	+
1j	Analysis of significant variants[e]	++	++	++	++

a In order to derive the necessary test cases efficiently, analysis of similarities can be conducted.

b For example, values approaching and crossing the boundaries between specified values, and out of range values.

c "Error guessing tests" can be based on data collected through a lessons-learned process, or expert judgment, or both. It can be supported by FMEA.

d Existing standards include ISO 16750 and ISO 11452.

e The analysis of significant variants includes worst-case analysis.

10.4.5 The hardware integration and verification activities shall verify the completeness and correctness of the implementation of the hardware safety requirements. To achieve these objectives, the methods listed in Table 11 shall be considered.

Table 11 — Hardware integration tests to verify the completeness and correctness of the implementation of the hardware safety requirements

Methods		ASIL			
		A	B	C	D
1	Functional testing[a]	++	++	++	++
2	Fault injection testing[b]	+	+	++	++
3	Electrical testing[c]	++	++	++	++

a Functional testing aims at verifying that the specified characteristics of the item have been achieved. The item is given input data, which adequately characterises the expected normal operation. The outputs are observed and their response is compared with that given by the specification. Anomalies with respect to the specification and indications of an incomplete specification are analysed.

b Refer to ISO 26262-11:2018, 4.8 for further details on fault injection for semiconductor component.

c Electrical testing aims at verifying compliance with hardware safety requirements within the specified (static and dynamic) voltage range. Existing standards include ISO 16750 and ISO 11452.

10.4.6 The hardware integration and verification activities shall verify the durability and robustness of hardware against environmental and operational stress factors. To achieve these objectives, the methods listed in Table 12 shall be considered.

Table 12 — Hardware integration tests to verify durability, robustness and operation under stresses

Methods		ASIL			
		A	B	C	D
1a	Environmental testing with basic functional verification[a]	++	++	++	++
1b	Expanded functional test[b]	o	+	+	++
1c	Statistical test[c]	o	o	+	++
1d	Worst case test[d]	o	o	o	+
1e	Over limit test[e]	+	+	+	+
1f	Mechanical test[f]	++	++	++	++
1g	Accelerated life test[g]	+	+	++	++
1h	Mechanical Endurance test[h]	++	++	++	++
1i	EMC and ESD test[i]	++	++	++	++
1j	Chemical test[j]	++	++	++	++

[a] During environmental testing with basic functional verification the hardware is put under various environmental conditions during which the hardware requirements are assessed. ISO 16750-4 can be applied.

[b] Expanded functional testing checks the functional behaviour of the item in response to input conditions that are expected to occur only rarely (for instance extreme mission profile values), or that are outside the specification of the hardware (for instance, an incorrect command). In these situations, the observed behaviour of the hardware element is compared with the specified requirements.

[c] Statistical tests aim at testing the hardware element with input data selected in accordance with the expected statistical distribution of the real mission profile. The acceptance criteria are defined so that the statistical distribution of the results confirms the required failure rate.

[d] Worst-case testing aims at testing cases found during worst-case analysis. In such a test, environmental conditions are changed to their highest permissible marginal values defined by the specification. The related responses of the hardware are inspected and compared with the specified requirements.

[e] In over limit testing, the hardware elements are submitted to environmental or functional constraints increasing progressively to values more severe than specified until they stop working or they are destroyed. The purpose of this test is to determine the margin of robustness of the elements under test with respect to the required performance.

[f] Mechanical test applies to mechanical properties such as tensile strength. ISO 16750-3 can be applied.

[g] Accelerated life test aims at predicting the behaviour evolution of a product in its normal operational conditions by submitting it to stresses higher than those expected during its operational lifetime. Accelerated testing is based on an analytical model of failure mode acceleration.

[h] The aim of these tests is to study the mean time to failure or the maximum number of cycles that the element can withstand. Test can be performed up to failure or by damage evaluation.

[i] ISO 7637-2, ISO 7637-3, ISO 11452-2 and ISO 11452-4 can be applied for EMC tests; ISO 10605 can be applied for ESD tests.

[j] For chemical tests, ISO 16750-5 can be applied.

10.5 Work products

10.5.1 Hardware integration and verification specification resulting from requirements 10.4.1 to 10.4.6.

10.5.2 Hardware integration and verification report resulting from requirements 10.4.1 to 10.4.6.

Annex A
(informative)

Overview of and workflow of product development at the hardware level

Table A.1 provides an overview of objectives, prerequisites and work products of the particular phases of product development at the hardware level.

NOTE ISO 26262-11:2018 provides guidelines on how to tailor the following work products for integrated circuits.

Table A.1 — Overview of product development at the hardware level

Clause	Objectives	Prerequisites	Work products
6 Specification of hardware safety requirements	a) to specify the hardware safety requirements. They are derived from the technical safety concept and the system architectural design specification; b) to refine the hardware-software interface (HSI) specification initiated in ISO 26262-4:2018, 6.4.7; and c) to verify that the hardware safety requirements and the hardware-software interface (HSI) specification are consistent with the technical safety concept and the system architectural design specification.	— technical safety concept in accordance with ISO 26262-4:2018, 6.5.2; — system architectural design specification in accordance with ISO 26262-4:2018, 6.5.3; and — hardware-software interface (HSI) specification in accordance with ISO 26262-4:2018, 6.5.4.	6.5.1 Hardware safety requirements specification (including test and evaluation criteria) **resulting from requirements** 6.4.1 to 6.4.8 6.5.2 Hardware-software interface specification (HSI) (refined) **resulting from requirement** 6.4.10 6.5.3 Hardware safety requirements verification report **resulting from requirements** 6.4.9 and 6.4.11
7 Hardware design	a) to create a hardware design that: — supports the safety-oriented analyses; — considers the results of the safety-oriented analyses; — fulfils the hardware safety requirements; — fulfils the hardware-software interface (HSI) specification; — is consistent with system architectural design specification; and — satisfies the required hardware design properties; and b) to specify requirements and to provide information regarding functional safety of the hardware during production, operation, service and decommissioning; and	— hardware safety requirements specification in accordance with 6.5.1; — hardware-software interface specification (HSI) (refined) in accordance with 6.5.2; and — system architectural design specification in accordance with ISO 26262-4:2018, 6.5.3.	7.5.1 Hardware design specification **resulting from requirements in** 7.4.1 and 7.4.2 7.5.2 Hardware safety analysis report **resulting from requirements in** 7.4.3 7.5.3 Hardware design verification report **resulting from requirements in** 7.4.4 7.5.4 Specification of requirements related to production, operation, service and decommissioning **resulting from requirements in** 7.4.5

Table A.1 *(continued)*

Clause	Objectives	Prerequisites	Work products
	c) to verify: — that the hardware design can fulfil the hardware safety requirements and the hardware-software inter-face (HSI) specification; — the validity of the assumptions used to develop each SEooC integrated in the developed hardware; and — the suitability of the safety-related special characteristics to achieve functional safety during produc-tion and service.		

Table A.1 *(continued)*

Clause	Objectives	Prerequisites	Work products
8 Evaluation of the hardware architectural metrics	to provide evidence based on the hardware architectural metrics for the suitability of the hardware architectural design of the item with respect to detection and control of safety-related random hardware failures.	— hardware safety requirements specification in accordance with 6.5.1; — hardware design specification in accordance with 7.5.1; and — hardware safety analysis report in accordance with 7.5.2.	8.5.1 **Analysis of the effectiveness of the architecture of the item to cope with the random hardware failures** resulting from requirements 8.4.1 to 8.4.8 8.5.2 Verification review report of evaluation of the effectiveness of the architecture of the item to cope with the random hardware **failures resulting from requirement** 8.4.9
9 Evaluation of safety goal violations due to random HW failures	to provide evidence that the residual risk of a safety goal violation, due to random hardware failures of the item, is sufficiently low.	— hardware safety requirements specification in accordance with 6.5.1; — hardware design specification in accordance with 7.5.1; and — hardware safety analysis report in accordance with 7.5.2.	9.5.1 Analysis of safety goal violations due to random hardware failures **resulting from requirements in** 9.4.2 or in 9.4.3 9.5.2 Specification of dedicated measures for hardware, **if needed, including the rationale regarding the effectiveness of the dedicated measures resulting from requirements** 9.4.1.2 **and** 9.4.1.3 9.5.3 Verification review report of evaluation of safety goal violations due to random hardware failures **resulting from requirement** 9.4.4
10 Hardware integration and verification	to ensure the compliance of the developed hardware with the hardware safety requirements.	— hardware safety requirements specification in accordance with 6.5.1; and — hardware design specification in accordance with 7.5.1.	10.5.1 Hardware integration and verification specification **resulting from requirements** 10.4.1 to 10.4.6 10.5.2 Hardware integration and verification report **resulting from requirements** 10.4.1 **to** 10.4.6

Annex B
(informative)

Failure mode classification of a hardware element

Failure modes of a hardware element can be classified as shown in Figure B.1. The flow diagram shown in Figure B.2 describes how a failure mode of a hardware element can be placed into one of these classifications.

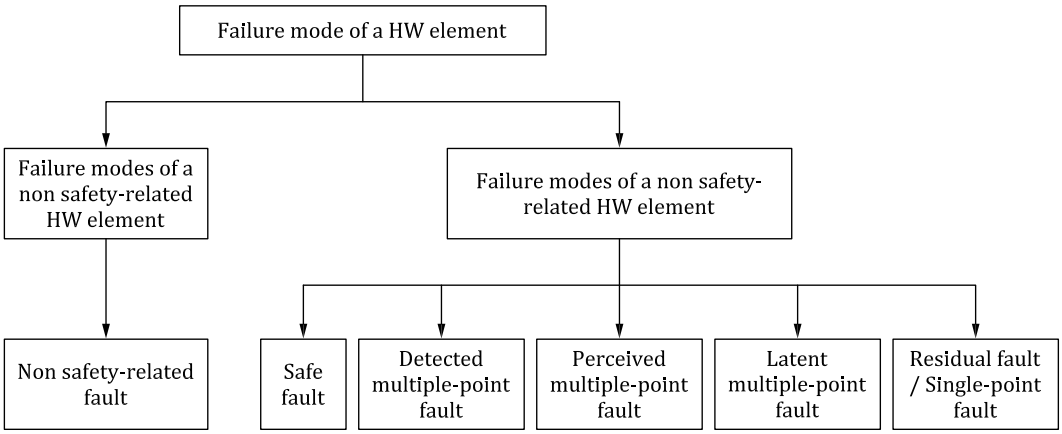


Figure B.1 — Failure mode classifications of a hardware element

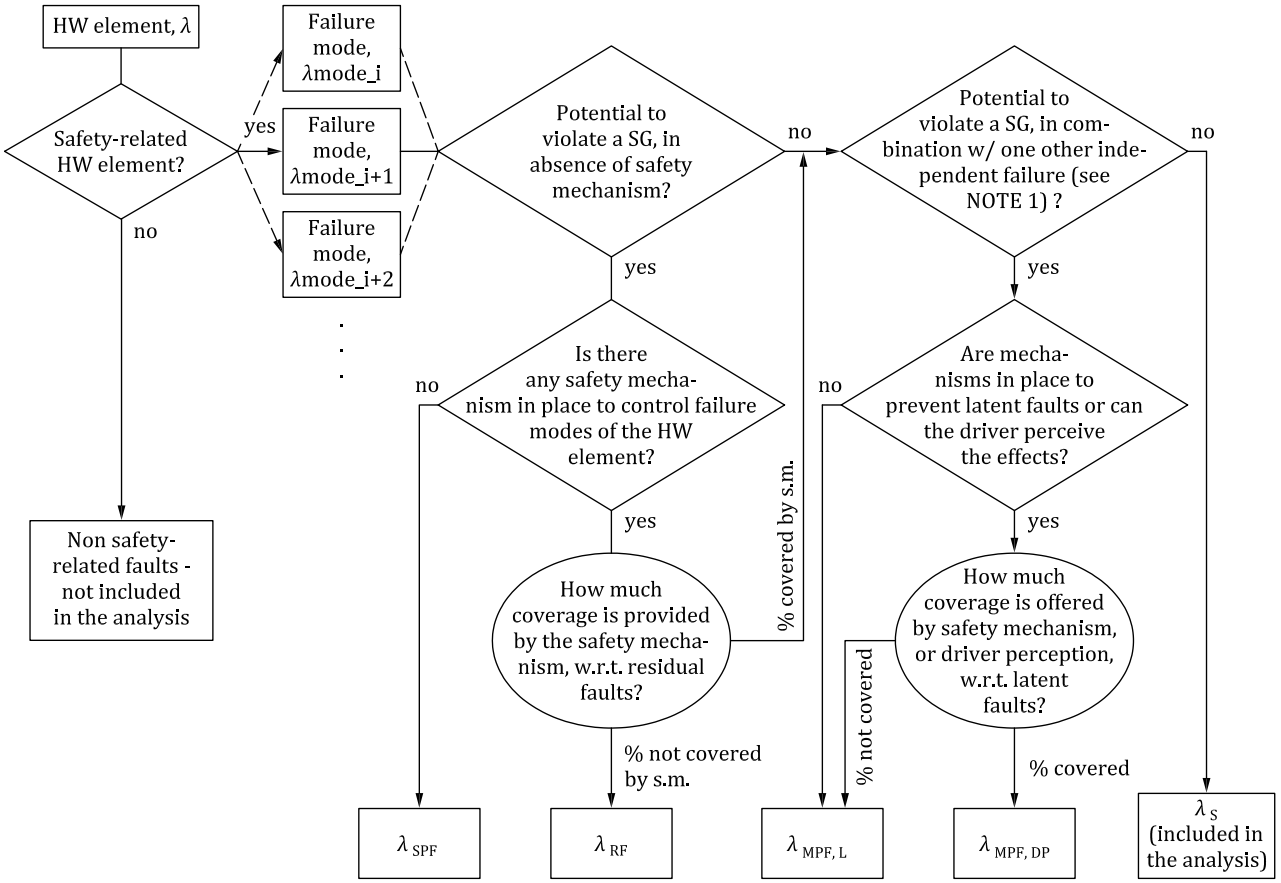


Figure B.2 — Example of flow diagram for failure mode classification

NOTE 1 The elements with failures that do not significantly increase the probability of the violation of a safety goal can be omitted from the analyses and their failure modes can be classified as safe faults, e.g. hardware elements whose faults only contribute to multiple-point failures of order n, with n > 2, are considered as safe faults unless shown to be relevant in the technical safety concept.

EXAMPLE The portion of a failure mode of a hardware element that has the potential to violate a safety goal in absence of safety mechanism, which is covered by two independent safety mechanisms can be considered as a multiple-point fault of order of 3. It can be considered as safe unless it is shown to be relevant in the safety concept.

NOTE 2 The same fault can be placed in different classes when being considered for different safety goals.

Annex C
(normative)

Hardware architectural metrics

C.1 Fault classification and diagnostic coverage

C.1.1 This requirement applies to ASIL (B), C, and D of the safety goal. Hardware architectural metrics shall be defined for the hardware of an item and shall address only safety-related hardware elements that have the potential to contribute significantly to the violation of the safety goal.

EXAMPLE Hardware elements whose faults are multiple-point faults of order n, with n > 2, can be omitted from the calculations unless shown to be relevant in the technical safety concept.

C.1.2 This requirement applies to ASIL (B), C, and D of the safety goal. Each fault occurring in a safety-related hardware element shall be classified, as illustrated in Figure B.1, as:

a) single-point fault;

b) residual fault;

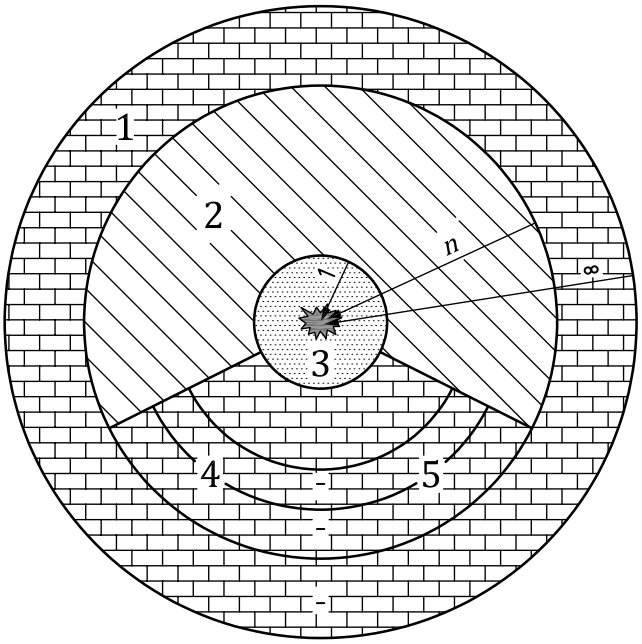
EXAMPLE 1 A hardware element that can have “open”, “short to ground”, and “short to high” faults, but only the “open” and “short to ground” faults are covered by safety mechanisms. The “short to high” fault is a residual fault, since it is not covered by a safety mechanism, if it leads to the violation of the specified safety goal.

c) multiple-point fault; or

NOTE 1 Classification of multiple-point fault needs to distinguish between latent, detected and perceived.

d) safe fault.

Figure C.1 gives a graphical representation of fault classification of safety-related hardware elements of an item:



Key

1 safe faults
2 latent multiple-point faults
3 single-point or residual faults
4 detected multiple-point faults
5 perceived multiple-point faults

Figure C.1 — Fault classification of safety-related hardware elements of an item

In this graphical representation:

— the distance n represents the number of independent faults present at the same time that cause a violation of the safety goal (n = 1 for single-point or residual faults, n = 2 for dual-point faults, etc.);

— faults with distance equal to n are located in the area between the circles n and n-1; and

— multiple-point faults of distance strictly higher than n = 2 shall be considered as safe faults unless shown to be relevant in the technical safety concept.

NOTE 2 In the case of a transient fault, for which a safety mechanism restores the item to a fault free state, such a fault can be considered as a detected multiple-point fault even if the driver is never informed of its existence.

EXAMPLE 2 In the case of an error correction code used to protect a memory against transient faults, the item is restored to a fault free state if the safety mechanism – in addition to delivering a correct value to the CPU - repairs the content of the flipped bit inside the memory array (e.g. by writing back the corrected value).

The failure rate, λ, of each safety-related hardware element can therefore be expressed according to Equation (C.1) (assuming all failures are independent and follow the exponential distribution), as follows:

$$\lambda = \lambda_{SPF} + \lambda_{RF} + \lambda_{MPF} + \lambda_{S} \quad \text{(C.1)}$$

where

λ_{SPF} is the failure rate associated with hardware element single-point faults;

λ_{RF} is the failure rate associated with hardware element residual faults;

λ_{MPF} is the failure rate associated with hardware element multiple-point faults;

λ_{S} is the failure rate associated with hardware element safe faults.

The failure rate associated with hardware element multiple-point faults, λ_{MPF}, can be expressed according to Equation (C.2), as follows:

$$\lambda_{MPF} = \lambda_{MPF,DP} + \lambda_{MPF,L} \quad \text{(C.2)}$$

where

$\lambda_{MPF,DP}$ is the failure rate associated with hardware element perceived or detected multiple-point faults;

$\lambda_{MPF,L}$ is the failure rate associated with hardware element latent faults.

The failure rate assigned to residual faults can be determined using the diagnostic coverage of safety mechanisms that avoid single-point faults of the hardware element. Equation (C.3) gives a conservative estimation of the failure rate associated with the residual faults:

$$\lambda_{RF} \leq \lambda_{RF,est} = \lambda \times \left(1 - \frac{K_{DC,RF}}{100\,\%}\right) \quad \text{(C.3)}$$

where

$\lambda_{RF,est}$ is the estimated failure rate with respect to residual faults;

$K_{DC,RF}$ (also known as DC_{RF}) is the diagnostic coverage with respect to residual faults, expressed as a percentage.

NOTE 3 When failure mode distribution, and coverage of failure modes, are known, λ_{RF} can be calculated as follows:

$$\lambda_{RF} = \sum_{alli} \lambda \times D_{FMi,SR} \times \left(1 - F_{FMi,safe}\right) \times F_{FMi,PVSG} \times \left(1 - K_{FMCi,RF}\right) \quad \text{(C.4)}$$

where

$\lambda \times D_{FMi,SR}$ the failure rate associated to i^{th} failure mode of the safety related hardware element;

$F_{Mi,safe}$ is the fraction of faults of i^{th} failure mode which are considered as safe;

$(1 - F_{Mi,safe}) \times F_{FMi,PVSG}$ is the fraction of faults of i^{th} failure mode which have the potential to directly violate the safety goal in absence of safety mechanism;

$K_{FMCi,RF}$ is the failure mode coverage of i^{th} failure mode with respect to residual faults.

The failure rate assigned to latent faults can be determined using the diagnostic coverage of safety mechanisms that avoid latent faults of the hardware element. Equation (C.5) gives a conservative estimation of the failure rate associated with latent faults:

$$\lambda_{MPF,L} \leq \lambda_{MPF,L,est} = \lambda \times \left(1 - \frac{K_{DC,MPF,L}}{100\,\%}\right) \quad \text{(C.5)}$$

where

$\lambda_{MPF,L,est}$ is the estimated failure rate with respect to latent faults;

$K_{DC,MPF,L}$ (also known as $DC_{MPF,L}$) is the diagnostic coverage with respect to latent faults, expressed as a percentage.

NOTE 4 When failure modes distribution and coverage of failure modes are known, $\lambda_{MPF,L}$, can be calculated as follows:

$$\lambda_{MPF,L} = \sum_{alli} \lambda \times D_{FMi,SR} \times \left(1 - F_{FMi,safe}\right) \times \left[F_{FMi,PVSG} \times K_{FMCi,RF} + \left(1 - F_{FMi,PVSG}\right)\right] \times \left(1 - K_{FMCi,MPF}\right) \quad \text{(C.6)}$$

where

$\lambda \times D_{FMi,SR}$ is the failure rate associated to i^{th} failure mode of the safety related hardware element;

$F_{FMI,safe}$ is the fraction of faults of i^{th} failure mode which are considered as safe;

$(1 - F_{FMi,safe}) \times F_{FMi,PVSG}$ is the fraction of faults of i^{th} failure mode which have the potential to directly violate the safety goal in absence of safety mechanism;

$(1 - F_{FMi,safe}) \times (1 - F_{FMi,PVSG})$ is the fraction of faults of i^{th} failure mode which are not considered as safe but which don't have the potential to directly violate the safety goal in absence of safety mechanism;

$K_{FMCi,RF}$ is the failure mode coverage of i^{th} failure mode with respect to residual faults;

$K_{FMCi,MPF}$ is the failure mode coverage of i^{th} failure mode with respect to latent faults.

NOTE 5 For this purpose, Annex D can be used as a basis for diagnostic coverage with the claimed DC supported by a proper rationale.

NOTE 6 If the above estimates are considered too conservative, then a detailed analysis of the failure modes of the hardware element can classify each failure mode into one of the fault classes (single-point faults, residual faults, latent, detected or perceived multiple-point faults or safe faults) with respect to the specified safety goal and determine the failure rates distributed to the failure modes. Annex B describes a flow diagram that can be used to classify the faults.

C.2 Single-point fault metric

C.2.1 This metric reflects the robustness of the item to single-point and residual faults either by coverage from safety mechanisms or by design (primarily safe faults). A high single-point fault metric implies that the proportion of single-point faults and residual faults in the hardware of the item is low.

C.2.2 This requirement applies to ASIL (B), C, and D of the safety goal. The calculation in Equation (C.7) shall be used to determine the single-point fault metric:

$$1 - \frac{\sum_{SR,HW} \left(\lambda_{SPF} + \lambda_{RF}\right)}{\sum_{SR,HW} \lambda} = \frac{\sum_{SR,HW} \left(\lambda_{MPF} + \lambda_{S}\right)}{\sum_{SR,HW} \lambda} \quad \text{(C.7)}$$

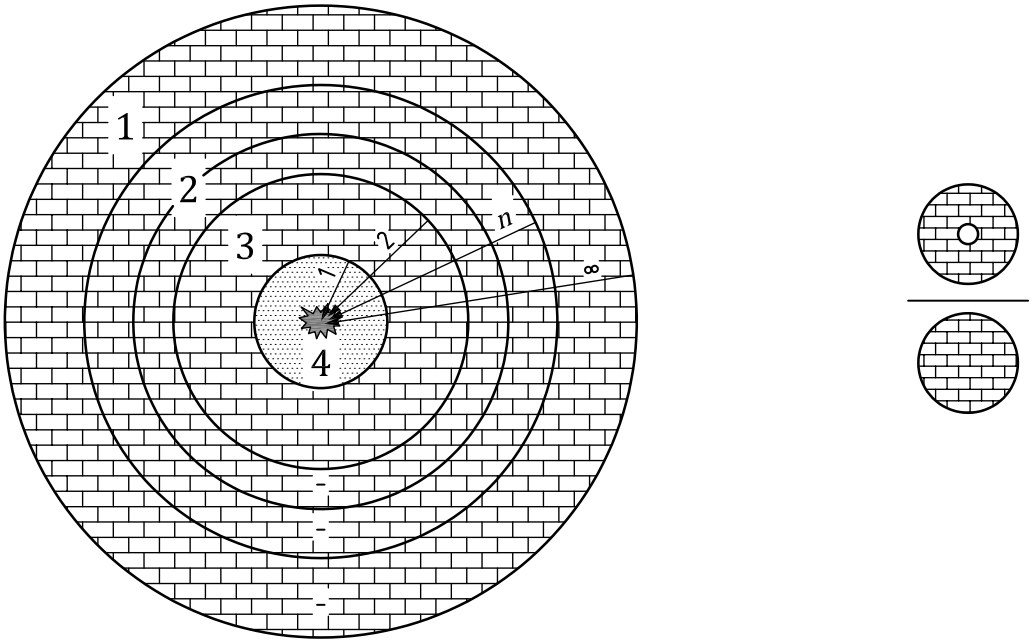
where $\sum_{SR,HW} \lambda_x$ is the sum of λ_x of the safety-related hardware elements of the item to be considered for the metrics.

NOTE 1 Only the safety-related hardware elements of the item are considered for this metric.

EXAMPLE Hardware elements where all the faults are safe or multiple-point faults of order n, with n > 2, could be omitted from the calculations unless shown to be relevant in the technical safety concept.

NOTE 2 Figure C.2 gives a graphical representation of the single-point fault metric.

NOTE 3 An example of calculation of "single-point fault metric" is given in Annex E.



Key

1 safe faults
2 multiple-point faults
3 dual-point faults
4 single-point or residual faults

Figure C.2 — Graphical representation of the single-point fault metric

C.3 Latent-fault metric

C.3.1 This metric reflects the robustness of the item to latent faults either by coverage of faults in safety mechanisms or by the driver recognizing that the fault exists before the violation of the safety goal, or by design (primarily safe faults). A high latent-fault metric implies that the proportion of latent faults in the hardware is low.

C.3.2 This requirement applies to ASIL (B), (C), and D of the safety goal. The calculation in Equation (C.8) shall be used to determine the latent-fault metric:

$$1-\frac{\sum_{SR,HW}\left(\lambda_{MPF,L}\right)}{\sum_{SR,HW}\left(\lambda-\lambda_{SPF}-\lambda_{RF}\right)}=\frac{\sum_{SR,HW}\left(\lambda_{MPF,DP}+\lambda_{S}\right)}{\sum_{SR,HW}\left(\lambda-\lambda_{SPF}-\lambda_{RF}\right)} \quad \text{(C.8)}$$

where $\sum_{SR,HW}\lambda_x$ is the sum of λ_x of the safety-related hardware elements of the item to be considered for the metrics.

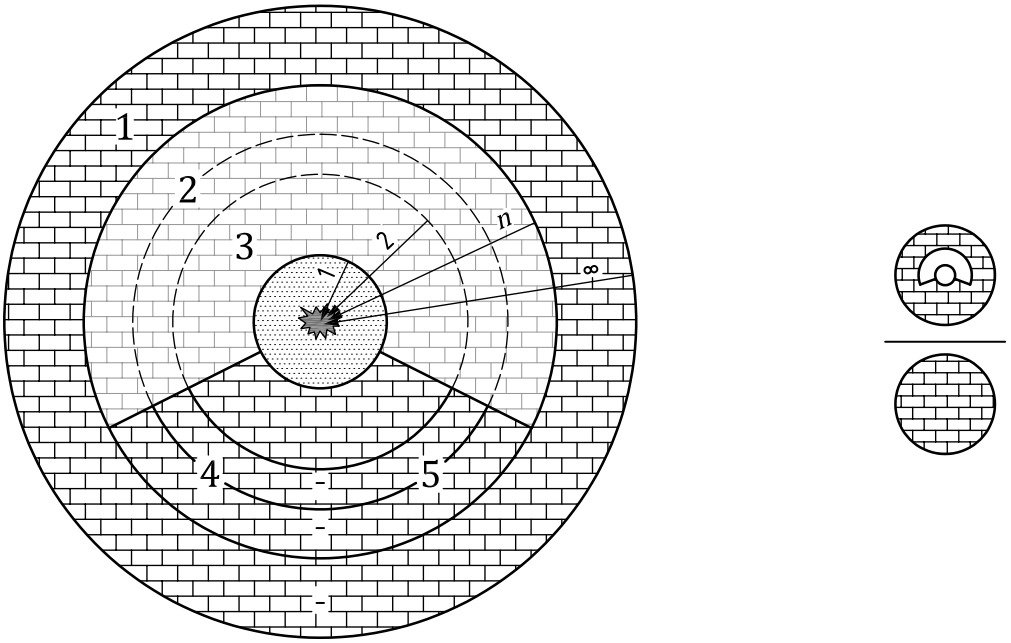
NOTE 1 Only the safety-related hardware elements of the item are considered for this metric.

EXAMPLE Hardware elements where all the faults are safe or multiple-point faults of order n, with n > 2, could be omitted from the calculations unless shown to be relevant in the technical safety concept.

NOTE 2 Figure C.3 gives a graphical representation of the latent-fault metric.

NOTE 3 An example of calculation of "latent-fault metric" is given in Annex E.

NOTE 4 For Latent Fault metrics of items implementing fault tolerance in order to address safety relevant availability requirements, it can be important to identify the multiple-point faults with a higher order than two. This can be applicable to latent faults of a redundant system, if the intention is to operate on the redundant system for significant amount of time after the primary system fails.



Key

1 safe faults
2 latent multiple-point faults
3 single-point or residual faults
4 detected multiple-point faults
5 perceived multiple-point faults

Figure C.3 — Graphical representation of the latent-fault metric

Annex D
(informative)

Evaluation of the diagnostic coverage

D.1 General

This annex is intended to be used as:

a) an evaluation of the diagnostic coverage to produce a rationale for:

c) the compliance with the single-point fault and latent-fault metrics defined in Clause 8;

d) the compliance with the evaluation of the safety goal violations due to random hardware failures as defined in Clause 9;

b) a guideline in order to choose appropriate safety mechanisms to be implemented in the E/E architecture to detect failures of elements.

Figure D.1 shows the generic hardware of an embedded system. Typical failure modes of the hardware elements of this system are shown in Table D.1. Each element listed in the leftmost column is associated with one or more failure modes which are captured in the column to the right of the element. The listing does not claim exhaustiveness and can be adjusted based on additional known failure modes or depending on the application.

Additional detail on the safety mechanisms associated with these element faults are referenced in each row (Tables D.2 to D.10). The effectiveness of these typical safety mechanisms for the given elements is categorized according to their ability to cover the listed failure modes to achieve low, medium or high diagnostic coverage of the element. These low, medium and high diagnostic coverage rankings correspond to typical coverage levels at 60 %, 90 % or 99 %, respectively.

The assignment of the failure modes and their corresponding safety mechanisms can vary from that listed in Table D.1 depending on:

a) variations in the source of the failure mode detected by the diagnostic;

b) the effectiveness of the safety mechanism;

c) the specific implementation of the safety mechanism;

d) the execution timing of the safety mechanism (periodicity);

e) the hardware technologies implemented in the system;

f) the probability of the failure modes, based on hardware in the system; and

g) a more detailed analysis of the failure modes and their classification into several sub-classes with different failure mode coverage levels.

In summary, Table D.1 provides guidelines which are adapted based on analysis of the system elements.

These guidelines do not address specific constraints that can be specified in the safety concepts in order to avoid the violation of the safety goals. These constraints, such as timing aspects (periodicity of diagnostic) for example, are not considered when evaluating the generic typical diagnostic coverage by

the safety mechanism. They will be considered when evaluating the specific diagnostic coverage by a safety mechanism used in the item to avoid the violation of the safety goals.

EXAMPLE A safety mechanism can have a high generic typical diagnostic coverage in this annex but if the diagnostic test interval used is longer than the diagnostic test interval needed to comply with the relevant fault tolerant time interval, the specific diagnostic coverage with respect to the avoidance of violation of the safety goal, will be much lower.

Therefore Tables D.2 to D.10 can be used as a starting point to evaluate the diagnostic coverage of these safety mechanisms and the claimed diagnostic coverage is supported by a proper rationale (e.g. using fault injection methods or analytical arguments). In addition, the given information is intended to help define the failure modes of the element; however, the relevant failure modes are ultimately dependent on the application in which the elements are used.

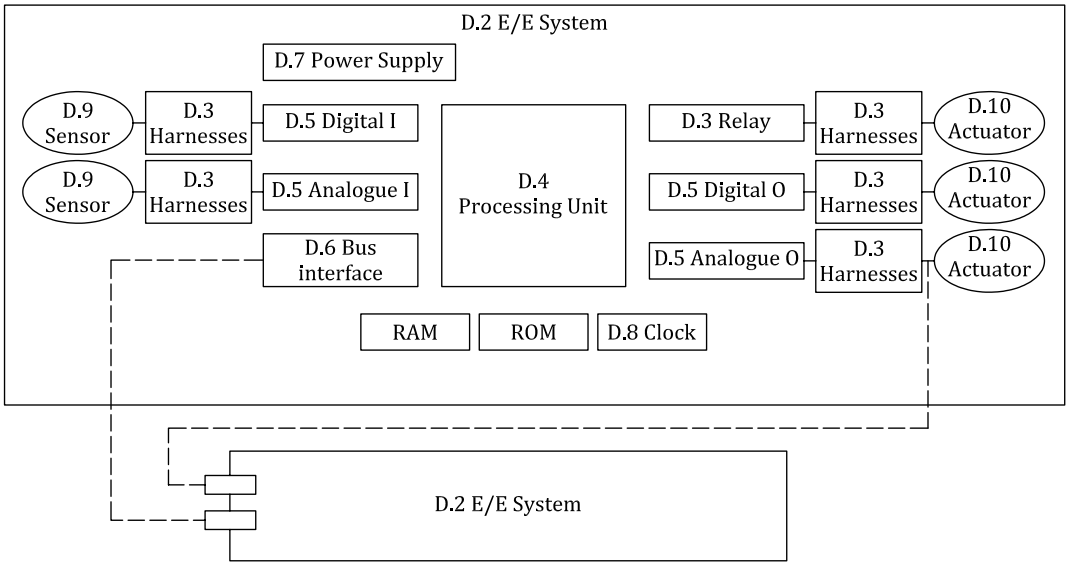


Figure D.1 — Generic hardware of a system

Tables D.2 to D.10 support the information of Table D.1 by giving guidelines on techniques for diagnostic tests. Tables D.1 to D.10 are not exhaustive and other techniques can be used, provided evidence is available to support the claimed diagnostic coverage. If justified, higher diagnostic coverage can be estimated, up to 100 % for simple or complex elements.

Table D.1 — Analysed failure modes

Element	See tables	Analysed failure modes
General elements		
E.E Systems	D.2 — E/E Systems	No generic failure modes available Detailed analysis necessary
Electrical elements		
Relays	D.3 — Electrical elements	Does not energize or de-energize Individual contacts welded
Harnesses including splice and connectors		Open circuit Contact resistance Short circuit to Ground (d.c coupled) Short circuit to Vbat Short circuit between neighbouring pins Resistive drift between pins
Sensors including signal switches	D.9 — Sensors	Detailed analysis necessary Typical failure modes to be covered include: — Out-of-range — Offsets — Stuck in range — Oscillations See also ISO 26262-11:2018, 5.5 for integrated sensors and transducers.
Final elements (actuators, lamps, buzzers, screens…)	D.10 — Actuators	No generic failure modes available Detailed analysis necessary
General semiconductor elements		
Power supply	D.7 — Power supply	Drift and oscillation Under and over Voltage Power spikes See also ISO 26262-11:2018, 5.2

NOTE 1 The relevant failure modes and fault models are identified on a case by case basis and typically depend on the technology and the implementation used. See ISO 26262-11:2018, 4.3.1 for details on semiconductor fault models.

EXAMPLE If an element has the failure modes *x*, *y*, and *z* with a failure mode distribution of *X*, *Y*, *Z* then the effective diagnostic coverage is calculated as follows:

$K_{DC} = X \times K_{FMC,x} + Y \times K_{FMC,y} + Z \times K_{FMC,z}$

where

K_{DC} is the diagnostic coverage of the hardware element;

X is the failure mode distribution for failure mode x; $K_{FMC,x}$ is the failure mode coverage of failure mode x;

Y is the failure mode distribution of failure mode y; $K_{FMC,y}$ is the failure mode coverage of failure mode y;

Z is the failure mode distribution for failure mode z; $K_{FMC,z}$ is the failure mode coverage of failure mode z; and

$X + Y + Z = 100\ \%$

NOTE 2 For semiconductors, see ISO 26262-11:2018, 4.3 for details on the relationship between fault models, failure modes and related distribution.

Table D.1 *(continued)*

Element	See tables	Analysed failure modes
Clock	D.8 — Programme sequence monitoring/Clock	Incorrect frequency Jitter See also ISO 26262-11:2018, 5.2
Non-volatile memory	ISO 26262-11:2018, Table 32	See ISO 26262-11:2018, 5.1, Table 29
Volatile memory	ISO 26262-11:2018, Table 33	See ISO 26262-11:2018, 5.1, Table 29
Digital I/O	D.5 — Analogue and digital I/O	Incorrect I/O See also ISO 26262-11:2018, 5.1, Table 30
Analogue I/O		Incorrect I/O See also ISO 26262-11:2018, 5.2, Table 36
Processing Unit	D.4 — Processing units / D.8 — Programme sequence monitoring/Clock	Incorrect output See also ISO 26262-11:2018, 5.1, Table 30

NOTE 1 The relevant failure modes and fault models are identified on a case by case basis and typically depend on the technology and the implementation used. See ISO 26262-11:2018, 4.3.1 for details on semiconductor fault models.

EXAMPLE If an element has the failure modes x, y, and z with a failure mode distribution of X, Y, Z then the effective diagnostic coverage is calculated as follows:

$K_{DC} = X \times K_{FMC,x} + Y \times K_{FMC,y} + Z \times K_{FMC,z}$

where

K_{DC} is the diagnostic coverage of the hardware element;

X is the failure mode distribution for failure mode x; $K_{FMC,x}$ is the failure mode coverage of failure mode x;

Y is the failure mode distribution of failure mode y; $K_{FMC,y}$ is the failure mode coverage of failure mode y;

Z is the failure mode distribution for failure mode z; $K_{FMC,z}$ is the failure mode coverage of failure mode z; and

$X + Y + Z = 100\ \%$

NOTE 2 For semiconductors, see ISO 26262-11:2018, 4.3 for details on the relationship between fault models, failure modes and related distribution.

Table D.1 *(continued)*

Element	See tables	Analysed failure modes
Communication		
Data transmission (to be analysed with ISO 26262-6:2018, D.2.4)	D.6 — Communication bus (serial, parallel)	Loss of communication peer Message corruption Message unacceptable delay Message loss Unintended message repetition Incorrect sequencing of messages Message insertion Message masquerading Message incorrect addressing

NOTE 1 The relevant failure modes and fault models are identified on a case by case basis and typically depend on the technology and the implementation used. See ISO 26262-11:2018, 4.3.1 for details on semiconductor fault models.

EXAMPLE If an element has the failure modes x, y, and z with a failure mode distribution of X, Y, Z then the effective diagnostic coverage is calculated as follows:

$K_{DC} = X \times K_{FMC,x} + Y \times K_{FMC,y} + Z \times K_{FMC,z}$

where

K_{DC} is the diagnostic coverage of the hardware element;

X is the failure mode distribution for failure mode x; $K_{FMC,x}$ is the failure mode coverage of failure mode x;

Y is the failure mode distribution of failure mode y; $K_{FMC,y}$ is the failure mode coverage of failure mode y;

Z is the failure mode distribution for failure mode z; $K_{FMC,z}$ is the failure mode coverage of failure mode z; and

$X + Y + Z = 100\ \%$

NOTE 2 For semiconductors, see ISO 26262-11:2018, 4.3 for details on the relationship between fault models, failure modes and related distribution.

Table D.2 — E/E Systems

Safety mechanism/ measure	See overview of techniques	Typical diagnostic coverage considered achievable	Notes
Failure detection by on-line monitoring	D.2.1.1	Low	Depends on diagnostic coverage of failure detection
Comparator	D.2.1.2	High	Depends on the quality of the comparison
Majority voter	D.2.1.3	High	Depends on the quality of the voting
Dynamic principles	D.2.2.1	Medium	Depends on diagnostic coverage of failure detection
Analogue monitoring of digital signals	D.2.2.2	Low	—
Self-test by software cross exchange between two independent units	D.2.3.3	Medium	Depends on the quality of the self-test

Table D.3 — Electrical elements

Safety mechanism/ measure	See overview of techniques	Typical diagnostic coverage considered achievable	Notes
Failure detection by on-line monitoring	D.2.1.1	High	Depends on diagnostic coverage of failure detection
NOTE This table deals only with safety mechanisms dedicated to electrical elements. General techniques like a technique based on a data comparison (see D.2.1.2) are also able to detect failures of electrical elements but are not integrated in this table (already included in Table D.2 — E/E Systems).			

NOTE The following tables deal with safety mechanisms mainly applied to components at a system level. Additional details on safety mechanisms that could be integrated in the component are described in ISO 26262-11:2018:

— 5.1 for digital components;

— 5.2 for analogue and mixed signal components;

— 5.3 for programmable logic devices;

— 5.4 for multi-core components; and

— 5.5 for sensors & transducers.

Table D.4 — Processing units

Safety mechanism/ measure	See overview of techniques	Typical diagnostic coverage considered achievable	Notes
Self-test by software: limited number of patterns (one channel)	D.2.3.1	Medium	Depends on the quality of the self-test
Self-test by software cross exchange between two independent units	D.2.3.3	Medium	Depends on the quality of the self-test
Self-test supported by hardware (one-channel)	D.2.3.2	Medium	Depends on the quality of the self-test
Software diversified redundancy (one hardware channel)	D.2.3.4	High	Depends on the quality of the diversification. Common mode failures can reduce diagnostic coverage
Reciprocal comparison by software	D.2.3.5	High	Depends on the quality of the comparison
HW redundancy (e.g. dual core lockstep, asymmetric redundancy, coded processing)	D.2.3.6	High	It depends on the quality of redundancy. Common mode failures can reduce diagnostic coverage
Configuration register test	D.2.3.7	High	Configuration registers only
Stack over/under flow Detection	D.2.3.8	Low	Stack boundary test only
Integrated hardware consistency monitoring	D.2.3.9	High	Coverage for illegal hardware exceptions only
NOTE This table deals only with safety mechanisms dedicated to processing units. General techniques like one based on data comparison (see D.2.1.2) are also able to detect failures of electrical elements but are not integrated in this table (already included in Table D.2 — E/E Systems).			

Table D.5 — Analogue and digital I/O

Safety mechanism/ measure	See overview of techniques	Typical diagnostic coverage considered achievable	Notes
Failure detection by on-line monitoring (Digital I/O)[a]	D.2.1.1	Low	Depends on diagnostic coverage of failure detection
Test pattern	D.2.4.1	High	Depends on type of pattern
Code protection for digital I/O	D.2.4.2	Medium	Depends on type of coding
Multi-channel parallel output	D.2.4.3	High	—
Monitored outputs	D.2.4.4	High	Only if dataflow changes within diagnostic test interval
Input comparison/ voting (1oo2, 2oo3 or better redundancy)	D.2.4.5	High	Only if dataflow changes within diagnostic test interval
[a] Digital I/O can be periodic.			

Table D.6 — Communication bus (serial, parallel)

Safety mechanism/ measure	See overview of techniques	Typical diagnostic coverage considered achievable	Notes
One-bit hardware redundancy	D.2.5.1	Low	—
Multi-bit hardware redundancy	D.2.5.2	Medium	—
Read back of sent message	D.2.5.9	Medium	—
Complete hardware redundancy	D.2.5.3	High	Common mode failures can reduce diagnostic coverage
Inspection using test patterns	D.2.5.4	High	—
Transmission redundancy	D.2.5.5	Medium	Depends on type of redundancy. Effective only against transient faults
Information redundancy	D.2.5.6	Medium	Depends on type of redundancy
Frame counter	D.2.5.7	Medium	—
Timeout monitoring	D.2.5.8	Medium	—
Combination of information redundancy, frame counter and timeout monitoring	D.2.5.6, D.2.5.7 and D.2.5.8	High	For systems without hardware redundancy or test patterns, high coverage can be claimed for the combination of these safety mechanisms

Table D.7 — Power supply

Safety mechanism/ measure	See overview of techniques	Typical diagnostic coverage considered achievable	Notes
Voltage or current control (input)	D.2.6.1	Low	—
Voltage or current control (output)	D.2.6.2	High	—

Table D.8 — Programme sequence monitoring/Clock

Safety mechanism/ measure	See overview of techniques	Typical diagnostic coverage considered achievable	Notes
Watchdog with separate time base without time-window	D.2.7.1	Low	—
Watchdog with separate time base and time-window	D.2.7.2	Medium	Depends on time restriction for the time-window
Logical monitoring of programme sequence	D.2.7.3	Medium	Only effective against clock failures if external temporal events influence the logical program flow. Provides coverage for internal hardware failures (such as interrupt frequency errors) that can cause the software to run out of sequence
Combination of temporal and logical monitoring of programme sequence	D.2.7.4	High	—
Combination of temporal and logical monitoring of programme sequences with time dependency	D.2.7.5	High	Provides coverage for internal hardware failures that can cause the software to run out of sequence. When implemented with asymmetrical designs, provides coverage regarding communication sequence between main and monitoring device NOTE Method to be designed to account for execution jitter from interrupts, CPU loading, etc.

Table D.9 — Sensors

Safety mechanism/ measure	See overview of techniques	Typical diagnostic coverage considered achievable	Notes
Failure detection by on-line monitoring	D.2.1.1	Low	Depends on diagnostic coverage of failure detection
Test pattern	D.2.4.1	High	—
Input comparison/ voting (1oo2, 2oo3 or better redundancy)	D.2.4.5	High	Only if dataflow changes within diagnostic test interval
Sensor valid range	D.2.8.1	Low	Detects shorts to ground or power and some open circuits
Sensor correlation	D.2.8.2	High	Detects in range failures
Sensor rationality check	D.2.8.3	Medium	—

Table D.10 — Actuators

Safety mechanism/ measure	See overview of techniques	Typical diagnostic coverage considered achievable	Notes
Failure detection by on-line monitoring	D.2.1.1	Low	Depends on diagnostic coverage of failure detection
Test pattern	D.2.4.1	High	—
Monitoring (i.e. coherence control)	D.2.9.1	High	Depends on diagnostic coverage of failure detection

D.2 Overview of techniques for embedded diagnostic self-tests

D.2.1 Electrical

Global objective: To control failures in electromechanical elements.

D.2.1.1 Failure detection by on-line monitoring

NOTE 1 This technique/measure is referenced in Tables D.2, D.3, D.5, D.9 and D.10.

Aim: To detect failures by monitoring the behaviour of the system in response to the normal (on-line) operation.

Description: Under certain conditions, failures can be detected using information about (for example) the time behaviour of the system. For example, if a switch is normally actuated and does not change state at the expected time, a failure will have been detected. It is not usually possible to localize the failure.

NOTE 2 In general, there is no specific hardware element for the realisation of the on-line monitoring diagram. On-line monitoring detects abnormal behaviour of the system with respect to certain conditions of activation. For example, if such parameter is inverted when the vehicle speed is different from zero, then detection of incoherence between this parameter and vehicle speed leads to failure detection.

D.2.1.2 Comparator

NOTE This technique/measure is referenced in Table D.2.

Aim: To detect, as early as possible, (non-simultaneous) failures in independent hardware or software.

Description: The output signals of independent hardware or output information of independent software, are compared cyclically or continuously by a comparator. Detected differences lead to a failure message. For instance: two processing units exchange data (including results, intermediate results and test data) reciprocally. A comparison of the data is carried out using software in each unit and detected differences lead to a failure message.

D.2.1.3 Majority voter

NOTE 1 This technique/measure is referenced in Table D.2.

Aim: To detect and mask failures in one of at least three channels.

Description: A voting unit using the majority principle (2 out of 3, 3 out of 3, or *m* out of *n*) is used to detect and mask failures.

NOTE 2 Unlike the comparator, the majority voter technique increases the availability by ensuring the functionality of the redundant channel even after the loss of one channel.

D.2.2 Electronic

Global objective: To control failure in solid-state elements.

D.2.2.1 Dynamic principles

NOTE This technique/measure is referenced in Table D.2.

Aim: To detect static failures by dynamic signal processing.

Description: A forced change of otherwise static signals (internally or externally generated) helps to detect static failures in elements. This technique is often associated with electromechanical elements.

D.2.2.2 Analogue monitoring of digital signals

NOTE This technique/measure is referenced in Table D.2.

Aim: To improve confidence in measured signals.

Description: A binary signal is evaluated on an analogue level in order to detect illegal signal levels.

EXAMPLE A switch has the signal high as closed and low as open. The monitoring detects if the output level is within the specified ranges. The specified ranges are chosen in such a way that short to ground, short to supply voltage and open connector leads to levels which are illegal.

D.2.3 Processing units

Global objective: To detect failures in processing units which lead to incorrect results.

D.2.3.1 Self-test by software

NOTE This technique/measure is referenced in Table D.4.

Aim: To detect, as early as possible, failures in the processing unit and other sub-elements consisting of physical storage (for example, registers) or functional units (for example, instruction decoder or an EDC coder/decoder), or both, by means of software.

Description: The failure detection is realised entirely by software which perform self-tests using a data pattern, or set of data patterns, to test the physical storage (for example, data and address registers) or the functional units (for example, the instruction decoder) or both.

EXAMPLE 1 The processing unit is tested for functional correctness by applying at least one pattern per instruction. Instructions not executed in the safety-related path can be omitted from the test but coverage can be limited as not all gates of the processing unit will be tested. In general, it is possible that not all dedicated and special purpose registers, core timers, and exceptions can be covered. Coverage for order dependencies, such as pipelines, or timing related fault modes can be limited. Determining the actual coverage of the tested gates (in contrast to covered instructions) typically requires extensive fault simulation. This test provides very limited or no coverage for soft errors.

EXAMPLE 2 In the case of sub-elements like an EDC coder/decoder, the software can read pre-written intentionally corrupted words to test the behaviour of the EDC logic. Corrupted words can also be written by the software test itself if the EDC and memory interface have an HW switch to access both data and code bits. Coverage depends on the amount and richness of patterns. This test provides no coverage for soft errors.

D.2.3.2 Self-test supported by hardware (one-channel)

NOTE This technique/measure is referenced in Table D.4.

Aim: To detect, as early as possible, failures in the processing unit and other sub-elements, using special hardware that increases the speed and extends the scope of failure detection.

Description: Additional special hardware facilities support self-test functions to detect failures in the processing unit and other sub-elements (for example an EDC coder/decoder) at a gate level. The test can achieve high coverage. Typically only run at the initialization or power-down of the processing unit due to its intrusive nature. Typical usage is for multiple-point fault detection.

EXAMPLE In the case of sub-elements like an EDC coder/decoder, a special HW mechanism, like a logic BIST, can be added to generate inputs to the coder-decoder and check for expected results. Typically inputs are generated by random pattern generators (e.g. MISR). Its coverage depends on the amount and richness of patterns – but usually the coverage is quite high due to the automatic pattern generation. This test provides no coverage for soft errors.

D.2.3.3 Self-test by software cross exchanged between two independent units

NOTE This technique/measure is referenced in Tables D.2 and D.4.

Aim: To detect, as early as possible, failures in the processing unit consisting of physical storage (for example registers) and functional units (for example, instruction decoder).

Description: The failure detection is realised entirely by means of two or more processing units each executing additional software functions which perform self-tests (for example walking-bit pattern) to test the physical storage (data and address registers) and the functional units (for example instruction decoder). The processing units exchange the results. This test provides very limited or no coverage for soft errors.

D.2.3.4 Software diversified redundancy (one hardware channel)

NOTE 1 This technique/measure is referenced in Table D.4.

Aim: To detect, as early as possible, failures in the processing unit, by dynamic software comparison.

Description: The design consists of two redundant diverse software implementations in one hardware channel. In some cases, using different hardware resources (e.g. different RAM, ROM memory ranges) can increase the diagnostic coverage.

One implementation, referred to as the primary path, is responsible for the calculations that if calculated erroneously can cause a hazard. The second implementation, referred to as the redundant path, is responsible for verifying the primary path's calculations and taking action if a failure is detected. Often the redundant path is implemented using separate algorithm designs and code to provide for software diversity. Once both paths are complete, a comparison of the output data of the two redundant software implementations is carried out. Detected differences lead to a failure message (see Figure D.2). The design includes methods to coordinate the two paths and to resynchronise the paths for transient errors.

Generally, the comparison involves some type of hysteresis and filtering to allow for minor differences due to the diverse software paths. Examples of algorithm diversity are: A+B=C versus C-B=A, and one path using normal calculations and the other path using two's complement mathematics. A redundant path can be as simple as a magnitude or rate-limit check on the calculation of the primary path.

NOTE 2 Due to the potential common cause failures between the primary and redundant paths, an additional watchdog processor can be used to verify the operation of the primary controller via a question and response diagnostic (see Reference [21]).

Another version of this safety mechanism is to implement the redundant path as an exact copy of the primary path (or to execute the primary path twice). This version, without software redundancy, only provides coverage for soft errors. Medium coverage can be achieved if the code is executed a third time with known inputs generating outputs to be verified versus a set of expected outputs. This technique results in a very easy pass-fail criterion (compared results are expected to agree exactly) and easy implementation (the redundant path does not need to be designed). However, since the same code is executed multiple times, the concept requires that history terms are preserved (e.g. dynamic states, integrators, rate-limits, etc.).

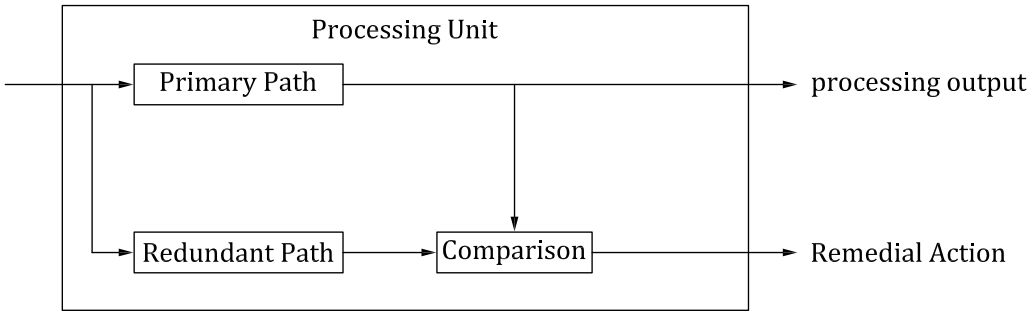


Figure D.2 — Redundant software comparison in same processing unit

D.2.3.5 Reciprocal comparison by software in separate processing units

NOTE This technique/measure is referenced in Table D.4.

Aim: To detect, as early as possible, failures in the processing unit, by dynamic software comparison.

Description: Two processing units exchange data (including results, intermediate results and test data) reciprocally. A comparison of the data is carried out using software in each unit and detected differences lead to a failure message (see Figure D.3). This approach allows for hardware and software diversity if different processor types are used as well as separate algorithm designs, code and compilers. The design includes methods to avoid false error detections due to differences between the processors (e.g. loop jitter, communication delays, processor initialization).

Paths can be implemented using separate cores of a dual core processor. In this case, the method includes analysis to understand common cause failures modes, due to the shared die and package, of the two cores.

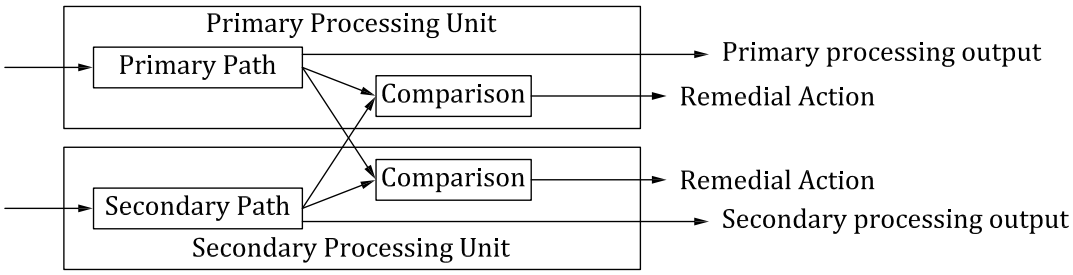


Figure D.3 — Redundant software comparison using different processing units

D.2.3.6 HW redundancy (e.g. Dual Core Lockstep, asymmetric redundancy, coded processing)

NOTE This technique/measure is referenced in Table D.4.

Aim: To detect, as early as possible, failures in the processing unit, by step-by-step comparison of internal or external results or both produced by two processing units operating in lockstep.

Description: In one version of this type of diagnostic technique, the Dual Core Lockstep, two symmetrical processing units are contained on one die (see Reference [22]). The processing units run duplicate operations in lockstep (or delayed by a fixed period) and the results are compared. Any mismatch results in an error condition and usually a reset condition. This is very effective for transient errors and for ALU type failures. Depending on the level of redundancy, coverage can be extended to the memory addressing lines and configuration registers. The technique has the advantage that no separate code for the parallel path is required but has the disadvantage of having two processing units providing only the performance of a single processing unit. In good designs, common cause failures

are understood and addressed (for example, common clock failure). This approach by itself does not provide coverage for systematic errors.

Other types of HW redundancies are possible, such as asymmetric redundancy. In those architectures (e.g. Reference [25]), a diverse and dedicated processing unit is tightly coupled with the main processing units by means of an interface enabling a step-by-step comparison of internal and external results. This is very effective for both a d.c. (direct current) fault model and for soft errors: moreover, the interface reduces complexity and shortens error detection latency, for example, for faults affecting the processing unit registers bank. No separate code is needed for the parallel path and the dedicated processing unit can be smaller than the main one. The hardware diversity provides effective coverage for common cause failures and systematic failures. The disadvantage of this approach is that it can require a detailed analysis to prove the diagnostic coverage.

Coded processing is also possible: processing units can be designed with special failure-recognising or failure correcting circuit techniques. These approaches can guarantee high coverage for very small processors with limited functionalities or they can be suitable for processor sub-units like ALU (e.g. Reference [26]). Hardware and software coding can be combined using approaches like the Vital Coded Processor (see Reference [27]). A detailed analysis can be needed to prove the diagnostic coverage.

D.2.3.7 Configuration register test

NOTE This technique/measure is referenced in Table D.4.

Aim: To detect, as early as possible, failures in the configuration resisters of a processing unit. Failures can be hardware related (stuck values or soft errors induced bit flips) or software related (incorrect value stored or register corrupted by software error).

Description: Configuration register settings are read and then compared to an encoding of the expected settings (e.g. a mask). If the settings do not match, the registers are reloaded with their intended value. If the error persists for a pre-determined number of checks, the fault condition is reported.

D.2.3.8 Stack over/under flow detection

NOTE This technique/measure is referenced in Table D.4.

Aim: To detect, as early as possible, stack over or under flows.

Description: The boundaries of the stack in volatile memory are loaded with predefined values. Periodically, the values are checked and if they have changed, an over or under flow is detected. A test is not needed if writes outside stack boundaries are controlled by a memory management unit.

D.2.3.9 Integrated hardware consistency monitoring

NOTE This technique/measure is referenced in Table D.4.

Aim: To detect, as early as possible, illegal conditions in the processing unit.

Description: Most processors are equipped with mechanisms that trigger hardware exceptions when errors are detected (division by zero and invalid op-codes, for example). Interrupt processing of these errors can then be used to trap these conditions to isolate the system from their effects. Typically, hardware monitoring is used to detect systematic failures but can also be used to detect certain kinds of random hardware faults. The technique provides low coverage for some coding errors and is good design practice.

D.2.4 I/O-units and interfaces

Global objective: To detect failures in input and output units (digital, analogue) and to prevent the sending of inadmissible outputs to the process.

D.2.4.1 Test pattern

NOTE This technique/measure is referenced in Tables D.5, D.9 and D.10.

Aim: To detect static failures (stuck-at failures) and cross-talk.

Description: This is a dataflow-independent cyclical test of input and output units. It uses a defined test pattern to compare observations with the corresponding expected values. Test coverage is dependent on the degree of independence between the test pattern information, the test pattern reception, and the test pattern evaluation. In a good design, the functional behaviour of the system is not unacceptably influenced by the test pattern.

D.2.4.2 Code protection

NOTE This technique/measure is referenced in Table D.5.

Aim: To detect random hardware and systematic failures in the input/output dataflow.

Description: This procedure protects the input and output information from both systematic and random hardware failures. Code protection gives dataflow-dependent failure detection of the input and output units, based on information redundancy, or time redundancy, or both. Typically, redundant information is superimposed on input data, or output data, or both. This gives a means to monitor the correct operation of the input or output circuits. Many techniques are possible, for example a carrier frequency signal can be superimposed on the output signal of a sensor and the logic unit can then check for the presence of the carrier frequency, or redundant code bits can be added to an output channel to allow the monitoring of the validity of a signal passing between the logic unit and final actuator.

D.2.4.3 Multi-channel parallel output

NOTE This technique/measure is referenced in Table D.5.

Aim: To detect random hardware failures (stuck-at failures), failures caused by external influences, timing failures, addressing failures, drift failures and transient failures.

Description: This is a dataflow-dependent multi-channel parallel output with independent outputs for the detection of random hardware failures. Failure detection is carried out via external comparators. If a failure occurs, the system can possibly be switched off directly. This measure is only effective if the dataflow changes during the diagnostic test interval.

D.2.4.4 Monitored outputs

NOTE This technique/measure is referenced in Table D.5.

Aim: To detect individual failures, failures caused by external influences, timing failures, addressing failures, drift failures (for analogue signals) and transient failures.

Description: This is a dataflow-dependent comparison of outputs with independent inputs to ensure compliance with a defined tolerance range (time, value). A detected failure cannot always be related to the defective output. This measure is only effective if the dataflow changes during the diagnostic test interval.

D.2.4.5 Input comparison/voting

NOTE This technique/measure is referenced in Tables D.5 and D.9.

Aim: To detect individual failures, failures caused by external influences, timing failures, addressing failures, drift failures (for analogue signals) and transient failures.

Description: This is a dataflow-dependent comparison of independent inputs to ensure compliance with a defined tolerance range (time, value). There will be 1 out of 2, 2 out of 3 or better redundancy. This measure is only effective if the dataflow changes during the diagnostic test interval.

D.2.5 Communication bus

Global objective: To detect failures in the information transfer.

D.2.5.1 One-bit hardware redundancy

NOTE This technique/measure is referenced in Table D.6.

Aim: To detect each odd-bit failure, i.e. 50 % of all the possible bit failures in the data stream.

Description: The communication bus is extended by one line (bit) and this additional line (bit) is used to detect failures by parity checking.

EXAMPLE Parity bit as implemented in a standard UART.

D.2.5.2 Multi-bit hardware redundancy

NOTE This technique/measure is referenced in Table D.6.

Aim: To detect failures during the communication on a bus and in serial transmission links.

Description: The communication bus is extended by two or more lines and these additional lines are used in order to detect failures by using block codes (e.g. Hamming code, Reed Solomon code, CRC, Low Density Parity Check code).

D.2.5.3 Complete hardware redundancy

NOTE This technique/measure is referenced in Table D.6.

Aim: To detect failures during the communication by comparing the signals on two buses.

Description: The bus is duplicated and the additional lines are used to detect failures.

EXAMPLE Dual channel FlexRay implementation: the bus is duplicated and the additional lines (bits) are used in order to detect failures.

D.2.5.4 Inspection using test patterns

NOTE This technique/measure is referenced in Table D.6.

Aim: To detect static failures (stuck-at failure) and cross-talk.

Description: This is a dataflow-independent cyclical test of data paths. It uses a defined test pattern to compare observations with the corresponding expected values.

Test coverage is dependent on the degree of independence between the test pattern information, the test pattern reception, and the test pattern evaluation. In a good design, the functional behaviour of the system is not unacceptably influenced by the test pattern.

D.2.5.5 Transmission redundancy

NOTE This technique/measure is referenced in Table D.6.

Aim: To detect transient failures in bus communication.

Description: The information is transferred several times in sequence. The technique is only effective in detecting transient failures.

D.2.5.6 Information redundancy

NOTE 1 This technique/measure is referenced in Table D.6.

Aim: To detect failures in bus communication.

Description: Data is transmitted in blocks, together with a calculated checksum or CRC (cyclic redundancy check) (see References [28] and [29]) for each block. The receiver then re-calculates the checksum of the received data and compares the result with the received checksum. For CRC coverage depends on the length of the data to be covered, the size of the CRC (number of bits) and the polynomial. The CRC can be designed to address the more probable communication failure modes of the underlying hardware (for example burst errors).

The message ID can be included in the checksum/CRC calculation to provide coverage for corruptions in this part of the message (masquerading).

a) Low overall coverage of failure modes in data transmission: Hamming distance of 2 or less.

EXAMPLE 1 CRC value for message information embedded in message; with a CRC size of 5 bits and a polynomial 0x12 results in a Hamming distance of 2 for a data length of less than 2 048 bits. The transmitter includes the CRC value mentioned and the receiver confirms the data after calculating and comparing the CRC value.

b) Medium overall coverage of failure modes in data transmission: Hamming distance of 3 or more.

EXAMPLE 2 CRC value for message information embedded in message; with a CRC size of 8 bits and a polynomial 0x97 results in a Hamming distance of 4 for a data length of less than 119 bits. The transmitter includes the CRC value mentioned and the receiver confirms the data after calculating and comparing the CRC value (typically used in a LIN bus).

EXAMPLE 3 CRC value for message information and message ID embedded in the message; with a CRC size of 10 bits and a polynomial 0x319 results in a Hamming distance of 4 for a data length of less than 501 bits. The transmitter includes the CRC value mentioned and the receiver confirms the data after calculating and comparing the CRC value.

EXAMPLE 4 CRC value for message information and message ID embedded in the message; with a CRC size of 15 bits and a polynomial 0x4599 results in a Hamming distance of 5 for a data length of less than 127 bits. As well, burst errors of length up to 15 can be detected. The transmitter includes the CRC value mentioned and the receiver confirms the data after calculating and comparing the CRC value (as used in CAN).

EXAMPLE 5 CRC value for message information embedded in the message, with a CRC size of 24 bits and a polynomial 0x5D6DCB results in a Hamming distance of the CRC of 6 for a data length of less than or equal to 248 bytes and a Hamming distance of the CRC of 4 for a data length of greater than 248 bytes. The transmitter includes the CRC value mentioned and the receiver confirms the data after calculating and comparing the CRC value (as used in FlexRay for the frame CRC).

EXAMPLE 6 CRC value for message header including the message ID embedded in the message; with a CRC size of 11 bits and a polynomial 0x385 results in a Hamming distance of 6 for a data length of less than or equal to 20 bits. The transmitter includes the CRC value mentioned and the receiver confirms the data after calculating and comparing the CRC value (as used in FlexRay for the header CRC).

NOTE 2 High coverage can be reached concerning data and ID corruption, however, overall high coverage cannot be reached by checking only the coherence of the data and the ID with a signature, whatever the efficiency of the signature. Specifically, a signature does not cover the message loss or the unintended message repetition.

NOTE 3 If a checksum algorithm has a Hamming distance of less than 3, a high coverage concerning data and ID corruption can still be claimed if supported by a proper rationale.

D.2.5.7 Frame counter

NOTE This technique/measure is referenced in Table D.6.

Aim: To detect frame losses. A frame is a coherent set of data sent from one controller to other controller(s). The unique frame is identified by a message ID.

Description: Each unique safety-related frame includes a counter as part of the message which is transmitted on the bus. The counter is incremented (with roll-over) during the creation of each successive frame transmitted. The receiver is then able to detect any frame loss or non-refreshment by verifying that the counter is incrementing by one.

A special version of the frame counter would be to include separate signal counters tied to the refreshment of safety-related data. In this situation, if a frame contained more than one piece of safety-related data, an individual counter for each piece of safety-related data is provided.

D.2.5.8 Timeout monitoring

NOTE This technique/measure is referenced in Table D.6.

Aim: To detect loss of data between the sending node and the receiving node.

Description: The receiver monitors each expected safety-related message ID for time between the receipt of valid frames with this message ID. A failure would be indicated by too long a period elapsing between messages. This is intended to detect continuous loss of a communications channel or the continuous loss of a specific message (no frames received for a specific message ID).

D.2.5.9 Read back of sent message

NOTE 1 This technique/measure is referenced in Table D.6.

Aim: To detect failures in bus communication.

Description: The transmitter reads back its sent message from the bus and compares it with the original message.

NOTE 2 This safety mechanism is used by CAN.

NOTE 3 High coverage can be reached concerning data and ID corruption, however, overall high coverage cannot be reached by checking only the coherence of the data and the ID. Other failure modes like the unintended message repetition are not necessarily covered by this safety mechanism.

D.2.6 Power supply

Global objective: To detect failures caused by a defect in the power supply.

D.2.6.1 Voltage or current control (input)

NOTE This technique/measure is referenced in Table D.7.

Aim: To detect as soon as possible wrong behaviour of input current or voltage values.

Description: Monitoring of input voltage or current.

D.2.6.2 Voltage or current control (output)

NOTE This technique/measure is referenced in Table D.7.

Aim: To detect as soon as possible wrong behaviour of output current or voltage values.

Description: Monitoring of output voltage or current.

D.2.7 Temporal and logical programme sequence monitoring

NOTE This group of techniques and measures is referenced in Table D.8.

Global objective: To detect a defective programme sequence. A defective programme sequence exists if the individual elements of a programme (for example, software modules, subprograms or commands) are processed in the wrong sequence or period of time, or if the clock of the processor is faulty.

D.2.7.1 Watchdog with separate time base without time-window

NOTE This technique/measure is referenced in Table D.8.

Aim: To monitor the behaviour and the plausibility of the programme sequence.

Description: External timing elements with a separate time base (for example, watchdog timers) are periodically triggered to monitor the processor's behaviour and the plausibility of the programme sequence. It is important that the triggering points are correctly placed in the programme. The watchdog is not triggered at a fixed period, but a maximum interval is specified.

D.2.7.2 Watchdog with separate time base and time-window

NOTE This technique/measure is referenced in Table D.8.

Aim: To monitor the behaviour and the plausibility of the programme sequence.

Description: External timing elements with a separate time base (for example watchdog timers) are periodically triggered to monitor the processor's behaviour and the plausibility of the programme sequence. It is important that the triggering points are correctly placed in the programme (e.g. not in an interrupt service routine). A lower and upper limit is given for the watchdog timer. If the programme sequence takes a longer or shorter time than expected, action is taken.

D.2.7.3 Logical monitoring of programme sequence

NOTE This technique/measure is referenced in Table D.8.

Aim: To monitor the correct sequence of the individual programme sections.

Description: The correct sequence of the individual programme sections is monitored using software (counting procedure, key procedure) or using external monitoring facilities (References [23,24]). It is important that the checking points are placed in the programme so that paths which can result in a hazard if they fail to complete or execute out of sequence, due to a single or multiple-point fault, are monitored. The sequences can be updated between each function call or more tightly integrated into the programme execution.

D.2.7.4 Combination of temporal and logical monitoring of programme sequences

NOTE This technique/measure is referenced in Table D.8.

Aim: To monitor the behaviour and the correct sequence of the individual programme sections.

Description: A temporal facility (for example a watchdog timer) monitoring the programme sequence is retriggered only if the sequence of the programme sections is also executed correctly. This is a combination of the technique in D.2.7.3 and either D.2.7.1 or D.2.7.2.

D.2.7.5 Combination of temporal and logical monitoring of programme sequences with time dependency

NOTE This technique/measure is referenced in Table D.8.

Aim: To monitor the behaviour, correct sequencing and the execution time interval of the individual programme sections.

Description: A Programme Flow Monitoring strategy is implemented where software update points are expected to occur within a relative time window. The PFM sequence result and time calculation are monitored by external monitoring facilities.

D.2.8 Sensors

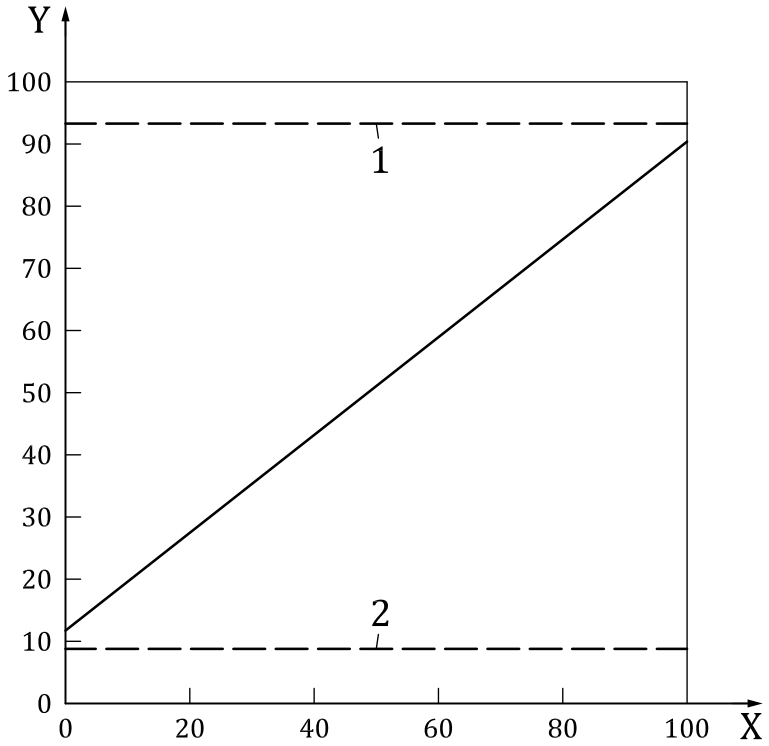
Global objective**:** To control failures in the sensors of the system.

D.2.8.1 Sensor valid range

NOTE This technique/measure is referenced in Table D.9.

Aim: To detect sensor shorts to ground or power and some open circuits.

Description: Limit valid reading to the middle part of the sensor electrical range (see Figure D.4 for example). If a sensor reading is in an invalid region, this indicates an electrical problem with the sensor such as a short to power or to ground. Typically used with sensors read by the ECU using ADCs.



Key

X physical sensor reading, in %

Y measured sensor reading, in % of reference voltage

1 out-of-range high

2 out-of-range low

Figure D.4 — Sensor with out-of-range region

D.2.8.2 Sensor correlation

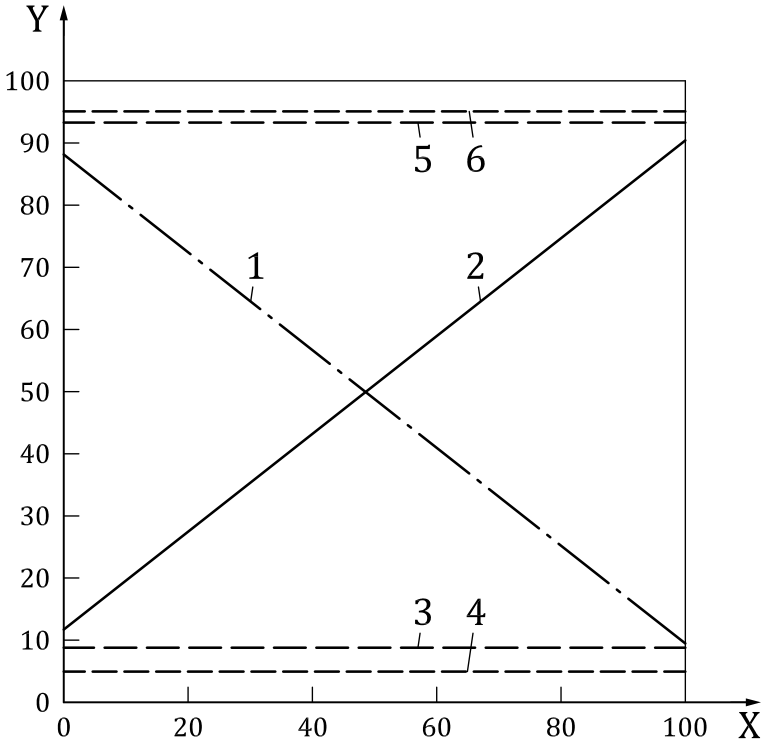
NOTE This technique/measure is referenced in Table D.9.

Aim: To detect sensor-in-range drifts, offsets or other errors using a redundant sensor.

Description: Comparison of two identical or similar sensors to detect in-range failures such as drifts, offsets or stuck-at failures. See Figure D.5 for an example with two equal but opposite slope sensors.

Note, that the out-of-range region is different for each sensor. Typically used with sensors read by the ECU using ADCs.

For the example of Figure D.5, sensors would be converted to equal slope and compared to agree within a threshold. The threshold is selected taking into account the ADC tolerance and the variation in the electrical elements. Both sensors are sampled by the ECU at as close to the same time as possible to avoid false failures due to the sensor readings dynamically changing.

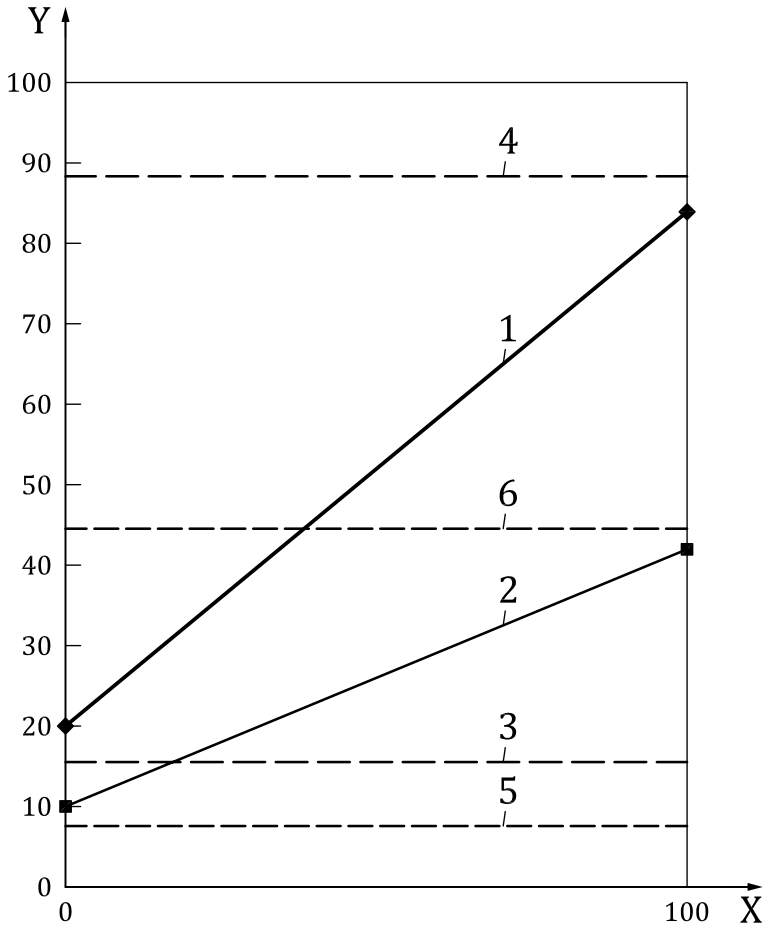


Key

X physical sensor reading, in %
Y measured sensor reading, in % of reference voltage
1 sensor 1
2 sensor 2
3 out-of-range sensor 1 low
4 out-of-range sensor 2 low
5 out-of-range sensor 1 high
6 out-of-range sensor 2 high

Figure D.5 — Equal and opposite slope sensors with out-of-range regions

Equal slope sensor based diagnostics do not detect situations where the two sensors are shorted together yielding correlated readings at the crossing point or common cause failures where a single component, e.g. the ADC, corrupts both sensor results in a similar way. An alternative design based on one full and one half slope sensor is given in Figure D.6.



Key

X physical sensor reading, in %
Y measured sensor reading, in % of reference voltage
1 sensor 1
2 sensor 2
3 out-of-range sensor 1 low
4 out-of-range sensor 1 high
5 out-of-range sensor 2 low
6 out-of-range sensor 2 high

Figure D.6 — One full and one half slope sensor with out-of-range regions

D.2.8.3 Sensor rationality check

NOTE This technique/measure is referenced in Table D.9.

Aim: To detect sensor-in-range drifts, offsets or other errors using multiple diverse sensors.

Description: Comparison of two (or more) sensors measuring different properties to detect in-range failures such as drifts, offsets or stuck-at failures. The sensor measurements are converted to equivalent values using a model to provide values that can be compared.

EXAMPLE The comparison of gasoline engine throttle position, manifold pressure and mass air flow sensors after each is converted to an air flow reading. The usage of diverse sensors reduces the problem of systematic faults.

D.2.9 Actuators

Global objective: To control failures in the final elements of the system.

D.2.9.1 Monitoring

NOTE 1 This technique/measure is referenced in Table D.10.

Aim: To detect the incorrect operation of an actuator.

Description: The operation of the actuator is monitored.

NOTE 2 Monitoring can be done at the actuator level by physical parameter measurements (which can have high coverage) but also at the system level regarding the actuator failure effect.

EXAMPLE 1 For a cooling radiator fan, monitoring at system level uses a temperature sensor to detect failure of the cooling radiator fan. Monitoring of physical parameters measures the voltage, or the current, or both, on the inputs of the cooling radiator fan.

EXAMPLE 2 Feedback control is used to move a throttle blade to a desired position. The actual position is measured and compared to the expected throttle position determined from the commanded throttle position and a model of the desired performance. If the two values differ from each other, after taking into account hysteresis, an error can be declared.

Annex E
(informative)

Example calculation of hardware architectural metrics:"single-point fault metric" and "latent-fault metric"

This annex gives an example of calculating the single-point fault metric and the latent-fault metric for each safety goal of the item as required in alternative a) of 8.4.7 and 8.4.8.

The system for this example realises two functions implemented on a single ECU.

Function 1 has one input (temperature measured via sensor R3) and one output (valve 2 controlled by I71) and its behaviour is to open valve 2 when the temperature is higher than 90 °C.

If no current flows through I71, valve 2 is open.

The associated safety goal 1 is "valve 2 shall not be closed for longer than 100 ms when the temperature is higher than 100 °C". The safety goal is assigned ASIL B. The safe state is: valve 2 open.

The value of sensor R3 is read by the microcontroller ADC. R3 resistance decreases as the temperature rises. There is no monitoring on this input. The output stage controlling T71 is monitored by the analogue input InADC1 (Safety mechanism SM1 in the tables). In this example, we will assume that the safety mechanism SM1 is able to detect failure modes of T71, which have the potential to directly violate the safety goal, with a 90 % coverage level. If a failure is detected by SM1, the safe state is activated but no lamp is switched on. Therefore, the diagnostic coverage with respect to latent faults for the failures modes detected by SM1 is claimed to be only 80 % (the driver will notice the failure through the functionality degradation).

Function 2 has two inputs (wheel speed measured via sensors I1 and I2 generating pulses) and one output (valve 1 controlled by I61) and its behaviour is to open valve 1 when the vehicle speed is higher than 90 km/h.

If no current flows through I61, valve 1 is open.

The associated safety goal 2 is "valve 1 shall not be closed for longer than 200 ms when the speed is higher than 100 km/h". The safety goal is assigned ASIL C. The safe state is: valve 1 open.

The values of sensors I1 and I2 pulses are read by the microcontroller. The wheel speed is computed using the mean value given by the sensors. The safety mechanism 2 (Safety Mechanism SM2 in the tables) compares both inputs. It detects the failures of each input with a 99 % diagnostic coverage. In the case of an inconsistency, Out.1 is set to 0. This opens valve1 (A "0" voltage on transistor opens the gate. A "0" voltage on I61 opens valve 1). Therefore, 99 % of the faults that have the potential to violate the safety goal are detected and lead to the safe state. When the safe state is activated, the lamp L1 is on. Therefore, these faults are 100 % perceived. The remaining 1 % of faults are residual faults and not latent faults.

The output stage controlling T61 is monitored by the analogue input In ADC2 (Safety mechanism SM3 in the tables).

The microcontroller has no internal redundancy. In this example, a ratio of 50 % of safe faults is assumed. A global coverage of 90 % with respect to the violation of the safety goal, through internal self-tests and the external watchdog (Safety Mechanism SM4 in the tables) is also assumed. The watchdog gets a live signal via the output 0 of the microcontroller. When the watchdog is no longer refreshed, its output goes low. A fault detection by SM4 (watchdog and microcontroller self-tests) switches both functions to their safe state and switches L1 on. Therefore the diagnostic coverage with respect to the latent faults is claimed to be 100 %.

L1 is an LED on the dashboard, it is lit upon detection of a multiple-point failure, of which only a proportion can be detected, and indicates to the driver that the safe state of function 2 (valve 1 open) has been activated.

NOTE 1 The harness failures are not considered in this example.

NOTE 2 The fault model used for a given electronic part can differ depending on the application.

EXAMPLE 1 The fault model of a resistor depends if the hardware part is used in a digital input (such as R11, R12, R13...) or an analogue input (such as R3). In the first case the fault model can be “open/closed” whereas in the second case it can be “open/closed/drift”.

NOTE 3 The first metric only uses the failure mode coverage of the safety mechanisms that aim at preventing the violation of the safety goal. The second metric only uses the failure mode coverage of the safety mechanisms that aim at preventing the failure mode from being latent.

EXAMPLE 2 The failure mode “open” of R21 has the potential to violate safety goal 2 in the absence of a safety mechanism. Safety mechanism 2 detects this failure mode with a failure mode coverage of 99 % and switches the system into a safe state. When detecting this failure mode, an alert is displayed; the failure mode coverage with respect to latent failures is 100 %.

NOTE 4 In this example, assumptions on the failure mode distribution of the hardware elements have been considered. If no particular failure mode distribution can be argued or referenced, an equal distribution of the failure modes can be assumed.

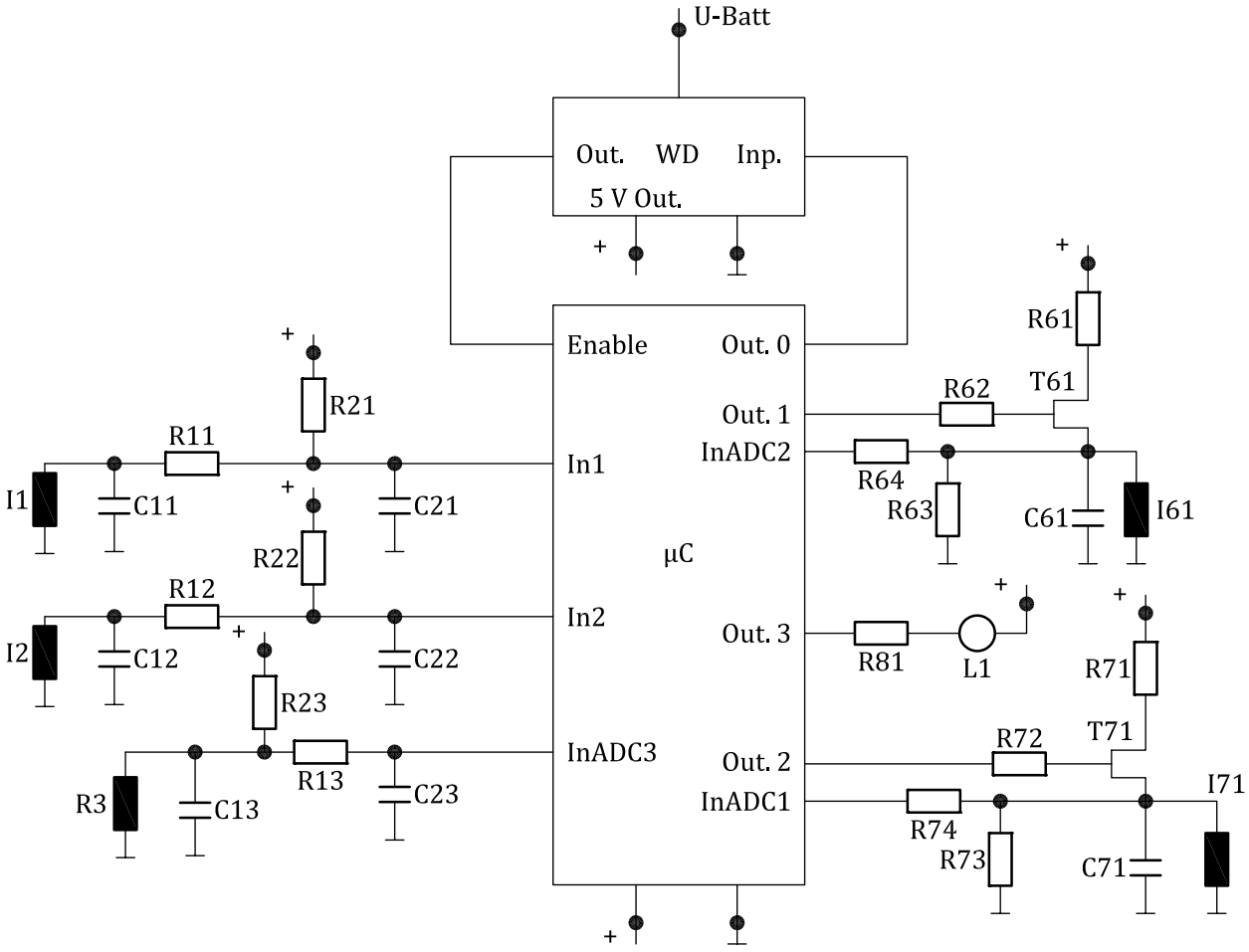


Figure E.1 — Example diagram

Table E.1 — Safety goal 1

Component Name	Failure rate/ FIT	Safety-related component to be considered in the calculations?	Failure Mode	Failure rate distribution	Failure mode that has the potential to violate the safety goal in absence of safety mechanisms?	Safety mechanism(s) allowing to prevent the failure mode from violating the safety goal?	Failure mode coverage wrt. violation of safety goal	Residual or Single-Point Fault failure rate/FIT	Failure mode that may lead to the violation of safety goal in combination with an independent failure of another component?	Detection means? Safety mechanism(s) allowing to prevent the failure mode from being latent?	Failure mode coverage with respect to latent failures	Latent Multiple-Point Fault failure rate/FIT
R3 NOTE 1	3	YES	open	30 %	X	none	0 %	0,9				
			closed	10 %								
			drift 0,5	30 %								
			drift 2	30 %	X		0 %	0,9				
R13 NOTE 1, NOTE 2 and NOTE 6	2	YES	open	90 %	X	none	0 %	1,8				
			closed	10 %	X		0 %	0,2				
R23 NOTE 1	2	YES	open	90 %		none						
			closed	10 %	X		0 %	0,2				
C13 NOTE 3 and NOTE 6	2	YES	open	20 %	X	none	0 %	0,4				
			closed	80 %								
C23 NOTE 4	2	NO	open	20 %								
			closed	80 %								
WD	20	YES	Out. Stuck at 1	50 %					X	none	0 %	10
			Out. Stuck at 0	50 %								
T71 NOTE 5	5	YES	open circuit	50 %		SM1				SM1		
			short circuit	50 %	X		90 %	0,25	X		80 %	0,45

Table E.1 *(continued)*

Component Name	Failure rate/ FIT	Safety-related component to be considered in the calculations?	Failure Mode	Failure rate distribution	Failure mode that has the potential to violate the safety goal in absence of safety mechanisms?	Safety mechanism(s) allowing to prevent the failure mode from violating the safety goal?	Failure mode coverage wrt. violation of safety goal	Residual or Single-Point Fault failure rate/FIT	Failure mode that may lead to the violation of safety goal in combination with an independent failure of another component?	Detection means? Safety mechanism(s) allowing to prevent the failure mode from being latent?	Failure mode coverage with respect to latent failures	Latent Multiple-Point Fault failure rate/FIT
R71	2	YES	open	90 %						none		
NOTE 2 and NOTE 6			closed	10 %					X		0 %	0,2
R72	2	YES	open	90 %						none		
NOTE 2 and NOTE 6			closed	10 %					X		0 %	0,2
R73	2	NO	open	90 %								
NOTE 4			closed	10 %								
R74	2	YES	open	90 %					X	none	0 %	1,8
NOTE 2 and NOTE 6			closed	10 %					X		0 %	0,2
I71	5	NO	open	80 %								
NOTE 4			closed	20 %								
C71	2	YES	open	20 %					X	none		0,4
NOTE 3			closed	80 %								
R81	2	NO	open	90 %								
NOTE 4			closed	10 %								
L1	10	NO	open	90 %								
NOTE 4			closed	10 %								
µC	100	YES	All	50 %	X	SM4	90 %	5	X	SM4	100 %	0
			All	50 %								
							∑ 9,65				∑	13,25

Table E.1 *(continued)*

Component Name	Failure rate/ FIT	Safety-related component to be considered in the calculations?	Failure Mode	Failure rate distribution	Failure mode that has the potential to violate the safety goal in absence of safety mechanisms?	Safety mechanism(s) allowing to prevent the failure mode from violating the safety goal?	Failure mode coverage wrt. violation of safety goal	Residual or Single-Point Fault failure rate/FIT	Failure mode that may lead to the violation of safety goal in combination with an independent failure of another component?	Detection means? Safety mechanism(s) allowing to prevent the failure mode from being latent?	Failure mode coverage with respect to latent failures	Latent Multiple-Point Fault failure rate/FIT
Total failure rate		163 FIT		Single-Point Fault Metric = 1-(9,65/142) = 93,2 %					Latent Fault Metric = 1-(13,25/(142-9,65)) = 90,0 %			
Total Safety-Related		142 FIT										
Total Non Safety-Related		21 FIT										

Safety goal 1 is assigned ASIL B, which has, if Table 4 is used, a single-point fault metric recommendation of ≥ 90 %, and, if Table 5 is used, a latent-fault metric recommendation of ≥ 60 %. The single-point fault metric recommendation is satisfied by the calculated metric of 93,2 % and the latent-fault metric recommendation is satisfied by the value of 90 %.

NOTE 1 The failure modes "open" on R3 and R13 and "closed" on R23 are single-point faults. They lead directly to the violation of the safety goal and no safety mechanism covers faults of these hardware parts.

NOTE 2 The purpose of this hardware part is electrical protection. The closed failure mode means loss of protection.

NOTE 3 The purpose of this hardware part is ESD protection. The open failure mode means loss of protection.

NOTE 4 The elements with failures that do not have the potential to significantly contribute to the violation of the safety goal, i.e. with only safe failure modes, are not considered in the calculations in order to be more conservative. E.g. here, L1 and R81 are elements which implement a safety mechanism to prevent dual-point faults from being latent. The multiple-point faults of order n, with n > 2, are considered to be safe faults.

NOTE 5 The faults that lead directly to the violation of the safety goal (single-point faults or residual faults) cannot contribute anymore to the latent faults population. Therefore, for instance, the failure rate of the latent failure mode "closed gate" of T71 is computed as follows:

$$\lambda_{\mathrm{MPF,L}} = [(\lambda_{\mathrm{T71}} \times \mathit{FailureModeDistrib}_{\mathrm{closed\ gate}}) - \lambda_{\mathrm{T71,RF}}] \times (1 - \mathit{FMC}_{\mathrm{Latent\ Faults}})$$

$$\lambda_{\mathrm{MPF,L}} = [(5 \times 0{,}5) - 0{,}25] \times (1 - 0{,}8) = 0{,}45$$

NOTE 6 The classification of the failure modes leading to the loss of ESD or electrical protection is based on a case-by-case analysis and takes into consideration the likelihood of the ESD or electrical stress and the characterized effects of the ESD or electrical stress with respect to the safety goal. If for example the ESD event is likely to occur during the vehicle lifetime and its effects can lead to the violation of the safety goal in the absence of the given protection, then the failure mode leading the loss of the protection is classified as a single-point fault. This annex is an example on how to handle those cases within the metrics. In practice ESD or EMI stresses do not have this impact on typical designs similar to that of the example.

Table E.2 — Safety goal 2

Component Name	Failure rate/ FIT	Safety-related component to be considered in the calculation?	Failure Mode	Failure rate distribution	Failure mode that has the potential to violate the safety goal in absence of safety mechanisms?	Safety mechanism(s) allowing to prevent the failure mode from violating the safety goal?	Failure mode coverage wrt. violation of safety goal	Residual or Single-Point Fault failure rate/ FIT	Failure mode that may lead to the violation of safety goal in combination with an independent failure of another component?	Detection means? Safety mechanism(s) allowing to prevent the failure mode from being latent?	Failure mode coverage wrt. Latent failures	Latent Multiple-Point Fault failure rate/FIT
R11	2	YES	open	90 %	X	SM2	99 %	0,018	X	SM2	100 %	0
NOTE 1, NOTE 6 and NOTE 7			closed	10 %	X		99 %	0,002	X		100 %	0
R12	2	YES	open	90 %	X	SM2	99 %	0,018	X	SM2	100 %	0
NOTE 1, NOTE 6 and NOTE 7			closed	10 %	X		99 %	0,002	X		100 %	0
R21	2	YES	open	90 %	X	SM2	99 %	0,018	X	SM2	100 %	0
NOTE 2			closed	10 %	X		99 %	0,002	X		100 %	0
R22	2	YES	open	90 %	X	SM2	99 %	0,018	X	SM2	100 %	0
NOTE 2			closed	10 %	X		99 %	0,002	X		100 %	0
C11	2	YES	open	20 %	X	SM2	99 %	0,004	X	SM2	100 %	0
NOTE 1, NOTE 6 and NOTE 7			closed	80 %	X		99 %	0,016	X		100 %	0
C12	2	YES	open	20 %	X	SM2	99 %	0,004	X	SM2	100 %	0
NOTE 1, NOTE 6 and NOTE 7			closed	80 %	X		99 %	0,016	X		100 %	0
C21	2	YES	open	20 %		SM2				SM2		
			closed	80 %	X		99 %	0,016	X		100 %	0

Table E.2 *(continued)*

Component Name	Failure rate/ FIT	Safety-related component to be considered in the calculation?	Failure Mode	Failure rate distribution	Failure mode that has the potential to violate the safety goal in absence of safety mechanisms?	Safety mechanism(s) allowing to prevent the failure mode from violating the safety goal?	Failure mode coverage wrt. violation of safety goal	Residual or Single-Point Fault failure rate/ FIT	Failure mode that may lead to the violation of safety goal in combination with an independent failure of another component?	Detection means? Safety mechanism(s) allowing to prevent the failure mode from being latent?	Failure mode coverage wrt. Latent failures	Latent Multiple-Point Fault failure rate/FIT
C22	2	YES	open	20 %		SM2				SM2		
			closed	80 %	X		99 %	0,016	X		100 %	0
I1	4	YES	open	70 %	X	SM2	99 %	0,028	X	SM2	100 %	0
			closed	20 %	X		99 %	0,008	X		100 %	0
			drift 0,5	5 %	X		99 %	0,002	X		100 %	0
			drift 2	5 %								
I2	4	YES	open	70 %	X	SM2	99 %	0,028	X	SM2	100 %	0
			closed	20 %	X		99 %	0,008	X		100 %	0
			drift 0,5	5 %	X		99 %	0,002	X		100 %	0
			drift 2	5 %								
WD	20	YES	Out. Stuck at 1	50 %					X	none	0 %	10
			Out. Stuck at 0	50 %								
T61	5	YES	open circuit	50 %		SM3				SM3		
			short circuit	50 %	X		90 %	0,25	X		100 %	0

Table E.2 *(continued)*

Component Name	Failure rate/ FIT	Safety-related component to be considered in the calculation?	Failure Mode	Failure rate distribution	Failure mode that has the potential to violate the safety goal in absence of safety mechanisms?	Safety mechanism(s) allowing to prevent the failure mode from violating the safety goal?	Failure mode coverage wrt. violation of safety goal	Residual or Single-Point Fault failure rate/ FIT	Failure mode that may lead to the violation of safety goal in combination with an independent failure of another component?	Detection means? Safety mechanism(s) allowing to prevent the failure mode from being latent?	Failure mode coverage wrt. Latent failures	Latent Multiple-Point Fault failure rate/FIT
R61	2	YES	open	90 %						none		
NOTE 3 and NOTE 6			closed	10 %					X		0 %	0,2
R62	2	YES	open	90 %						none		
NOTE 3 and NOTE 6			closed	10 %					X		0 %	0,2
R63	2	NO	open	90 %								
NOTE 5			closed	10 %								
R64	2	YES	open	90 %					X	none	0 %	1,8
NOTE 1 and NOTE 6			closed	10 %					X		0 %	0,2
I61	5	NO	open	80 %								
NOTE 5			closed	20 %								
C61	2	YES	open	20 %					X	none	0 %	0,4
NOTE 4 and NOTE 6			closed	80 %								
R81	2	NO	open	90 %								
NOTE 5			closed	10 %								
L1	10	NO	open	90 %								
NOTE 5			closed	10 %								

Table E.2 *(continued)*

Component Name	Failure rate/ FIT	Safety-related component to be considered in the calculation?	Failure Mode	Failure rate distribution	Failure mode that has the potential to violate the safety goal in absence of safety mechanisms?	Safety mechanism(s) allowing to prevent the failure mode from violating the safety goal?	Failure mode coverage wrt. violation of safety goal	Residual or Single-Point Fault failure rate/ FIT	Failure mode that may lead to the violation of safety goal in combination with an independent failure of another component?	Detection means? Safety mechanism(s) allowing to prevent the failure mode from being latent?	Failure mode coverage wrt. Latent failures	Latent Multiple-Point Fault failure rate/FIT
µC	100	YES	All	50 %	X	SM4	90 %	5	X	SM4	100 %	0
			All	50 %								
							∑	5,48			∑	12,80

Total failure rate	176 FIT	Single-Point Fault Metric = 1-(5,48/157) = 96,5 %	Latent Fault Metric = 1-(12,8/(157-5,48)) = 91,6 %
Total Safety-Related	157 FIT		
Total Non Safety-Related	19 FIT		

Safety goal 2 is assigned ASIL C, which has, if Table 4 is used, a single-point fault metric requirement of ≥ 97 %, and, if Table 5 is used, a latent-fault metric recommendation of ≥ 80 %. The single-point fault metric requirement is not satisfied by the calculated metric of 96,5 % and the latent-fault metric recommendation is satisfied by the value of 91,6 %.

NOTE 1 The purpose of this hardware part is electrical protection. One failure mode is the loss of electrical protection. The other mode has the potential to violate the safety goal in absence of safety mechanisms.

NOTE 2 Both failure modes have the potential to violate the safety goal in absence of safety mechanisms as in both cases, no speed pulses are transmitted. This leads to a wrong speed acquisition. The sensor is an open-collector sensor.

NOTE 3 The purpose of this hardware part is electrical protection. The close failure mode means loss of protection.

NOTE 4 The purpose of this hardware part is ESD protection. The open failure mode means loss of protection.

NOTE 5 The elements with failures that do not have the potential to significantly contribute to the violation of the safety goal, i.e. with only safe failure modes, are not considered in the calculations in order to be more conservative. E.g. here, L1 and R81 are elements which implement a safety mechanism to prevent dual-point faults from being latent. The multiple-point faults of order n, with $n > 2$, are considered to be safe faults.

NOTE 6 The classification of the failure modes leading to the loss of ESD or electrical protection is based on a case-by-case analysis and takes into consideration the likelihood of the ESD or electrical stress and the characterized effects of the ESD or electrical stress with respect to the safety goal. If, for example, the ESD event is likely to occur during the vehicle lifetime, and its effects can lead to the violation of the safety goal in the absence of the given protection, then the failure mode leading the loss of the protection is classified as a single-point fault. This annex is an example on how to handle those cases within the metrics. In practice ESD or EMI stresses do not have this impact on typical designs similar to that of the example. Moreover it is considered here that SM4 does not cover these failure modes even if they can lead to some damage to the microcontroller.

NOTE 7 The loss of electrical protection will cause a wrong input value and will be detected by SM2 and therefore will not be latent.

Annex F
(informative)

Example for rationale that objectives of Clause 9 in accordance with 4.2 are met

F.1 General

Within this annex, an example is given on how a hardware design can be evaluated based on the results of the safety analysis, to argue that the objectives of Clause 9 are met. The example is based on the hardware design and the analysis of Annex E for the safety goal 2 (Valve 1 shall not be closed for longer than 200 ms when the speed is higher than 100 km/h [ASIL C]). The evaluation is done in following steps:

1. Safety analysis providing the PMHF evaluation.
2. Definition of selection criteria of faults or failure modes to be evaluated.
3. Application of the selection criteria.
4. Evaluation regarding the compliance with the objectives of Clause 9

F.2 Safety analysis providing the PMHF evaluation

The starting point is the safety analysis which has been done in Annex E. In addition to the evaluations done in Annex E, the PMHF value is approximated using the formula $PMHF_{est} = \lambda_{SPF} + \lambda_{RF} + \lambda_{DPF\_det} \times \lambda_{DPF\_latent} \times T_{Lifetime}$ (for more details on how to estimate a PMHF value see clause concerning the calculation of PMHF of ISO 26262-10:2018) and for each failure mode the contribution to the overall PMHF value in percent is calculated.

Table F.1 — Quantitative FMEA of Annex E for safety goal 2

Component Name	Failure rate/ FIT	Safety-related component to be considered in the calculation?	Failure Mode	Failure rate distribution	Failure mode that has the potential to violate the safety goal in absence of safety mechanisms?	Safety mechanism(s) allowing to prevent the failure mode from violating the safety goal?	Failure mode coverage wrt. violation of safety goal		Failure mode that may lead to the violation of safety goal in combination with an independent failure of another component?	Detection means? Safety mechanism(s) allowing to prevent the failure mode from being latent?	Failure mode coverage wrt. Latent failures		DPF_det	PMHF [%]
R11 NOTE 1, NOTE 6 and NOTE 7	2	YES	open	90 %	X	SM2	99 %	0,018	X	SM2	100 %	0	1,782	0,3 %
			closed	10 %	X		99 %	0,002	X		100 %	0	0,198	0,0 %
R12 NOTE 1, NOTE 6 and NOTE 7	2	YES	open	90 %	X	SM2	99 %	0,018	X	SM2	100 %	0	1,782	0,3 %
			closed	10 %	X		99 %	0,002	X		100 %	0	0,198	0,0 %
R21 NOTE 2	2	YES	open	90 %	X	SM2	99 %	0,018	X	SM2	100 %	0	1,782	0,3 %
			closed	10 %	X		99 %	0,002	X		100 %	0	0,198	0,0 %
R22 NOTE 2	2	YES	open	90 %	X	SM2	99 %	0,018	X	SM2	100 %	0	1,782	0,3 %
			closed	10 %	X		99 %	0,002	X		100 %	0	0,198	0,0 %
C11 NOTE 1, NOTE 6 and NOTE 7	2	YES	open	20 %	X	SM2	99 %	0,004	X	SM2	100 %	0	0,396	0,1 %
			closed	80 %	X		99 %	0,016	X		100 %	0	1,584	0,3 %

Table F.1 *(continued)*

Component Name	Failure rate/ FIT	Safety-related component to be considered in the calculation?	Failure Mode	Failure rate distribution	Failure mode that has the potential to violate the safety goal in absence of safety mechanisms?	Safety mechanism(s) allowing to prevent the failure mode from violating the safety goal?	Failure mode coverage wrt. violation of safety goal		Failure mode that may lead to the violation of safety goal in combination with an independent failure of another component?	Detection means? Safety mechanism(s) allowing to prevent the failure mode from being latent?	Failure mode coverage wrt. Latent failures		DPF_det	PM-HF [%]
C12 NOTE 1, NOTE 6 and NOTE 7	2	YES	open	20 %	X	SM2	99 %	0,004	X	SM2	100 %	0	0,396	0,1 %
			closed	80 %	X		99 %	0,016	X		100 %	0	1,584	0,3 %
C21	2	YES	open	20 %		SM2				SM2			0	0,0 %
			closed	80 %	X		99 %	0,016	X		100 %	0	1,584	0,3 %
C22	2	YES	open	20 %		SM2				SM2			0	0,0 %
			closed	80 %	X		99 %	0,016	X		100 %	0	1,584	0,3 %
I1	4	YES	open	70 %	X	SM2	99 %	0,028	X	SM2	100 %	0	2,772	0,5 %
			closed	20 %	X		99 %	0,008	X		100 %	0	0,792	0,1 %
			drift 0,5	5 %	X		99 %	0,002	X		100 %	0	0,198	0,0 %
			drift 2	5 %									0	0,0 %
I2	4	YES	open	70 %	X	SM2	99 %	0,028	X	SM2	100 %	0	2,772	0,5 %
			closed	20 %	X		99 %	0,008	X		100 %	0	0,792	0,1 %
			drift 0,5	5 %	X		99 %	0,002	X		100 %	0	0,198	0,0 %
			drift 2	5 %									0	0,0 %

Table F.1 *(continued)*

Component Name	Failure rate/ FIT	Safety-related component to be considered in the calculation?	Failure Mode	Failure rate distribution	Failure mode that has the potential to violate the safety goal in absence of safety mechanisms?	Safety mechanism(s) allowing to prevent the failure mode from violating the safety goal?	Failure mode coverage wrt. violation of safety goal		Failure mode that may lead to the violation of safety goal in combination with an independent failure of another component?	Detection means? Safety mechanism(s) allowing to prevent the failure mode from being latent?	Failure mode coverage wrt. Latent failures		DPF_det	PMHF [%]
WD	20	YES	Out. Stuck at 1	50 %					X	none	0 %	10	0	0,0 %
			Out. Stuck at 0	50 %									0	0,0 %
T61	5	YES	open circuit	50 %		SM3				SM3			0	0,0 %
			short circuit	50 %	X		90 %	0,25	X		100 %	0	2,25	4,6 %
R61	2	YES	open	90 %						none			0	0,0 %
NOTE 3 and NOTE 6			closed	10 %					X		0 %	0,2	0	0,0 %
R62	2	YES	open	90 %						none			0	0,0 %
NOTE 3 and NOTE 6			closed	10 %					X		0 %	0,2	0	0,0 %
R63	2	NO	open	90 %									0	0,0 %
NOTE 5			closed	10 %									0	0,0 %

Table F.1 *(continued)*

Component Name	Failure rate/ FIT	Safety-related component to be considered in the calculation?	Failure Mode	Failure rate distribution	Failure mode that has the potential to violate the safety goal in absence of safety mechanisms?	Safety mechanism(s) allowing to prevent the failure mode from violating the safety goal?	Failure mode coverage wrt. violation of safety goal		Failure mode that may lead to the violation of safety goal in combination with an independent failure of another component?	Detection means? Safety mechanism(s) allowing to prevent the failure mode from being latent?	Failure mode coverage wrt. Latent failures		DPF_det	PMHF [%]
R64	2	YES	open	90 %					X	none	0 %	1,8	0	0,0 %
NOTE 1 and NOTE 6			closed	10 %					X		0 %	0,2	0	0,0 %
I61	5	NO	open	80 %									0	0,0 %
NOTE 5			closed	20 %									0	0,0 %
C61	2	YES	open	20 %					X	none	0 %	0,4	0	0,0 %
NOTE 4 and NOTE 6			closed	80 %									0	0,0 %
R81	2	NO	open	90 %									0	0,0 %
NOTE 5			closed	10 %									0	0,0 %
L1	10	NO	open	90 %									0	0,0 %
NOTE 5			closed	10 %									0	0,0 %
µC	100	YES	All	50 %	X	SM4	90 %	5	X	SM4	100 %	0	45	
			All	50 %									0	0,0 %
							∑	5,48			∑	12,80	69,822	

Table F.2 — Results of Quantitative FMEA

			Lifetime	10 000 h
Total Safety-Related	157 FIT	PMHF_DPF	0,009 FIT	0,16 %
Total Non Safety-Related	19 FIT	PMHF_RF	5,48 FIT	99,84 %
Total failure rate	176 FIT	PMHF	5,489 FIT	100,00 %
Single-Point Fault Metric = 1-(5,48/157) = 96,5 %				
Latent Fault Metric = 1-(12,8/(157-5,48)) = 91,6 %				

F.3 Definition of selection criteria of faults or failure modes to be evaluated

A safety analysis of a complex system can be very large. The principle behind of choosing selection criteria is to focus the evaluation to the relevant points. In this example, following selection criteria have been chosen:

1. All faults or failure modes with a FMC with regard to residual or single-point faults ≤ 90 %.
2. All faults or failure modes with a contribution ≥ 2 % to the overall PMHF value.
3. The top 20 faults or failure modes contributing to the overall PMHF value.

NOTE The selection criteria are defined based on the safety analysis results (e.g. deviation from target value, dispersion of contributors).

F.4 Application of the selection criteria

Table F.3 — List of all faults or failure modes with a FMC with regard to residual or single-point faults ≤90 %

ID	Component Name	Failure rate/ FIT	Safety-related component to be considered in the calculation?	Failure Mode	Failure rate distribution	Safety mechanism(s) allowing to prevent the failure mode from violating the safety goal	Failure mode coverage wrt violation of safety goal	Contribution to PMHF [FIT]	Contribution to PMHF [%]
45	µC	100	YES	All	0,5	SM4	90,00 %	5	91,13 %
28	T61	5	YES	short circuit	0,5	SM3	90,00 %	0,25	4,56 %
					Overall PMHF contribution [FIT] & [%]			5,25	95,68 %

Table F.4 — List of all faults or failure modes with a contribution ≥ 2 % to the overall PMHF value

ID	Component Name	Failure rate/ FIT	Safety-related component to be considered in the calculation?	Failure Mode	Failure rate distribution	Safety mechanism(s) allowing to prevent the failure mode from violating the safety goal	Failure mode coverage wrt violation of safety goal	Contribution to PMHF [FIT]	Contribution to PMHF [%]
45	µC	100	YES	All	0,5	SM4	90,00 %	5	91,13 %
28	T61	5	YES	short circuit	0,5	SM3	90,00 %	0,25	4,56 %
					Overall PMHF contribution [FIT] & [%]			5,25	95,68 %

Table F.5 — List of top 20 faults or failure modes contributing to the overall PMHF value

PMHF contribution	ID	Component Name	Failure rate/ FIT	Safety-related component to be considered in the calculation?	Failure Mode	Failure rate distribution	Safety mechanism(s) allowing to prevent the failure mode from violating the safety goal	Failure mode coverage wrt violation of safety goal	Contribution to PMHF [FIT]	Contribution to PMHF [%]
1	45	µC	100	YES	All	0,5	SM4	90,00 %	5	91,13 %
2	28	T61	5	YES	short circuit	0,5	SM3	90,00 %	0,25	4,56 %
3	21	I2	4	YES	open	0,7	SM2	99,00 %	0,028	0,51 %
4	17	I1	4	YES	open	0,7	SM2	99,00 %	0,028	0,51 %
5	1	R11	2	YES	open	0,9	SM2	99,00 %	0,018	0,33 %
6	3	R12	2	YES	open	0,9	SM2	99,00 %	0,018	0,33 %
7	5	R21	2	YES	open	0,9	SM2	99,00 %	0,018	0,33 %
8	7	R22	2	YES	open	0,9	SM2	99,00 %	0,018	0,33 %
9	10	C11	2	YES	closed	0,8	SM2	99,00 %	0,016	0,29 %
10	16	C22	2	YES	closed	0,8	SM2	99,00 %	0,016	0,29 %
11	14	C21	2	YES	closed	0,8	SM2	99,00 %	0,016	0,29 %
12	12	C12	2	YES	closed	0,8	SM2	99,00 %	0,016	0,29 %
13		PMHF_DPF							0,008 94	0,16 %
14	22	I2	4	YES	closed	0,2	SM2	99,00 %	0,008	0,15 %
15	18	I1	4	YES	closed	0,2	SM2	99,00 %	0,008	0,15 %
16	9	C11	2	YES	open	0,2	SM2	99,00 %	0,004	0,07 %
17	11	C12	2	YES	open	0,2	SM2	99,00 %	0,004	0,07 %

Table F.5 *(continued)*

PMHF contribution	ID	Component Name	Failure rate/ FIT	Safety-related component to be considered in the calculation?	Failure Mode	Failure rate distribution	Safety mechanism(s) allowing to prevent the failure mode from violating the safety goal	Failure mode coverage wrt violation of safety goal	Contribution to PMHF [FIT]	Contribution to PMHF [%]
18	23	I2	4	YES	drift 0,5	0,05	SM2	99,00 %	0,002	0,04 %
19	8	R22	2	YES	closed	0,1	SM2	99,00 %	0,002	0,04 %
20	19	I1	4	YES	drift 0,5	0,05	SM2	99,00 %	0,002	0,04 %
						Overall PMHF contribution [FIT] & [%]			5,48	99,89 %

NOTE In Table F.5, PMHF_DPF denotes the contribution of dual point failures to the calculated PMHF value, as given in Table F.2.

F.5 Evaluation regarding the compliance with the objectives of Clause 9

Since the faults or failure modes identified in Table F.3 and in Table F.4 contribute more than 95 % of the PMHF value, it is sufficient to restrict the evaluation to these criteria. In order to assess the compliance with the objectives of Clause 9 following topics can be taken into consideration:

— How do well-trusted systems already in use treat these issues?

— What is the state of the art to treat these issues?

— What is the field experience regarding the hardware elements and their faults or failure modes under consideration?

— What are the results from reliability evaluations regarding the hardware elements and their faults or failure modes under consideration?

— Are dedicated measures (see 9.4.1.2 and 9.4.1.3) in place in order to reduce the risk of occurrence in the field?

Annex G
(informative)

Example of a PMHF budget assignment for an item consisting of two systems

G.1 Objectives

In this example, a procedure is given for PMHF budgeting across two systems which both contribute to the same safety goal

NOTE The example has been exaggerated in order to highlight the pitfalls of certain procedures.

G.2 Item architecture

The item consists of two systems, system A and system B (see Figure G.1) which are connected to each other via the vehicle network bus (e.g. CAN, FlexRay or Ethernet).

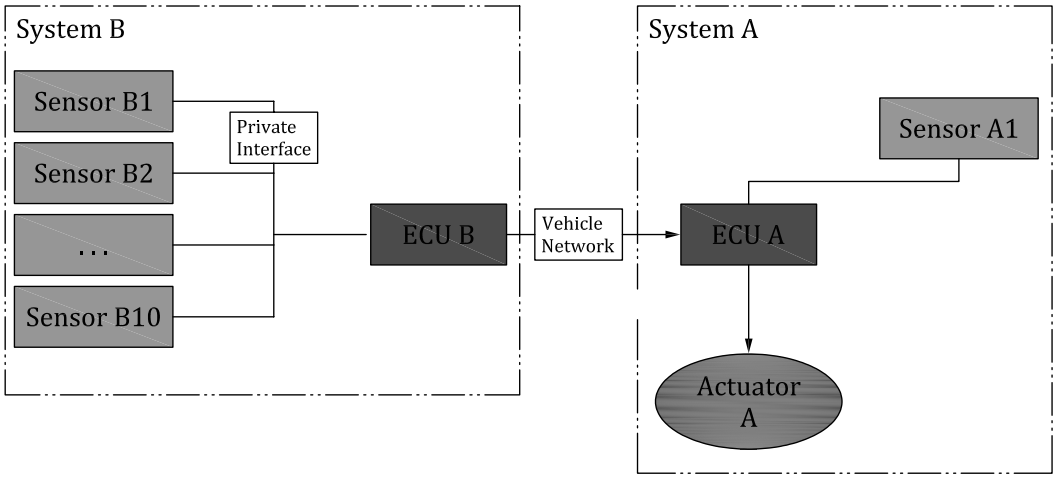


Figure G.1 — System architecture of the item

System B consists out of one ECU, ECU B, and ten sensors, sensor B1 to sensor B10. System A consists out of one ECU, ECU A, one sensor, sensor A1, and one actuator, actuator A.

G.3 Chain of events

The sensors sensor B1 to sensor B10 forward their signals SigB1 to SigB10 to ECU B (see Figure G.2). ECU B uses these signals to calculate new signal values SigB11 to SigB1000. Signals SigB1 to SigB10 received from the sensors and signals SigB11 to SigB1000 are then forwarded by ECU B to ECU A.

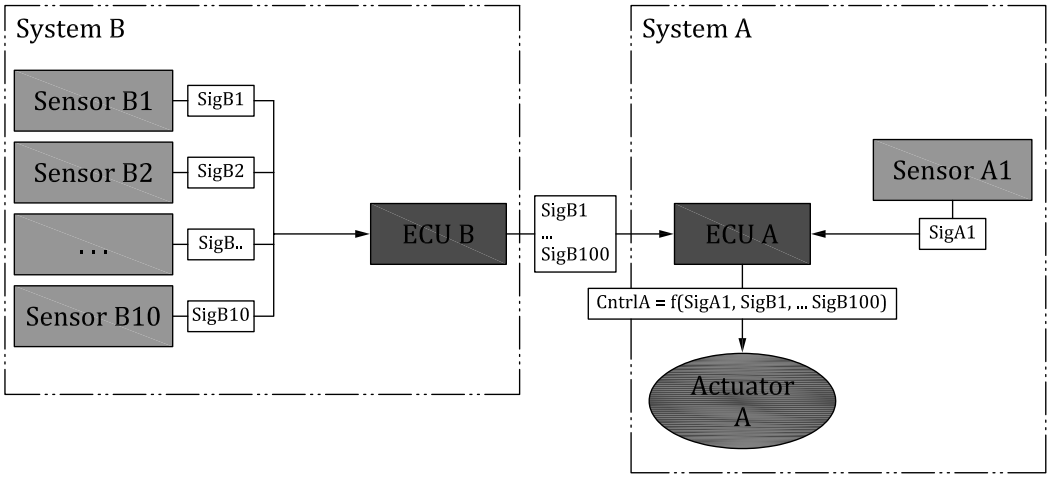


Figure G.2 — Chain of events

ECU A uses the signal SigB1 to SigB1000 and signal SigA1 from sensor A1 to calculate the control value CntrlA which is then applied to the actuator.

G.4 Conditions

The following conditions apply:

— Safety goal SG_A: Avoid incorrect actuation of actuator A for longer than 100 ms (ASIL D);

— Each signal SigA1, SigB1 to SigB1000 can lead to the violation of the safety goal A in case of an incorrect value;

— ECU A has no possibility to check SigB1 to SigB1000; and

— System B needs to check SigB1 to SigB1000 for correctness.

G.5 Overall PMHF target value

According to 9.4.2.3, the PMHF budget can be adapted according to the number of systems in the item. In this case, two systems and the vehicle network can contribute to the violation of the safety goal. For each system, a budget of 10^{-8}/h is allocated. In addition, for the vehicle network a budget of 10^{-10}/h is allocated (e.g. derived from quantitative analysis applied to similar design). As a result, the overall budget is $2{,}01 \times 10^{-8}$/h for the violation of the safety goal SG_A (see Figure G.3).

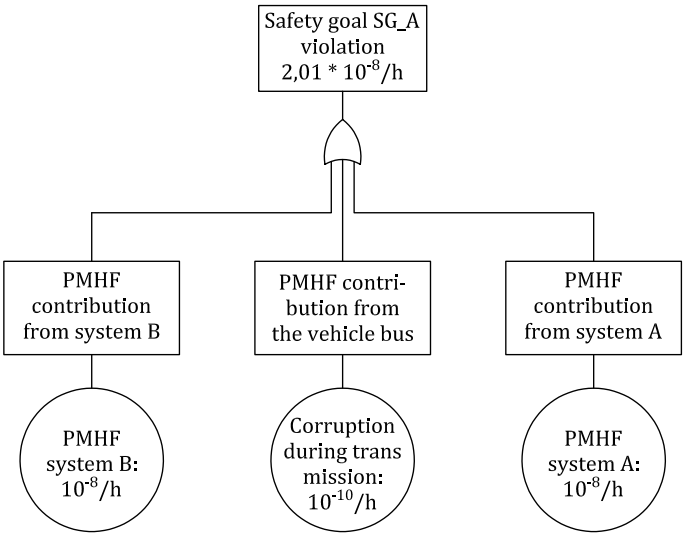


Figure G.3 — PMHF target allocation

G.6 PMHF budget specification for system B

As a first step, all signals provided by system B are identified which can contribute to the violation of the safety goal. They are assigned to one signal group, e.g. SigGroup_SG_A = [SigB1, ..., SigB1000].

As a next step, the budget allocated to system B concerning the violation of the safety goal A is allocated to the corruption of one or more signals of the signal group SigGroup_SG_A.

SafReq_B1: Prevent output of one or more incorrect signals of signal group SigGroup_SG_A (ASIL D) with:

— PMHF ≤ 10^{-8}/h;

— SPFM ≥ 99 %;

— LFM ≥ 90 %; and

— A signal is considered incorrect signal if the signal deviation from the correct value is ≥ maximum of (constant, x %).

This allows the supplier of system B to identify all relevant HW elements within one safety analysis and allocate failure rate budgets as he sees appropriate (e.g. Figure G.4).

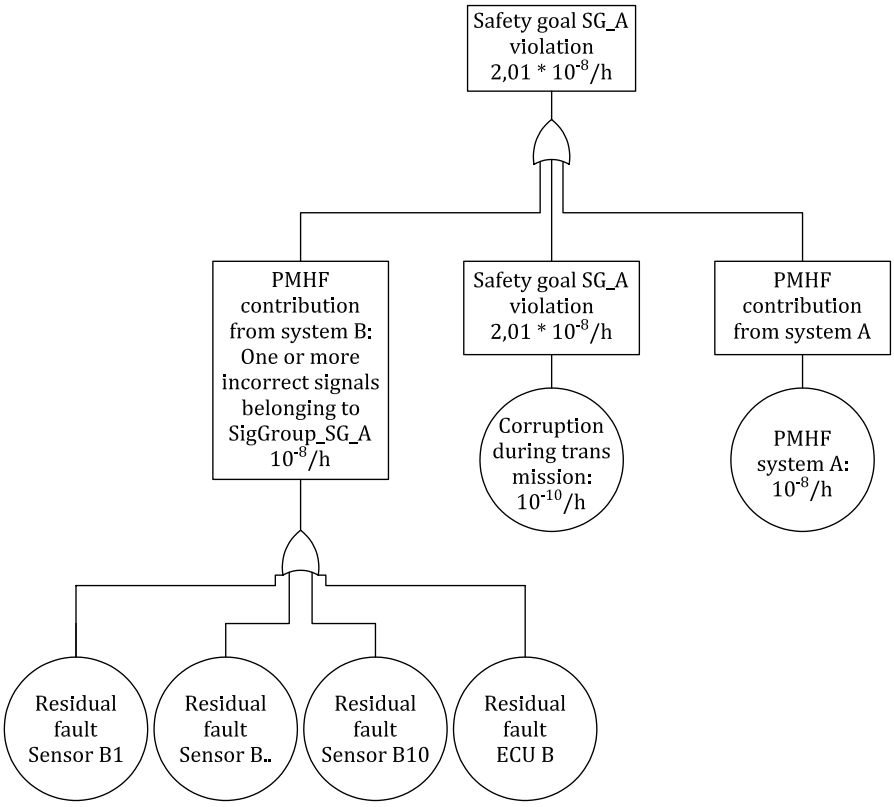


Figure G.4 — FTA considering HW elements

NOTE Since each and every one of the 1 000 signals provided by system B can violate SG_A, and since system A cannot check the correctness of these signals, one could be tempted to allocate the system B PMHF budget equally to each signal, leading to a PMHF budget of 10^{-8} / 1 000/h = 10^{-11}/h per signal. In this case, the safety requirement forwarded to the system A provider could then be formulated as "Prevent output of an incorrect value of signal Bx (ASIL D, PMHF_SigBx ≤ 10^{-11}/h, SPFM ≥ 99 %, LFM ≥ 90 %), for x = 1 to 1 000". However, the overall PMHF calculation for system B considers that these signals have common hardware elements (e.g. ECU B), contributing to their failure rates, making their failures not independent from each other. In this example, each signal can be affected by faults in ECU B. This means that faults in the ECU B can corrupt 1 to 1 000 signals. If the residual failure rate of each signal is summed without considering this fact, this sum could be higher than the base failure rate of the whole system B. Therefore, this approach is not appropriate.

Annex H
(informative)

Example of latent fault handling

H.1 General

This annex is intended to clarify how to deal with different types of safety mechanisms, giving two examples, in order to evaluate coherently the HW metrics as required in alternative a) of 8.4.8. Safety mechanisms are divided in two groups: safety mechanisms based on a combination of fault detection and control (a transition to a safe state in which even a subsequent failure of the safety mechanism would have no effect), and safety mechanisms based only on control of the fault effect. This annex is also intended to elucidate the reason for the application of alternative c) of 8.4.8 restricted to safety mechanisms performing fault detection and control.

H.2 Example with a safety mechanism based on fault detection and control

In the first example (see Figure H.1), the system is designed with a safety mechanism based on fault detection and control:

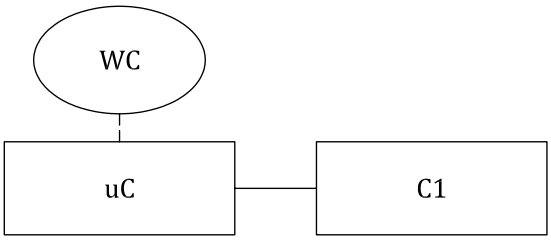


Figure H.1 — Safety mechanism based on fault detection and control

In this graphical representation: WD= Window Watchdog; uC= microcontroller; C1= Component 1.

The window watchdog is a safety mechanism monitoring the uC; its diagnostic coverage with regard to uC's residual faults (e.g. clock related problems, etc.) is 60 %. This percentage of covered faults with the potential to violate a safety goal in absence of safety mechanism is considered as a detected multiple-point fault (in particular it is a dual-point fault), because the fault is detected by a safety mechanism, indicated to the driver and the violation of the safety goal is prevented (i.e. the fault is controlled). Once the window watchdog detects the fault, it will transition the system into a safe state.

A fault occurring on the window watchdog can deactivate its detection or error control capability of the uC failure modes: if this fault is neither detected by a safety mechanism (WD start-up test), nor perceived by the driver (assuming the effects of the faulty WD cannot be perceived by the driver), it is considered a latent fault.

Considering the classification just mentioned, the following table is a quantitative analysis excerpt that describes the evaluation of the HW metric:

Table H.1 — Quantitative analysis excerpt with a safety mechanism based on fault detection and control

Component Name	Failure Mode	Failure Rate λ	Potential to directly violate the SG?	Safety mechanism preventing the failure mode from violating the safety goal?	DC	Residual Fault failure rate	Potential to violate the SG in combination with another fault?	Safety Mechanism preventing the failure mode from being latent?	FM coverage respect to LF	LF failure rate
uC	Clock related problems	100 FIT	Yes	Window Watchdog	60 %	40 FIT	Yes	Window Watchdog	100 %	0
Window Watchdog	Failure	40 FIT	No	—	—	—	Yes	WD start-up test	90 %	4 FIT
C1	Failure	50 FIT	Yes	uC	97 %	1,5 FIT	Yes	Yes	100 %	0
…										

NOTE 1 The alternative c) of 8.4.8 can be applied to systems in which each safety mechanism, whose unavailability can contribute to the violation of the safety goal, is based on fault detection and control. In this case, the alternative c) is effective because the failure mode monitored by the safety mechanism does not gives contribution to the latent fault FIT. Thus, the only contribution to the latent fault FIT comes from the Safety Mechanism itself.

NOTE 2 The failure mode of the uC (clock related problems – 100 FIT) has the potential to violate the safety goal; the window watchdog is the safety mechanism in place to control the fault. The failure of the window watchdog has the potential to violate the safety goal in combination with another fault. The window watchdog faults which are not detected by the watchdog start-up test are accounted as latent faults (4 FIT). The fraction of the uC failure mode which is prevented by the safety mechanism from violating the safety goal is the 60 % (60 FIT), thus the failure rate related to the residual fault is 40 FIT. The 60 % of the considered failure mode (60 FIT) which are prevented from the violation of the safety goal are detected by the safety mechanism in place (window watchdog), which detects the fault and triggers the transition to the safe state. According to the classification of random hardware faults of ISO 26262-10:2018, the faults covered by the safety mechanism (60 FIT) are considered as detected multiple point faults and, consequently, they cannot be considered as latent faults by definition.

NOTE 3 Faults in the component C1 are controlled by the uC with a DC of 97 %.

The argument is valid for safety mechanisms based on fault detection and control; the argument is not valid for Safety Mechanisms performing only a control of the fault effects (e.g. RAM's EDC without error signalling).

H.3 Example with a safety mechanism based on control of the fault effects

In the second example (see Figure H.2), the system is designed with a Safety Mechanism based on control of the fault effects:

Figure H.2 — Safety Mechanism based on control of the fault effects

In this graphical representation (Figure H.2): A = component A; Filter = discrete component; B = Component B.

The safety requirement is that “The system has to provide a signal within its electrical specification”; the violation of the safety requirement could lead to the violation of the safety goal.

NOTE 1 It is assumed that component A provides a correct signal if fault-free.

If a fault occurs on the component A, such that the signal provided by A is disturbed showing a noise, it is filtered by the safety mechanism (filter). Thus, if the signal shows a permanent noise, it will be always punctually restored by the filter, since it works correctly. The failure mode coverage for the failure mode “noise on signal” of the safety mechanism in place is 99,9 %. Since in this case, where the safety mechanism in place performs a permanent and punctual recovery, the fault with the potential to violate a safety goal in absence of a safety mechanism is controlled, but it is neither detected nor perceived; thus it is considered as a latent fault.

Once the permanent fault on the component A occurred, the dual-point failure is realized when the fault on the filter occurs; until that moment, the fault covered on component A will not be visible at system level.

NOTE 2 Safety mechanisms acting like the filter in the example do not perform detection of the fault, but they only control it. Thus the system is not able to distinguish between a malfunction or a normal behaviour (the action of the safety mechanism is triggered until the failure of the safety mechanism itself occurs).

Table H.2 — Quantitative analysis excerpt with a safety mechanism based on control of the fault effects

Component Name	Failure Mode	Failure Rate λ	Potential to directly violate the SG?	Safety mechanism preventing the failure mode from violating the safety goal?	DC	Residual Fault failure rate	Potential to violate the SG in combination with another fault?	Safety Mechanism preventing the failure mode from being latent?	FM coverage respect to LF	LF failure rate
A	Noise on signal	100 FIT	Yes	Filter	99,9 %	0,1 FIT	Yes	No	0 %	99,9 FIT
Filter	Failure	40 FIT	No	—	—	—	Yes	No	0 %	40 FIT
B	Failure	20 FIT	Yes	—	0 %	20 FIT	—	—	—	—
...										

NOTE 3 Alternative c) of 8.4.8 is not applicable to this system characterized by a safety mechanism based on control of the fault effect. Thus, in this case alternatives a) and b) of 8.4.8 are the only possibilities.

Bibliography

[1] ISO 7637-2, *Road vehicles — Electrical disturbances from conduction and coupling — Part 2: Electrical transient conduction along supply lines only*

[2] ISO 7637-3, *Road vehicles — Electrical disturbances from conduction and coupling — Part 3: Electrical transient transmission by capacitive and inductive coupling via lines other than supply lines*

[3] ISO 10605, *Road vehicles — Test methods for electrical disturbances from electrostatic discharge*

[4] ISO 11452-2, *Road vehicles — Component test methods for electrical disturbances from narrowband radiated electromagnetic energy — Part 2: Absorber-lined shielded enclosure*

[5] ISO 11452-4, *Road vehicles — Component test methods for electrical disturbances from narrowband radiated electromagnetic energy — Part 4: Harness excitation methods*

[6] ISO 16750-2, *Road vehicles — Environmental conditions and testing for electrical and electronic equipment — Part 2: Electrical loads*

[7] ISO 16750-4, *Road vehicles — Environmental conditions and testing for electrical and electronic equipment — Part 4: Climatic loads*

[8] ISO 16750-5, *Road vehicles — Environmental conditions and testing for electrical and electronic equipment — Part 5: Chemical loads*

[9] IEC 61508 (all parts), *Functional safety of electrical/electronic/programmable electronic safety-related systems*

[10] IEC 61709, *Electronic components — Reliability — Reference conditions for failure rates and stress models for conversion*

[11] SN 29500 (2004), *Siemens AG, Failure Rates of Components — Expected Values, General*

[12] *Intentionally left blank*

[13] EN 50129:2003, *Railway applications — Communication, signalling and processing systems — Safety related electronic systems for signalling*

[14] MIL HDBK 217 F notice 2, Military handbook: Reliability prediction of electronic equipment

[15] MIL HDBK 338, Military handbook: Electronic reliability design handbook

[16] NPRD-2016, Non-electronic Parts Reliability Data

[17] RIAC FMD-2016. Failure Mode / Mechanism Distributions

[18] RIAC HDBK 217 Plus, Reliability Prediction Models

[19] UTE C80-811, Reliability methodology for electronic systems

[20] BIROLINI. A., Reliability Engineering, *Theory and Practice*, 2014

[21] Sundaram P., & D'Ambrosio J.G. Controller Integrity in Automotive Failsafe System Architectures, SAE 2006 World Congress, 2006-01-0840

[22] Fruehling T., & Delphi Secured Microcontroller Architecture S.A.E. 2000 World Congress, SAE# 2000-01-1052

[23] Mahmood A., & McCluskey E.J. " Concurrent Error Detection Using Watchdog Processors – A Survey", *IEEE Trans. Computers*, **37**(2), 160-174 (1988)

[24] Leaphart E., Czerny B., D'Ambrosio J. Survey of Software Failsafe Techniques for Safety-Critical Automotive Applications, SAE 2005 World Congress, 2005-01-0779

[25] Mariani R., Fuhrmann P., Vittorelli B. Cost-effective Approach to Error Detection for an Embedded Automotive Platform, 2006-01-0837, SAE 2006 World Congress & Exhibition, April 2006, Detroit, MI, USA

[26] Patel J., & Fung L. " Concurrent Error Detection in ALU's by Recomputing with Shifted Operands", *IEEE Transactions on Computers*, Vol. **C-31**, pp.417-422, July 1982

[27] Forin P. Vital Coded Microprocessor: Principles and Application for various Transit Systems, Proc. IFAC-GCCT, Paris, France, 1989

[28] Ramabadran T.V., & Gaitonde S.S. 1988), " A tutorial on CRC computations". *IEEE Micro* **8** (4): 62–75, 1988

[29] Koopman P., & Chakravarty T. 2004), Cyclic Redundancy Code (CRC) Polynomial Selection For Embedded Networks The International Conference on Dependable Systems and Networks, DSN-2004, http://www.ece.cmu.edu/~koopman/roses/dsn04/koopman04_crc_poly_embedded.pdf

[30] FIDES guide 2009 edition A (September 2010), Reliability Methodology for Electronic Systems

ICS 43.040.10

Price based on 94 pages