DRAFT INTERNATIONAL STANDARD ISO/DIS 26262-5

ISO/TC **22**/SC **3** Secretariat: **DIN**

Voting begins on: **2009-07-08** Voting terminates on: **2009-12-08**

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION

Road vehicles — Functional safety —

Part 5:
Product development: hardware level

Véhicules routiers — Sécurité fonctionnelle —

Partie 5: Développement du produit : niveau matériel

ICS 43.040.10

In accordance with the provisions of Council Resolution 15/1993 this document is circulated in the English language only.

Conformément aux dispositions de la Résolution du Conseil 15/1993, ce document est distribué en version anglaise seulement.

To expedite distribution, this document is circulated as received from the committee secretariat. ISO Central Secretariat work of editing and text composition will be undertaken at publication stage.

Pour accélérer la distribution, le présent document est distribué tel qu'il est parvenu du secrétariat du comité. Le travail de rédaction et de composition de texte sera effectué au Secrétariat central de l'ISO au stade de publication.

Contents

Page

Foreword v
Introduction vi
1 Scope 1
2 Normative references 1
3 Terms and definitions 2
4 Requirements for compliance 2
4.1 General requirements 2
4.2 Interpretations of tables 2
4.3 ASIL dependent requirements and recommendations 2
5 Initiation of product development at the hardware level 3
5.1 Objectives 3
5.2 General 3
5.3 Inputs to this clause 4
5.4 Requirements and recommendations 5
5.5 Work products 5
6 Specification of hardware safety requirements 5
6.1 Objectives 5
6.2 General 5
6.3 Inputs to this clause 5
6.4 Requirements and recommendations 6
6.5 Work products 8
7 Hardware design 8
7.1 Objectives 8
7.2 General 8
7.3 Inputs to this clause 8
7.4 Requirements and recommendations 9
7.5 Work products 13
8 Hardware architectural metrics 13
8.1 Objectives 13
8.2 General 13
8.3 Inputs of this clause 14
8.4 Requirements and recommendations 14
8.5 Work products 16
9 Evaluation of violation of the safety goal due to random HW failures 16
9.1 Objectives 16

9.2 General ... 16
9.3 Inputs to this clause ... 17
9.4 Requirements and recommendations ... 17
9.5 Work products ... 25
10 Hardware integration and testing ... 25
10.1 Objectives ... 25
10.2 General ... 25
10.3 Inputs of this clause ... 26
10.4 Requirements and recommendations ... 26
10.5 Work products ... 29
Annex A (informative) Overview on and document flow of product development at the hardware level ... 30
Annex B (informative) Failure mode classification of a hardware element ... 32
Annex C (normative) Hardware architectural metrics ... 34
Annex D (informative) Evaluation of the diagnostic coverage ... 38
Annex E (informative) Target values for hardware architectural metrics ... 53
Annex F (informative) Example of calculation of the hardware architectural metrics: "single point faults metric" and "latent faults metric" ... 54
Annex G (informative) Target values for violation of a safety goal due to random hardware failures ... 60
Bibliography ... 61

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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

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.

ISO 26262-5 was prepared by Technical Committee ISO/TC 22, *Road vehicles*, Subcommittee SC 3, *Electrical and electronic equipment*.

ISO 26262 consists of the following parts, under the general title *Road vehicles — Functional safety*:

— *Part 1: Vocabulary*

— *Part 2: Management of functional safety*

— *Part 3: Concept phase*

— *Part 4: Product development: system level*

— *Part 5: Product development: hardware level*

— *Part 6: Product development: software level*

— *Part 7: Production and operation*

— *Part 8: Supporting processes*

— *Part 9: ASIL-oriented and safety-oriented analyses*

— *Part 10: Guideline on ISO 26262*

Introduction

ISO 26262 is the adaptation of IEC 61508 to comply with needs specific to the application sector of 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 elements that provide safety-related functions.

Safety is one of the key issues of future automobile development. New functionality not only in the area of driver assistance but also in vehicle dynamics control and active and passive safety systems increasingly touches the domain of safety engineering. Future development and integration of these functionalities will even strengthen the need of safe system development processes and the possibility to provide evidence that all reasonable safety objectives are satisfied.

With the trend of increasing complexity, software content and mechatronic implementation, there are increasing risks from systematic failures and random hardware failures. ISO 26262 includes guidance to avoid these risks by providing feasible requirements and processes.

System safety is achieved through a number of safety measures, which are implemented in a variety of technologies (for example: mechanical, hydraulic, pneumatic, electrical, electronic, programmable electronic etc). Although ISO 26262 is concerned with E/E systems, it provides a framework within which safety-related systems based on other technologies can be considered.

ISO 26262:

— provides an automotive safety lifecycle (management, development, production, operation, service, decommissioning) and supports tailoring the necessary activities during these lifecycle phases;

— provides an automotive specific risk-based approach for determining risk classes (Automotive Safety Integrity Levels, ASILs);

— uses ASILs for specifying the item's necessary safety requirements for achieving an acceptable residual risk; and

— provides requirements for validation and confirmation measures to ensure a sufficient and acceptable level of safety being achieved.

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 by the management processes.

Safety issues are intertwined with common function-oriented and quality-oriented development activities and work products. ISO 26262 addresses the safety-related aspects of the development activities and work products.

Figure 1 shows the overall structure of ISO 26262. ISO 26262 is based upon a V-Model as a reference process model for the different phases of product development. The shaded "V"s represents the relations between ISO 26262-3, ISO 26262-4, ISO 26262-5, ISO 26262-6 and ISO 26262-7.

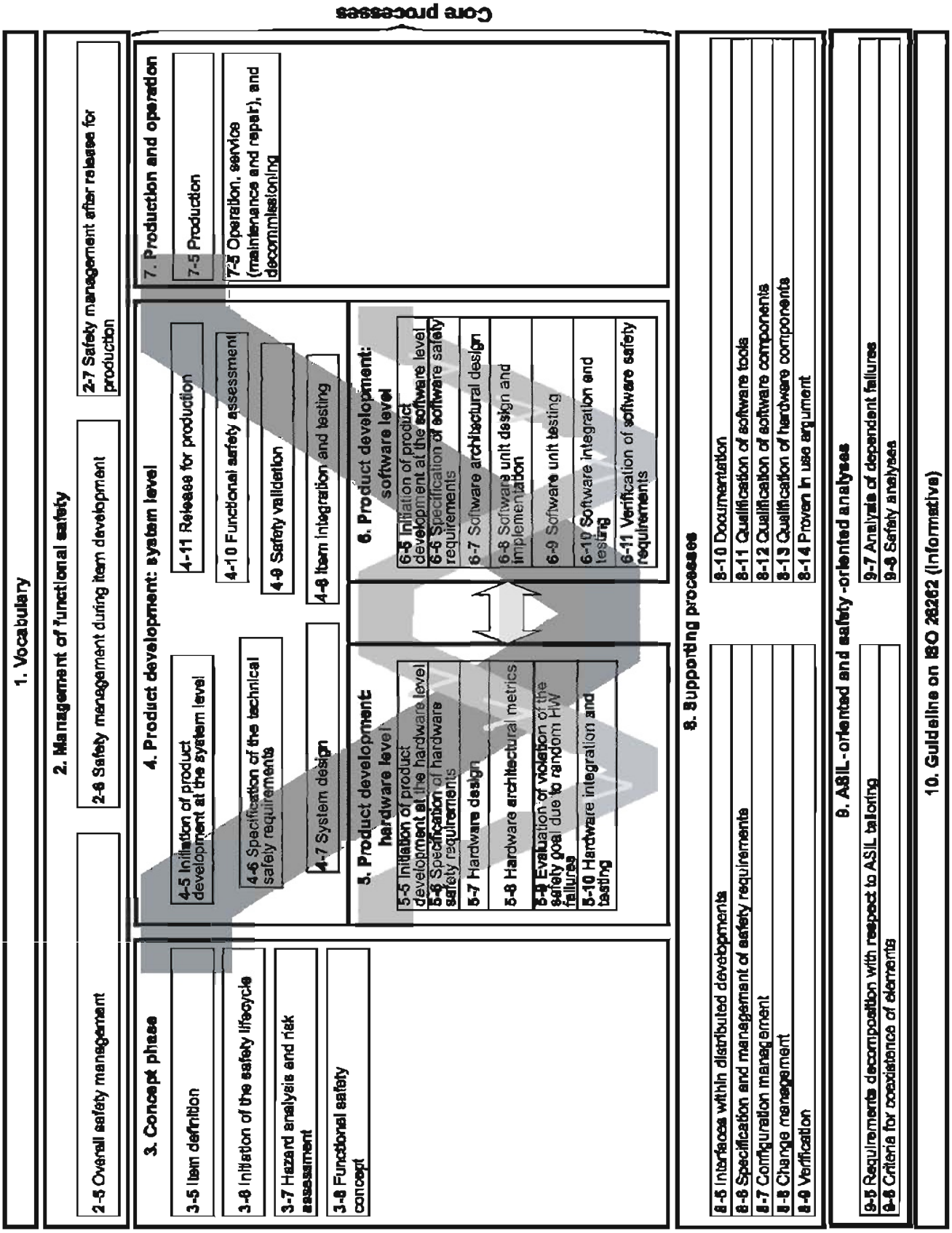


Figure 1 — Overview of ISO 26262

Road vehicles — Functional safety — Part 5: Product development: hardware level

1 Scope

ISO 26262 is intended to be applied to safety-related systems that include one or more E/E systems and that are installed in series production passenger cars with a max gross weight up to 3,5 t. ISO 26262 does not address unique E/E systems in special purpose vehicles such as vehicles designed for drivers with disabilities. Systems developed prior to the publication date of ISO 26262 are exempted from the scope.

ISO 26262 addresses possible hazards caused by malfunctioning behaviour of E/E safety-related systems including interaction of these systems. It does not address hazards as electric shock, fire, smoke, heat, radiation, toxicity, flammability, reactivity, corrosion, release of energy, and similar hazards unless directly caused by malfunctioning behaviour of E/E safety-related systems.

ISO 26262 does not address the nominal performance of E/E systems, even if dedicated functional performance standards exist for these systems (for example active and passive safety systems, brake systems, ACC).

This part of ISO26262 specifies the requirements on product development at the hardware level. These include requirements on the initiation of product development at the hardware level, the specification of the hardware safety requirements, hardware design, hardware architectural metrics, and evaluation of violation of the safety goal due to random hardware failures and hardware integration and testing.

2 Normative references

The following referenced documents are indispensable for the application 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: —[1] *Road vehicles – Functional Safety — Part 1: Vocabulary*

ISO 26262-2: —[1] *Road vehicles – Functional Safety — Part 2: Management of functional safety*

ISO 26262-4: —[1] *Road vehicles – Functional Safety — Part 4: Product development: system level*

ISO 26262-6: —[1] *Road vehicles – Functional Safety — Part 6: Product development: software level*

ISO 26262-8: —[1] *Road vehicles – Functional Safety — Part 8: Supporting processes*

ISO 26262-9: —[1] *Road vehicles – Functional Safety — Part 9: ASIL-oriented and safety-oriented analyses*

[1] To be published

3 Terms and definitions

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

4 Requirements for compliance

4.1 General requirements

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

1) Tailoring in accordance with ISO 26262-2 has been planned and shows that the requirement does not apply.

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

Information marked as a "NOTE" is only for guidance in understanding, or for clarification of, the associated requirement and shall not be interpreted as a requirement itself.

4.2 Interpretations of tables

Tables may be normative or informative depending on their context.

The different methods listed in a table contribute to the level of confidence that the corresponding requirement shall apply.

Each method in a table is either a consecutive entry (marked by a sequence number in the leftmost column, e.g. 1, 2, 3) or an alternative entry (marked by a number followed by a letter in leftmost column, e.g., 2a, 2b, 2c).

For consecutive entries all methods are recommended in accordance with the ASIL. If methods other than those listed are to be applied a rationale shall be given that they comply with the corresponding requirement.

For alternative entries an appropriate combination of methods shall be applied in accordance with the ASIL, independently of whether they are listed in the table or not. If methods are listed with different degrees of recommendation for an ASIL the higher one should be preferred. A rationale shall be given that the selected combination of methods complies with the corresponding requirement. If all highly recommended methods listed for a particular ASIL are selected a rationale needs not to be given.

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

”++” The method is highly recommended for this ASIL.

“+“ The method is recommended for this ASIL.

“o“ The method has no recommendation for or against its usage for this ASIL.

4.3 ASIL dependent requirements and recommendations

The requirements or recommendations of each subclause shall apply to 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:— Clause 54 the ASIL resulting from the decomposition will apply.

If an ASIL is given in parentheses, the corresponding subclause shall be read as a recommendation rather than a requirement for this ASIL.

5 Initiation of product development at the hardware level

5.1 Objectives

The objective of the initiation of the product development for the hardware is to determine and plan the functional safety activities during the individual sub-phases of hardware development. This also includes the necessary supporting processes described in ISO 26262-8.

This planning of hardware-specific safety activities is included in the safety plan.

5.2 General

Integration of the following activities is crucial for the product development at hardware level:

— Hardware implementation of the technical safety concept;

— Analysis of potential faults and their effects; and

— Coordination with software development.

The necessary activities and processes are planned. Figure 2 illustrates how the different activities are carried out in order to comply with the requirements of this part, and how these activities are integrated in the whole ISO 26262 frameworks.

NOTE Requirements of this part on hardware are applicable both to non-programmable and programmable elements such as ASIC, FPGA, PLD. Furthermore, for programmable electronic elements, requirements in ISO 26262-6:—, ISO 26262-8:—, Clause 11 and ISO 26262-8:—, Clause 12 are applicable.

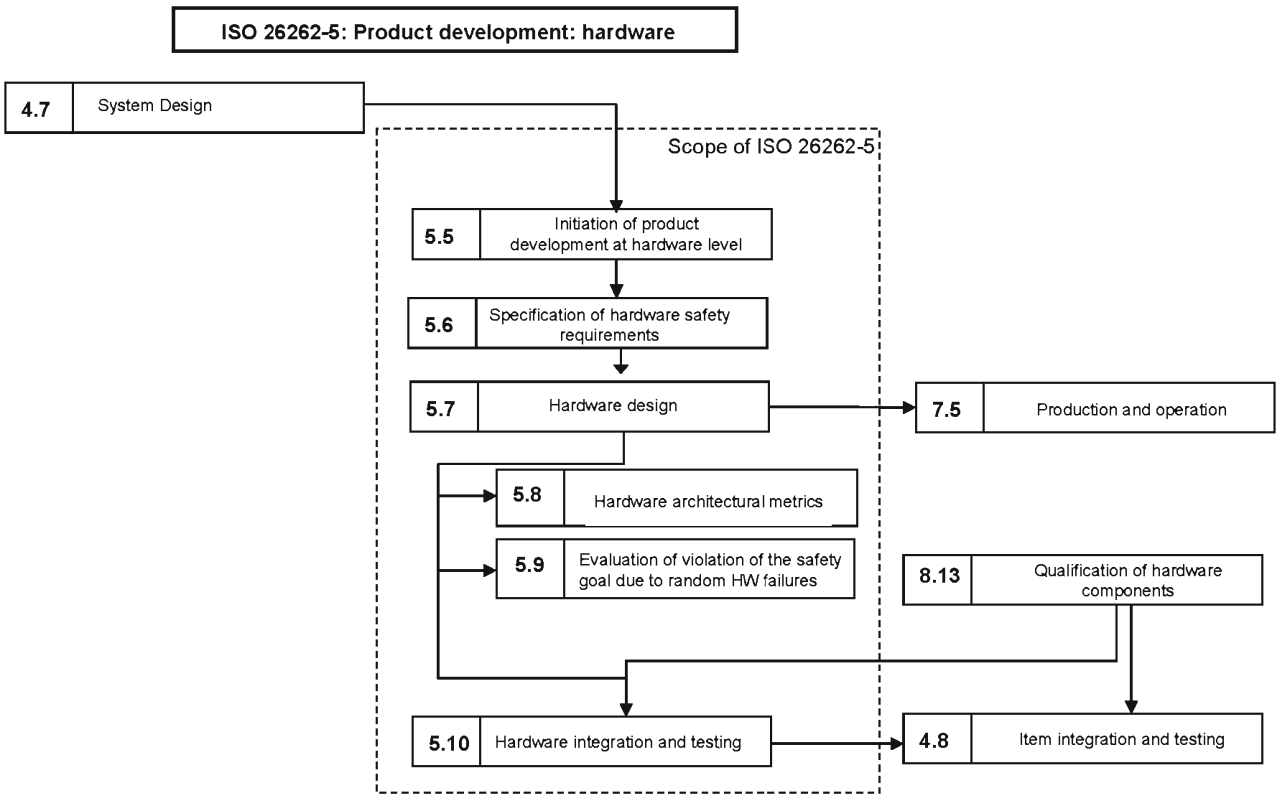


Figure 2 — Informative reference phase model for the development of a safety-related item

5.3 Inputs to this clause

5.3.1 Prerequisites

The following information shall be available:

— Overall project plan (refined) (see ISO 26262-4:—, 5.5.1)

— Safety plan (refined) (see ISO 26262-4:—, 5.5.2)

— Item integration and testing plan (refined) (see ISO 26262-4:—, 7.5.4)

5.3.2 Further supporting information

None

5.4 Requirements and recommendations

5.4.1 The activities for the product development of the hardware elements of the item described in Clause 6 to Clause 10 shall be planned and included in the safety plan consistently with the planning of activities in ISO 26262-6.

NOTE The project plan and safety plan as work products of ISO 26262-2:—, Clause 6 and ISO 26262-4:—, Clause 5 and ISO 26262-4:—, Clause 6 are detailed in this Clause 5.

5.4.2 The safety plan shall specify the activities to determine appropriate methods and measures to be used during the design to ensure the functional safety of the hardware elements of the item.

5.4.3 The hardware development process for the hardware of the item, including lifecycle phases, methods, and tools, shall be consistent across all subphases of the hardware lifecycle and compatible with system and software lifecycles, such that required data can be transformed correctly.

5.5 Work products

5.5.1 **Overall project plan (refined)** resulting from requirements 5.4.1 to 5.4.3.

5.5.2 **Safety plan (refined)** resulting from requirements 5.4.1 to 5.4.3.

6 Specification of hardware safety requirements

6.1 Objectives

The first objective of this clause is to make available a consistent and complete hardware specification that will be applied to the hardware of the item or element under consideration. The requirements of this specification are hardware safety requirements.

The second objective is to verify that the hardware safety requirements are consistent with the technical safety concept.

A further objective of this phase is to detail the hardware-software interface (HSI) requirements initiated in ISO 26262-4:—, Clause 7.

6.2 General

The hardware safety requirements to be defined can be derived from several sources:

— Technical safety concept specified in ISO 26262-4:—, Clause 7 (including relevant environmental conditions and conditions of operation)

— Software safety requirements specified in ISO 26262-6:—, Clauses 6 and 7.

6.3 Inputs to this clause

6.3.1 Prerequisites

The following information shall be available:

— Overall project plan (refined) (see 5.5.1)

— Safety plan (refined) (see 5.5.2)

— Technical safety concept (see ISO 26262-4:—, 7.5.1)

— System design specification (see ISO 26262-4:—, 7.5.2)

— Hardware software interface specification (see ISO 26262-4:—, 7.5.6)

6.3.2 Further supporting information

The following information may be considered:

— Software safety requirements specification (see ISO 26262-6:—, 6.5.1)

6.4 Requirements and recommendations

6.4.1 A consistent and complete hardware safety requirements specification for the hardware of the element under consideration shall be derived from the technical safety requirements allocated to hardware.

6.4.2 The hardware safety requirements specification shall include each hardware requirement that relates to safety including:

b) The hardware safety requirements of safety mechanisms to control internal failures of the hardware of the element, with their relevant attributes;

EXAMPLE 1 These attributes can be for instance the timing and detection abilities of a watchdog.

c) The hardware safety requirements of safety mechanisms to make the element under consideration tolerant to failures external to the element, with their relevant attributes

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

d) The hardware safety requirements of safety mechanisms to comply with the safety requirements of other elements

EXAMPLE 3 diagnoses of sensors or actuators.

e) The hardware safety requirements of safety mechanisms to detect and signal internal or external failures

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

NOTE 2 The hardware safety requirements, described in each bullet, include the characteristics needed to ensure the effectiveness of the above safety mechanisms.

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

NOTE 3 The hardware safety requirements include requirements on the target values as given by requirements 6.4.3 and 6.4.4, as well as requirements for avoidance of specific behaviour. An example of an avoidance requirement is that a particular sensor shall not produce wrong output, with an ASIL attached to this requirement.

6.4.3 This requirement applies to ASIL (B,) C, and D of the safety goal, in accordance with 4.3: The target values specified in ISO 26262-4:—, Clause 7 for the metrics of Clause 8 shall be considered when showing compliance of the hardware of the item.

NOTE This activity can include a split of target values in the case of a distributed development in accordance with ISO 26262-8:—, Clause 5.

6.4.4 This requirement applies to ASIL (B,) C, and D of the safety goal, in accordance with 4.3: The target values specified in ISO 26262-4:—, Clause 7 for the procedures of Clause 9 shall be considered when showing compliance of the hardware of the item.

NOTE This activity can include a split of target values in the case of a distributed development in accordance with ISO 26262-8:—, Clause 5.

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

6.4.6 The criteria for qualification and testing of the hardware of the item or element shall be specified in accordance with Clause 10, and ISO 26262-8:—, Clause 13. This shall include environmental conditions (temperature, vibration, EMC, etc), specific operational environment and component specific requirements.

6.4.7 The hardware safety requirements shall ensure compliance with the fault tolerant time interval for safety mechanisms as specified in ISO 26262-4:—, 6.4.9.

6.4.8 The hardware safety requirements shall ensure compliance with the multiple point fault detection interval as specified in ISO 26262-4:—, 6.4.10.2.

NOTE 1 For ASILC and D safety goals: if the corresponding safety concept does not prescribe specific values, the multiple point fault detection intervals can be specified to be equal 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 occurrence of random hardware failures, if they are considered in the calculation model (see Clause 9).

6.4.9 The hardware safety requirements shall be verified in accordance with Table 1 in order to show:

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 under consideration,

c) Correctness and accuracy;

d) Compliance with the requirements of this Clause.

Table 1 — Verification of the hardware safety requirements

Methods		***ASIL***			
		A	***B***	***C***	***D***
1a	*Walkthrough of hardware safety requirements*	++	+	o	o
1b	*Inspection of hardware safety requirements*	+	++	++	++

6.4.10 Hardware-software interface (HSI) specification

6.4.10.1 The hardware-software interface specification initiated in ISO 26262-4:—, Clause 7 shall be detailed down to a level allowing correct control and usage by software.

6.4.10.2 The persons responsible for hardware and software development shall jointly verify the adequacy of the refined HSI specification.

6.5 Work products

6.5.1 Hardware safety requirements specification (including test and qualification criteria) resulting from requirements 6.4.1, 6.4.2 and 6.4.6.

6.5.2 Hardware architectural metrics requirements resulting from requirement 6.4.3.

6.5.3 Random hardware failure requirements resulting from requirement 6.4.4.

6.5.4 Hardware-software interface specification (refined) resulting from requirements 6.4.10.1, 6.4.10.2.

6.5.5 Hardware safety requirements verification report resulting from requirement 6.4.9.

7 Hardware design

7.1 Objectives

The first objective of this clause is to design the hardware with respect to the system design specification and the hardware safety requirements.

The second objective of this clause is to verify the hardware design against the system design specification and the hardware safety requirements.

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 schematics representing the interconnections between hardware parts composing the hardware components.

In order to develop a single hardware design, the design complies with both hardware safety requirements as well as non-safety-related requirements, i.e. in this phase safety-related and non safety-related 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 (see 6.5.1)

— Hardware software interface specification (refined) (see 6.5.5)

— System design specification (ISO 26262-4:—, 7.5.2)

— Overall project plan (refined) (see 5.5.1)

— Safety plan (refined) (see 5.5.2)

7.3.2 Further supporting information

The following information may be considered:

— Software safety requirements specification (ISO 26262-6:—, 6.5.1)

7.4 Requirements and recommendations

7.4.1 Requirements for hardware architectural design

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

7.4.1.2 Each hardware element shall inherit the highest ASIL from the hardware safety requirements that it implements.

7.4.1.3 If ASIL decomposition is applied to the hardware architecture, it shall be applied in accordance with ISO 26262-9:—, Clause 5.

7.4.1.4 If a hardware element is made of sub-elements with different ASILs assigned, or of non-safety-related sub-elements and safety-related sub-elements, then each of these shall be treated in accordance with the highest ASIL, unless the criteria for coexistence, in accordance with ISO 26262-9:—, Clause 6, are met.

7.4.1.5 The traceability between the hardware safety requirements and their implementation shall be maintained down to hardware components.

NOTE This means that the traceability is not required down to hardware detailed design and no ASILs are assigned to hardware parts.

7.4.1.6 In order to achieve an adequate level of granularity and to avoid failures resulting from high complexity, the following modular design properties shall be considered:

1) This requirement applies to ASIL (A, B, C, and D), in accordance with 4.3: Hierarchical design;

2) This requirement applies to ASIL A, B, C, and D, in accordance with 4.3: Precisely defined interfaces of safety-related hardware components;

3) This requirement applies to ASIL (A, B, C), and D, in accordance with 4.3: Avoidance of unnecessary complexity of interfaces;

4) This requirement applies to ASIL (A, B), C, and D, in accordance with 4.3: Avoidance of unnecessary complexity of hardware components;

5) This requirement applies to ASIL (A, B, C, and D), in accordance with 4.3: Maintainability (service);

6) This requirement applies to ASIL (A, B), C, and D, in accordance with 4.3: Testability.

NOTE Testability includes testability during development and operation.

7.4.1.7 Well-trusted hardware components should be considered for re-use, in accordance with ISO 26262-4:—, 7.4.3.4.

NOTE The aim of the use of well-trusted hardware components is to avoid unknown and first time failures. However, it is not intended to limit the application of new technology where there is a benefit and the safety properties of the new technology have been analysed to a level of detail appropriate for the assigned ASIL.

7.4.1.8 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, cross-talks 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, lessons learned, if applicable, shall be used.

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-talks originating either from other hardware parts of the hardware component or from its environment.

7.4.2.3 The hardware detailed design shall ensure that hardware parts are used within their environmental and operational specifications.

7.4.2.4 Robust design principles shall be considered.

7.4.3 Safety Analyses

7.4.3.1 Safety analysis of hardware architectural and detailed design to determine effects and causes of faults shall be applied in accordance with Table 2 and ISO 26262-9:—, Clause 8.

NOTE 1 The purpose of these analyses is first to support the specification of the hardware architectural and detailed design. The same analyses can then be used for verification of the hardware design (see 7.4.4).

NOTE 2 At this stage, qualitative analyses might be appropriate and sufficient.

Table 2 — Hardware design safety analysis

Methods		ASIL			
		A	B	C	D
1	Deductive analysis[a]	o	+	++	++
2	Inductive analysis[b]	++	++	++	++
[a] A typical deductive analysis method is FTA.					
[b] A typical inductive analysis method is FMEA.					

NOTE The level of detail of the analysis is chosen in an appropriate manner.

7.4.3.2 This requirement applies to ASIL (B,) C, D of the safety goal, in accordance with 4.3: For each safety-related hardware element the safety analyses shall identify the following for the safety goal under consideration:

a) Safe faults;

b) Single point faults or residual faults;

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

NOTE 1 Multiple point faults of higher order than two are considered if shown relevant in the safety concept.

NOTE 2 The safety analysis is done at a feasible level with respect to the type of element.

NOTE 3 The intention of the identification of dual point faults is not to require a systematic analysis of every possible combination of 2 hardware faults but at least to consider combinations that derives from the safety concept (for instance the combination of 2 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).

7.4.3.3 This requirement applies to ASIL (B,) C, D of the safety goal, in accordance with 4.3. Evidence of effectiveness of the safety mechanisms to avoid single point faults shall be made available.

For that purpose:

a) Evidence of the ability of the safety mechanisms to maintain a safe state or to switch safely into a safe state shall be made available (in particular, appropriate failure mitigation ability within fault tolerant time interval); and

b) Diagnostic coverage shall be evaluated.

NOTE 1 A fault cannot be considered covered if its diagnostic test interval plus the fault reaction time of the associated safety mechanism is higher than the relevant fault tolerant time interval.

NOTE 2 A FMEA can be used to structure the justification.

7.4.3.4 This requirement applies to ASIL (B,) C, D of the safety goal, in accordance with 4.3. Evidence of effectiveness of the safety mechanisms to avoid multiple point faults remaining latent shall be made available.

For that purpose:

a) Evidence of the failure detection and the ability to signal to the driver within the acceptable multiple point fault interval for latent faults shall be made available in order to determine which faults remain latent and which faults are not latent; and

b) Diagnostic coverage with regard to latent multiple point faults shall be evaluated.

NOTE 1 A fault cannot be considered covered if its diagnostic test interval plus the fault reaction time of the associated safety mechanism is higher than the relevant multiple point fault interval for latent faults.

NOTE 2 A FMEA can be used to structure the justification.

7.4.3.5 In the case of independence requirements, evidence of the compliance of the hardware design shall be made available by an analysis of dependent failures in accordance with ISO 26262-9:—, Clause 7 applied on the hardware design.

7.4.3.6 If new hazards introduced by the hardware design are identified, they shall be introduced and evaluated in the hazard analysis and risk assessment in accordance with the change management process in ISO 26262-8:—, Clause 8.

NOTE Newly identified hazards, not already reflected in a safety goal, are usually non-functional hazards. If those non-functional hazards are outside the scope of ISO 26262 then it is recommended that they are 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 is allowed for reference purposes.

7.4.4 Verification of hardware design

7.4.4.1 The hardware design shall be verified for compliance and completeness with regard to the hardware safety requirements. To achieve this, the methods listed in Table 3 shall be considered:

Table 3 — Hardware design verification

Methods		ASIL			
		A	B	C	D
1a	Hardware design inspection[a]	+	+	++	++
1b	Hardware design walkthrough [a]	++	++	o	o
2	Safety analyses	See 7.4.3			
3a	Emulation by simulation[b]	o	+	+	+
3b	Development by prototype hardware	o	+	+	+

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

[b] Methods 3a and 3b serve as a check of particular points of hardware design for which analytical methods 1 and 2 are not considered sufficient. It can be used among other as a fault injection technique.

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

7.4.5 Requirements for production and operation

7.4.5.1 Safety-related special characteristics shall be specified if safety analyses have shown them to be relevant. Attributes of safety-related special characteristics shall include:

a) The verification measures for production and operation

b) The acceptance criteria for the measures.

7.4.5.2 Instructions for assembly, disassembly and decommissioning of safety-related hardware elements shall be issued, if these operations can impact the safety concept.

7.4.5.3 The traceability of safety-related hardware elements shall be ensured.

NOTE This can include adequate labelling.

7.4.5.4 Instructions for maintenance of safety-related hardware elements shall be issued, if maintenance can impact the safety concept.

7.5 Work products

7.5.1 Hardware design specification resulting from requirements 7.4.1, 7.4.2.

7.5.2 Hardware safety analysis report resulting from requirement 7.4.3.

7.5.3 Hardware design verification report resulting from requirement 7.4.4.

7.5.4 Requirements for production and operation resulting from requirement 7.4.5.

8 Hardware architectural metrics

8.1 Objectives

The objective of this clause is to evaluate the hardware architecture of the item against the requirements for fault handling as represented by the hardware architectural metrics.

8.2 General

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

These metrics and associated target values apply to the overall hardware architecture of the item and are complementary to the residual risk assessment described in Clause 9.

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

Compliance with the target figures prescribed for the hardware architectural metrics is achieved for each safety goal in which the item is involved.

Some or all of the applicable safety goals can be considered together for the determination of the metrics; but in this case the metrics' targets to be considered are those of the safety goal with the highest ASIL. These hardware architectural metrics are defined to achieve the following objectives:

— Be objectively assessable: metrics are verifiable, unambiguous, reproducible and precise enough to differentiate between different architectures;

— Support evaluation of the final design (the precise calculations are done with the detailed hardware design);

— Make available ASIL dependent pass/fail criteria;

— Reveal whether or not the coverage of the safety mechanisms, to control hardware faults in the E/E architecture, is sufficient;

— Reveal if the coverage of the safety mechanisms to prevent risk from latent multiple point faults in the hardware E/E architecture is sufficient;

— Address single point faults, residual faults and latent multiple point faults;

— Be robust concerning uncertainty of hardware failures rates;

— Be limited to safety-related elements; and

— Be usable on different elements levels, e.g. target values can be assigned at supplier's perimeter level for supplier's interfaces.

8.3 Inputs of this clause

8.3.1 Prerequisites

The following information shall be available:

— Hardware safety requirements specification (see 6.5.1)

— Hardware architectural metrics requirements (see 6.5.2)

— Hardware design specification (see 7.5.1)

— Hardware safety analysis report (see 7.5.2)

8.3.2 Further supporting information

The following information may be considered:

— Technical safety concept (ISO 26262-4:—, 7.5.1)

— System design specification (ISO 26262-4:—, 7.5.2)

8.4 Requirements and recommendations

8.4.1 General

The requirements of this Clause refer only to the ASIL of the safety goal.

8.4.2 This requirement applies to ASIL (B), C, and D of the safety goal, in accordance with 4.3: The concepts of diagnostic coverage, single point faults metric and latent fault metric, as defined in Annex C, shall be considered for the following requirements (8.4.3 to 8.4.9).

8.4.3 This requirement applies to ASIL (B), C, and D of the safety goal, in accordance with 4.3: The coverage of safety-related hardware elements by safety mechanisms shall be estimated with regard to residual faults and with regard to latent multiple point faults.

NOTE For this purpose, the Tables D.1 to D.12 can be used.

8.4.4 This requirement applies to ASIL (B), C, and D of the safety goal, in accordance with 4.3: The failure rates of safety-related hardware parts shall be estimated in accordance with 9.4.3.6.

8.4.5 If sufficient evidence of the failure rate used in the calculation of a single point fault or latent multiple point fault cannot be made available, alternative means should be proposed (e.g. add safety mechanisms to detect and control this fault).

8.4.6 This requirement applies to ASIL (B), C, and D of the safety goal, in accordance with 4.3. For each safety goal, numerical target values for "single point faults metric" and "latent faults metric" shall be based on one of the following sources of reference target values:

a) Derived from the safety hardware architectural metrics calculation applied on similar well-trusted designs; or

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

b) Derived from Annex E.

These numerical targets are intended to:

- Make available design guidance as described in 8.2.
- Make available evidence that the design satisfies the safety goals.

8.4.7 This requirement applies to ASIL (B), C, and D of the safety goal, in accordance with 4.3. For each safety goal, the overall hardware architecture of the item shall meet the target values of the "single point faults metric.

NOTE 1 The compliance with requirement 8.4.7 can be achieved with the following procedure:

1) Prescribe at the hardware element level, appropriate targets sufficient to comply with the metrics' target values given in requirement 8.4.4 assigned to the overall hardware architecture, and

2) Make available a rationale for compliance with these targets at the hardware element level.

NOTE 2 If an item contains different kinds of hardware elements with significantly different failure rate levels, compliance with the hardware architectural metrics could only focus on the kind of hardware elements with the highest magnitude of failure rates. (One example where this can be used is for the single point faults 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 3 Some or all of the applicable safety goals can be considered together for the determination of the metrics; but in this case the metrics' targets to be considered are those of the safety goal with the highest ASIL.

NOTE 4 If the target is not met, the rationale for how the safety goal is achieved will be available in the safety case.

EXAMPLE A rationale can be the compliance with the state of the art.

8.4.8 This requirement applies to ASIL (B), C, and D of the safety goal, in accordance with 4.3. For each safety goal, the overall hardware architecture of the item shall meet one of the following targets:

a) target values of the "latent faults metric";

b) when each safety mechanism of the item is based on fault detection, DC with regard to Latent Multiple Point Faults of each hardware element with fault(s) that could lead to the unavailability of a safety mechanism (to prevent a fault from violating the safety goal) compliant with the target value given in requirement 8.4.6 for the "latent faults metric" (treated as a diagnostic coverage).

NOTE 1 Approach b) can only be considered when each safety mechanism is based on fault detection. If this condition is not met then approach a) is the only possibility.

NOTE 2 In the case b), a metric is not calculated, only the coverage of the hardware elements by safety mechanisms with regard to latent multiple point faults is evaluated.

NOTE 3 The compliance with requirement 8.4.8 can be achieved with the following procedure:

1) Prescribe at the hardware element level, appropriate targets sufficient to comply with the metrics' target values given in requirement 8.4.4 assigned to the overall hardware architecture, and

2) Make available a rationale for compliance with these targets at the hardware element level.

NOTE 4 If an item contains different kinds of hardware elements with significantly different failure rate levels, compliance with the hardware architectural metrics could only focus on the kind of hardware elements with the highest magnitude of failure rates. (One example where this can be used is for the single point 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 5 Some or all of the applicable safety goals can be considered together for the determination of the metrics; but in this case the metrics' targets to be considered are those of the safety goal with the highest ASIL.

NOTE 6 If the target is not met, the rationale for how the safety goal is achieved will be available in the safety case

EXAMPLE A rationale can be the compliance with the state of the art.

8.4.9 This requirement applies to ASIL (B), C, and D of the safety goal, in accordance with 4.3: A review of the result of the applied methods in 8.4.7 and 8.4.8 shall be performed.

NOTE The careful verification of the denominator for the single point faults metric is to ensure that failure rates of safety-related hardware elements, without any single point fault and residual fault, do not falsify the metric.

8.5 Work products

8.5.1 Assessment of the effectiveness of the system architecture to cope with the hardware random failures resulting from requirements 8.4.2 to 8.4.9.

8.5.2 Review report of assessment of the effectiveness of the system architecture to cope with the hardware random failures resulting from requirement 8.4.9.

9 Evaluation of violation of the safety goal due to random HW failures

9.1 Objectives

The objective of the requirements in this clause is to make available criteria that can be used in a rationale that the residual risk of safety goal violation, due to random hardware failures of the item, is sufficiently low.

NOTE 'Sufficiently low' means "comparable to accepted risks on similar items already in use".

9.2 General

Two alternative methods are proposed to evaluate whether the residual risk of violation of safety goals is acceptable.

Both methods evaluate the residual risk of violating a safety goal due to single point faults, residual faults, and plausible dual point faults. In this analysis, coverage of safety mechanisms and exposure duration in case of a dual point fault will be considered.

The first method consists of using a probabilistic metric to evaluate violation of the considered safety goal (using for instance quantified FTA) and compares the result of this quantification with a target value.

The second method consists of the individual evaluation of each residual and single point fault and of each dual point failure leading to the violation of the considered safety goal. This analysis method can also be considered to be a cut-set analysis.

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 parts. For electromechanical parts, only the electrical failure modes and failure rate are considered.

9.3 Inputs to this clause

9.3.1 Prerequisites

The following information shall be available:

— Random hardware failure requirements (see 6.5.3);

— Hardware design specification (see 7.5.1);

— Hardware safety analysis report (see 7.5.2).

9.3.2 Further supporting information

The following information may be considered:

— Technical safety concept (ISO 26262-4:—, 7.5.1);

— System design specification (ISO 26262-4:—, 7.5.2);

— Hardware safety requirements specification (see 6.5.1).

9.4 Requirements and recommendations

9.4.1 General

The requirements of this Clause apply to the ASIL of the safety goal.

9.4.2 This requirement applies to ASIL (B), C and D of the safety goal, in accordance with 4.3: The item shall comply with one of the following sets of requirements:

a) Requirements 9.4.3;

b) Requirements 9.4.4.

9.4.3 Probabilistic metric for random hardware failures

9.4.3.1 This requirement applies to ASIL (B,) C, and D of the safety goal, in accordance with 4.3. Quantitative target values for maximum probability of violation of each safety goal due to hardware random failures as required in ISO 26262-4:—, 7.4.4.3 shall be defined using one of the following sources of reference target values:

a) Derived from quantitative analysis techniques applied on similar well-trusted designs using well known failure rate databases;

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

NOTE 2 The numbers do not have any absolute significance and are only useful to compare a new design with existing ones.

b) Derived from field data of similar well-trusted designs;

NOTE 3 In this case, the quantitative values can have an absolute significance.

c) Derived from Annex G.

These numerical targets are intended to:

— Make available design guidance as described in 9.1.

— Make available evidence that the design satisfies the safety goals.

NOTE 4 These target values described in Annex G might be adapted to fit specific uses of the item (for instance if the item is able to violate the safety goal for durations longer than the typical use of a passenger car).

9.4.3.2 This requirement applies to ASIL (B,) C, and D of the safety goal, in accordance with 4.3. Target values of requirement 9.4.3.1 shall be expressed in terms of average probability per hour over the operational lifetime of the item.

9.4.3.3 This requirement applies to ASIL (B,) C, and D of the safety goal, in accordance with 4.3. A quantitative analysis of the hardware architecture shall provide evidence that target values of requirement 9.4.3.1 are achieved. This quantitative analysis shall consider:

a) Single point faults, residual faults and dual point faults;

b) The architecture of the item;

c) The estimated failure rate of each hardware part in any failure modes which would cause a single point fault of the item;

d) The estimated failure rate of each hardware part with respect to its dual point faults;

e) The diagnostic coverage of safety-related hardware elements by safety mechanisms;

f) The exposure duration in case of multiple point faults; and

g) The remaining dependent failures due to random hardware faults.

NOTE 1 For safety mechanisms using integrated diagnostics Tables D.1 to D.12 can be used to evaluate the coverage rate of these safety mechanisms.

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

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

— The maximum duration of a trip (driver requested to stop in a safe way);

— The average duration until the vehicle is at the workshop for repair (driver alerted to have the vehicle repaired);

So exposure duration depends on the type of monitoring involved (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 that the driver will have his (or her) vehicle repaired in the following average vehicle trips:

— 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 driving disturbing behaviour;

— One vehicle trip for red warning lights.

Time taken to 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 1h.

NOTE 3 Dependent failures analysis as described in ISO 26262-9:—, Clause 7 is basically a qualitative analysis leading to the suppression or reduction of harmful dependencies.

NOTE 4 Multiple point faults of higher order than two are considered if shown relevant in the safety concept.

NOTE 5 Situations when the item is in power down mode are not included in the calculation of the average probability per hour.

NOTE 6 If the target is not met, the rationale for how the safety goal is achieved will be available in the safety case.

EXAMPLE A rationale can be the compliance with the state of the art.

9.4.3.4 This requirement applies to ASIL C and D of the safety goal of the safety goal, in accordance with 4.3: A single point fault occurring in a hardware part, shall only be considered acceptable if dedicated measures are taken to ensure the accuracy of its estimated failure rate used in the analysis of 9.4.3.3.

NOTE Dedicated measures can be among others:

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 plan.

9.4.3.5 This requirement applies to ASIL C and D of the safety goal, in accordance with 4.3. A residual fault occurring in a hardware part with a diagnostic coverage (with regard to residual faults) lower than 90 %, shall be considered as acceptable only if the corresponding hardware part is dealt with dedicated measures (the note in 9.4.3.4 gives examples of dedicated measures) to justify the failure rate claimed in the analysis of 9.4.3.3.

9.4.3.6 This requirement applies to ASIL (B,) C, and D of the safety goal, in accordance with 4.3. The estimated failure rates for hardware parts used in the analyses shall be determined either:

a) Using hardware part failure rates data from a recognised industry source;

EXAMPLE Commonly recognised industry sources to determine hardware part failure rate and failure mode distribution are IEC 62380, IEC 61709, MIL HDBK 217 F notice 2, RAC HDBK 217 Plus, NPRD95, EN50129 Annex C, EN 62061 Annex D, RAC FMD97 and MIL HDBK 338.

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

b) Using statistics based on field returns or tests. In this case, the estimated failure rate should have an adequate confidence level; or

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

NOTE 2 These criteria can consider field experience, testing, reliability analysis, and novelty of design.

NOTE 3 Different sources of failure rates (a, b or c) can be used for the different hardware parts involved in an analysis.

9.4.3.7 This requirement applies to ASIL (B,) C, and D of the safety goal, in accordance with 4.3. In order to avoid bias in the quantification, if failure rates from multiple sources are combined, they shall be scaled to

be consistent. Scaling is possible if a rationale for the factor between two failure rates sources is available, for instance if sufficient data exists about a "predecessor" system whose failure rate can be considered representative of that expected for the item under consideration.

For the three failure rate sources in 9.4.3.1, a, b and c, let $T \in \{a, b, c\}$ be the source of failure rate data for the target and let $\lambda_{representative.T}$ be the calculated overall failure rate for the representative system using T as the data source. For the three failure rate sources in 9.4.3.1, a, b and c, let $\lambda_{representative.A}$ be the calculated overall failure rate for the representative system using an alternative source of data $A \in \{a, b, c\}$. Then A can be employed as a source of failure rate data to calculate targets for an item under consideration based on T if each failure rate using A is multiplied by the scaling factor $\pi_{T,A} = \lambda_{representative.T} / \lambda_{representative.A}$, that is, if $\lambda_{i,A}$ is a component failure rate obtained from data source A, then $\pi_{T,A} \times \lambda_{i,A}$ can be used as a component failure rate to calculate a target value based on T.

Combining failure rates from multiple sources is only allowed if the rates can be made consistent.

EXAMPLE From a previous design, calculated failure rates from a data handbook and warrantee data have been obtained. We then know that $\lambda_{handbook} / \lambda_{warrantee} = \pi$

where $\lambda_{handbook}$ is the calculated failure rates from a data handbook,

$\lambda_{warrantee}$ is the calculated failure rates from warrantee data and

π is the resulting scaling factor.

For instance, an hardware element with $\lambda_{handbook} = 10^{-8} / h$, has $\lambda_{warrantee} = 2 \cdot 10^{-9} / h$. In this case $\pi = 10^{-8} / 2 \cdot 10^{-9}$.

If in a new design, we use the handbook data to determine the failure rates except for one hardware element (hardware element 1) for which we have warrantee data, $\lambda_{1(handbook)} = \lambda_{1(warrantee)} * \pi$

Where $\lambda_{1(handbook)}$ is the failure rate of the hardware element 1 using handbook data and

$\lambda_{1(warrantee)}$ is the failure rate of the hardware element 1 using warrantee data.

For instance, if $\lambda_{1(warrantee)} = 9 \cdot 10^{-9} / h$, then $\lambda_{1(handbook)}$ can be calculated as $9 \cdot 10^{-9} * \left(10^{-8} / 2 \cdot 10^{-9}\right) = 4{,}5 \cdot 10^{-9} / h$

Using this $\lambda_{1(handbook)}$, a consistent evaluation of the violation of the safety goal due to random hardware failures can be done.

9.4.3.8 Table 4 shows an example of combinations of target values and failure rates.

NOTE 1 If the source of data for the target and the new component failure rates is similar, then no scaling is necessary.

NOTE 2 The targets of Annex G are similar to handbook data.

Table 4 — Example showing combinations of sources of target values and failure rates to produce consistent failure rates for use in calculations.

		Data source for Target Value		
		Calculated value 9.4.3.1 a	Field data 9.4.3.1 b	Annex G 9.4.3.1 c
Data source for failure rates of hardware parts	Std. Database 9.4.3.6 a	(1)	$\pi_{b,a} \times \lambda_{i,b}$	$\lambda_{i,a}$
	Statistics 9.4.3.6 b	(1)	$\lambda_{i,b}$	$\pi_{a,b} \times \lambda_{i,b}$
	Expert judgment 9.4.3.6 c	(1)	$\pi_{b,c} \times \lambda_{i,c}$	$\pi_{a,c} \times \lambda_{i,c}$

(1) Failure rates should have the same origin as those used to calculate the target value

NOTE 1 $\pi_{T,A} = \lambda_{representative.T} / \lambda_{representative.A}$ where $\lambda_{representative.X}$ is the collective failure rate for a system whose data source is *X* and $\pi_{T,A}$ *is the scaling factor between data source T and data source A.*

NOTE 2 $\lambda_{i,X}$ is the failure rate of component *i* using *X* as a data source

9.4.4 Evaluation of each remaining cause of violation of safety goals

9.4.4.1 This second method is illustrated by flowchart in Figure 3. Each single point fault is evaluated using criteria on occurrence of the fault. Each residual fault is evaluated using criteria combining occurrence of the fault and efficiency of the safety mechanism.

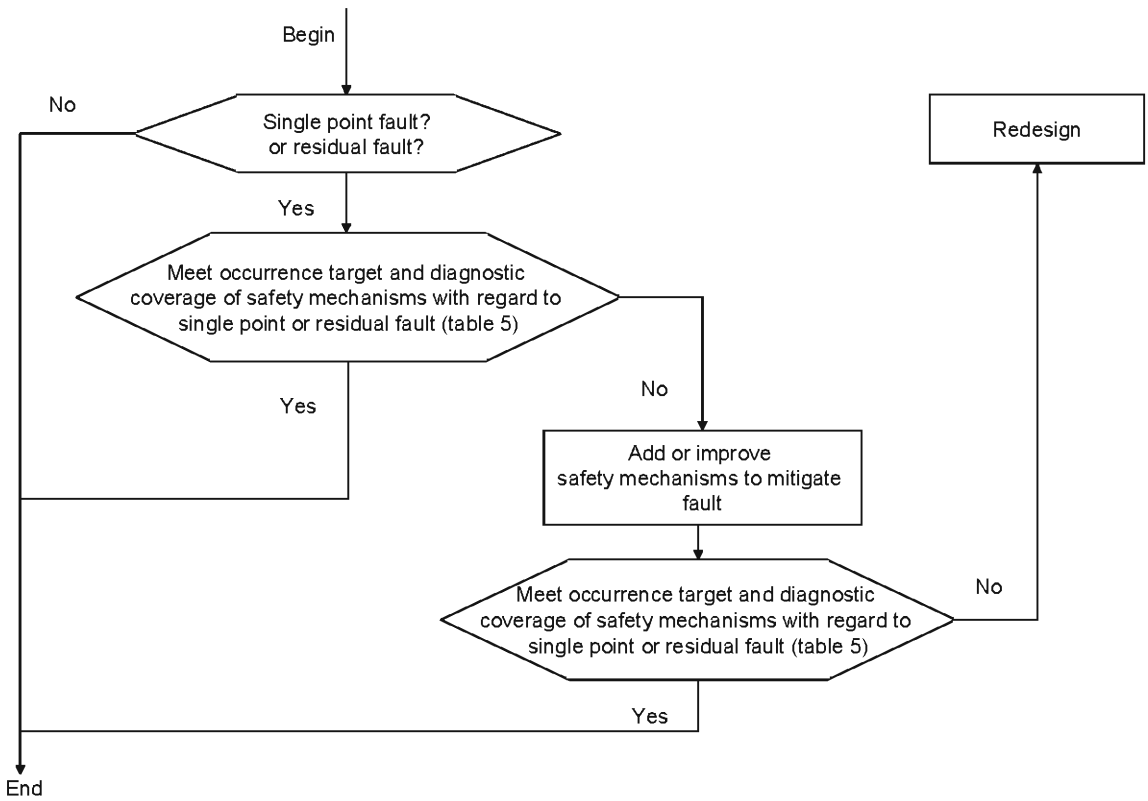


Figure 3 — Evaluation procedure for single point and residual faults

The procedure to be applied for dual point failures is illustrated by flowchart in Figure 4. Each dual point failure is first evaluated regarding its plausibility. A dual point failure is considered not plausible 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.

The evaluation procedures described in Figures 3 and 4 apply to the hardware parts (transistors, integrated circuits, etc) level.

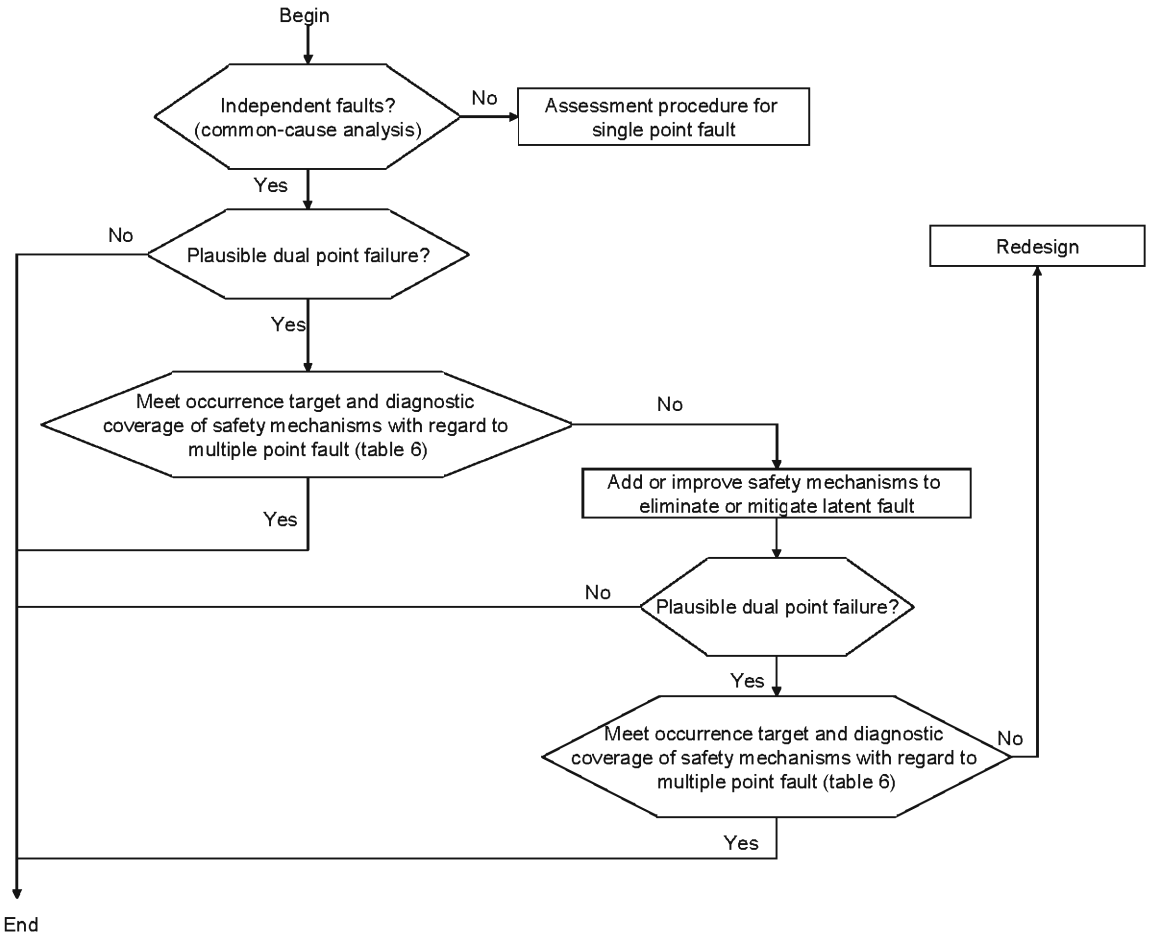


Figure 4 — Evaluation procedure for dual point failures

9.4.4.2 This requirement applies to ASIL (B,) C, and D of the safety goal, in accordance with 4.3. 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 show that each single point fault, residual fault and dual point failure violating the considered safety goal is acceptable in accordance with requirements 9.4.4.3 to 9.4.4.12.

NOTE This analysis can be viewed as review of cut sets where imperfect coverage is treated as a fault.

NOTE 2 Multiple point failures of higher order than two are considered if shown relevant in the safety concept.

9.4.4.3 This requirement applies to ASIL (B,) C, and D of the safety goal, in accordance with 4.3. The failure rate class ranking for a hardware part failure rate shall be determined as follows:

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

NOTE 1 The target values given in Annex G can be used.

NOTE 2 If it can be justified that the number of cut sets is lower than 100, the failure rate class ranking can be adapted – the ASILD target not being divided by 100 anymore. In this case, it is necessary to pay attention in order to keep proper ranking considering the single point faults, residual faults and higher degree cutsets together.

b) The failure rate corresponding to failure rate class 2 shall be less than ten times the failure rate corresponding to failure rate class 1;

c) The failure rate corresponding to failure rate class 3 shall be less than a hundred times the failure rate corresponding to failure rate class 1.

NOTE 3 Failure rate class 1, respectively 2 and 3 could be considered to be equivalent to occurrence level 1, respectively 2 and 3 in a FMEA.

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

NOTE 4 The failure rate class assignment is based upon the hardware part failure rate.

NOTE 5 For the case where a small number of components have failure rates slightly higher than failure rate class *i*, then the failure rate for failure rate class *i* can be assigned higher so as to include these components in failure rate class *i* if the resulting average failure rate corresponds to failure rate class *i*. A failure rate slightly higher than failure rate class *i* cannot be equal to or higher than failure rate class *i+1*.

9.4.4.4 This requirement applies to ASIL D of the safety goal, in accordance with 4.3. A single point fault occurring in a hardware part shall only be considered as acceptable, if the corresponding hardware part failure rate is ranked failure rate class 1 and dedicated measures are taken to ensure this occurrence level.

NOTE The note under 9.4.3.4 gives examples of dedicated measures.

9.4.4.5 This requirement applies to ASIL C of the safety goal, in accordance with 4.3: A single point fault occurring in a hardware part shall be considered as acceptable, if the corresponding hardware part failure rate is either:

a) Ranked failure rate class 2 and dedicated measures shall be taken to ensure this failure rate; or

b) Ranked failure rate class 1.

NOTE The note under 9.4.3.4 gives examples of dedicated measures.

9.4.4.6 This requirement applies to ASIL (B) of the safety goal, in accordance with 4.3: A single point fault occurring in a hardware part shall be considered as acceptable, if the corresponding hardware part failure rate is ranked failure rate class 2 or failure rate class 1.

9.4.4.7 This requirement applies to ASIL (B,) C, and D of the safety goal, in accordance with 4.3: A residual fault occurring in a hardware part shall be considered as acceptable if the failure rate class ranking and the diagnostic coverage (with regard to residual faults) of the corresponding hardware part comply with the targets given in Table 5. For failure rate classes i, i > 3, a residual fault shall be considered as acceptable if the diagnostic coverage is equal to (100% - $10^{(3-i)}$).

NOTE 1 Failure rate to be considered is the unmitigated occurrence of the hardware part failure. Therefore, it does not consider effectiveness of safety mechanisms.

NOTE 2 Table 5 specifies the maximum failure rate class allowed given the target ASIL level and the level of diagnostic coverage achieved. Lower failure rate classes are acceptable but not required.

Table 5 — Targets of failure rate class and coverage of hardware part regarding residual faults

		Diagnostic Coverage wrt. residual faults		
		>= 99 %	>= 90 %	< 90 %
ASIL of Safety Goal	D	Failure rate class 3 ++	Failure rate class 2 ++	Failure rate class 1 + dedicated measures[a] ++
	C	o	Failure rate class 3 ++	Failure rate class 2 + dedicated measures[a] ++
	B	o	Failure rate class 3 +	Failure rate class 2 +
[a] The note in requirement 9.4.3.4 gives examples of dedicated measures				

9.4.4.8 This requirement applies to ASIL D of the safety goal, in accordance with 4.3. A dual point failure shall be considered plausible if:

a) One or both hardware parts involved has a diagnostic coverage (with regard to the latent multiple point 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.

9.4.4.9 This requirement applies to ASIL C of the safety goal, in accordance with 4.3. A dual point failure shall be considered plausible if:

a) One or both hardware parts involved has a diagnostic coverage (with regard to the latent multiple point faults) of less than 60 %; 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.

9.4.4.10 This requirement applies to ASIL C and D of the safety goal, in accordance with 4.3. A dual point failure that is not plausible shall be considered acceptable.

9.4.4.11 This requirement applies to ASIL C and D, in accordance with 4.3. A dual point fault occurring in a hardware part and contributing to a plausible dual point failure shall be considered as acceptable if corresponding hardware part complies with the targets for failure rate class ranking and diagnostic coverage (with regard to latent multiple point faults) given in Table 6.

NOTE 1 Failure rate to be considered is the unmitigated occurrence of the hardware part failure. Therefore, it does not consider effectiveness of safety mechanisms.

NOTE 2 Table 6 specifies the maximum failure rate class allowed given the target ASIL level and the level of diagnostic coverage achieved. Lower failure rate classes are acceptable but not required.

Table 6 — Targets of failure rate class and coverage of hardware part regarding dual point faults

		Diagnostic Coverage wrt. latent multiple point faults		
		>= 99 %	>= 90 %	< 90 %
ASIL of Safety Goal	D	o	Failure rate class 3 ++	Failure rate class 2 ++
	C	o	o	Failure rate class 3 ++

9.4.4.12 This requirement applies to ASIL (B,) C, and D of the safety goal, in accordance with 4.3. The failure rate class ranking of the hardware part failure rate used in the analyses shall be justified by using sources of failure rates described in 9.4.3.6.

9.4.4.13 This requirement applies to ASIL (B,) C, and of the safety goal D, in accordance with 4.3. A review of the results of the applied method shall be performed.

9.5 Work products

9.5.1 Evaluation of random hardware failures resulting from requirement 9.4.3 or requirement 9.4.4.

9.5.2 Specification of dedicated measures, if needed, resulting from requirements 9.4.3.4, 9.4.3.5, 9.4.4.4 and 9.4.4.5.

9.5.3 Review report of evaluation of violation of the safety goal due to random HW failures resulting from requirement 9.4.3 or 9.4.4.

10 Hardware integration and testing

10.1 Objectives

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

The requirements in 10.4.1 to 10.4.7 apply to the hardware of an element.

10.2 General

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

Hardware integration and testing differ from the qualification of hardware components activity of ISO 26262-8:—, Clause 13. ISO 26262-8:—, Clause 13 gives evidence of suitability of intermediate level hardware components and parts for their use as parts of items, systems or elements developed in compliance with ISO 26262 concerning their functional behaviour and their operational limitations. ISO 26262-8:—, Clause 13 also gives relevant information regarding their failure modes and their distribution, and their ability for diagnostic with regard to the safety concept for the item.

10.3 Inputs of this clause

10.3.1 Prerequisites

The following information shall be available:

— Overall project plan (refined) (see 5.5.1)

— Safety plan (refined) (see 5.5.2)

— Item integration and testing plan (refined) (see ISO 26262-4:—, 7.5.4)

— Hardware safety requirements specification (see 6.5.1)

10.3.2 Further supporting information

The following information may be considered:

— Hardware design specification (see 7.5.1)

— Hardware safety analysis report (see 7.5.2)

10.4 Requirements and recommendations

10.4.1 Hardware integration and testing shall be planned and specified in accordance with the safety plan and with ISO 26262-8:—, Clause 9.

10.4.2 Hardware integration and testing activities shall be executed in accordance with the item integration and testing plan.

10.4.3 In the case of change, an impact analysis on test strategy shall be carried out in accordance with ISO 26262-8:—, Clause 8.

10.4.4 Test equipment shall be calibrated in accordance with international standards (e.g. ISO 17025) or company standards.

10.4.5 To demonstrate appropriate specification of test cases for the selected hardware integration test methods, test cases shall be derived using an appropriate combination of methods listed in Table 7.

Table 7 — 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 common cause	+	+	++	++
1h	Analysis of environmental conditions and operational use cases	+	++	++	++
1i	Standards if existing[d]	+	+	+	+
1j	Analysis of significant variants[e]	++	++	++	++

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

[b] EXAMPLE values approaching and crossing the boundaries between specified values and out of range values

[c] "Error guessing tests" should be based on data collected through lesson learned process, or expert judgment, or both. It can be supported by FMEA for example.

[d] Existing standards can be ISO 16750 and ISO 11452.

[e] The analysis of significant variants includes worst case analysis.

10.4.6 The hardware integration and testing activities shall verify the completeness and correctness of the implementation of safety mechanisms with respect to hardware safety requirements.

To achieve these objectives, the methods listed in Table 8 shall be considered.

Table 8 — Hardware integration tests to verify completeness and correctness of safety mechanisms implementation with respect to 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 guaranteeing that 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 regard to the specification and indications of an incomplete specification are analysed.

[b] Fault injection testing aims at introducing faults in the hardware product and analysing the response. This testing is appropriate whenever a safety mechanism is defined.

[c] Electrical testing aims at verifying the compliance with hardware safety requirements within the specified (static and dynamic) voltage range.

10.4.7 The hardware integration and testing activities shall verify robustness of hardware against external stresses.

To achieve these objectives, the methods listed in Table 9 shall be considered.

Table 9 — Hardware integration tests

Methods		ASIL			
		A	B	C	D
1	Functional testing under environmental conditions[a]	++	++	++	++
2a	Expanded functional testing[b]	o	+	+	++
2b	Statistical testing[c]	o	o	+	++
2c	Worst case testing[d]	o	o	o	+
2d	Over limit testing [e]	+	+	+	+
3a	Mechanical testing	++	++	++	++
3b	Environmental testing[f]	++	++	++	++
3c	Accelerated life test[g]	+	+	++	++
3d	Mechanical Endurance test[h]	++	++	++	++
4	EMI test[i]	++	++	++	++
5	Chemical testing[j]	++	++	++	++

[a] During functional testing under environmental conditions the hardware is put under various environmental conditions during which the hardware requirements are assessed.

[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 major failure or extreme mission profile values), or that are outside the specification of the hardware (for instance 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 estimate.

[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. 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 increased progressively to values much more severe than specified and up to functioning limit or hardware product destruction. The purpose of this test is to determine the margin of robustness of the element under test with regard to the required performance.

[f] For environmental test, ISO 16750-4 (Road vehicles -- Environmental conditions and testing for electrical and electronic equipment -- Part 4: Climatic loads) can be applied.

[g] Highly accelerated life test aims at predicting the behaviour evolution of a product in its normal operational conditions by submitting it to constraints higher than constraints expected during its mission profile. In order to achieve its goals, the accelerated testing is based on analytical model of failure modes acceleration.

[h] These tests are only applicable to electromechanical parts. The aim of these tests is to study the mean time to failure or the limit number of cycles. Test can be performed up to failure or by damage evaluation.

[i] For EMI test, ISO 11452-2; ISO 11452-4; ISO 7637-2; ISO 10605 and ISO 16750-2 can be applied.

[j] For chemical test, ISO 16750–5 can be applied.

10.5 Work products

10.5.1 Hardwareintegration and verification report resulting from requirements 10.4.2 to 10.4.8.

Annex A
(informative)

Overview on and document flow 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.

Table A.1 — Product development at the hardware level: Overview

Clause	Title	Objectives	Prerequisites	Work products
5	Initiation of product development at the hardware level	The objective of the initiation of the product development for the hardware is to determine and plan the functional safety activities during the individual sub-phases of hardware development. This also includes the necessary supporting processes described in ISO 26262-8:— "Supporting processes". This planning of hardware-specific safety activities is included in the safety plan.	— Overall project plan (refined) (ISO 26262-4:— work product 5.5.1) — Safety plan (refined) (ISO 26262-4:— work product 5.5.2) — Item integration and testing plan (refined) (ISO 26262-4:— work product 7.5.4)	5.5.1 Overall project plan (refined) 5.5.2 Safety plan (refined)
6	Specification of hardware safety requirements	The first objective of this clause is to make available a consistent and complete hardware specification that will be applied to the hardware of the item or element under consideration. The requirements of this specification are hardware safety requirements. The second objective is to verify that the hardware safety requirements are consistent with the technical safety concept. A further objective of this phase is to detail the hardware-software interface (HSI) requirements initiated in ISO 26262-4:—, Clause 7.	— Overall project plan (refined) (see 5.5.1) — Safety plan (refined) (see 5.5.2) — Technical safety concept (see ISO 26262-4:—, 6.5.1) — System design specification (see ISO 26262-4:—, 7.5.2) — Hardware software interface specification (see ISO 26262-4:—, 7.5.6)	6.5.1 Hardware safety requirements specification 6.5.2 Hardware architectural metric requirements 6.5.3 Random hardware failure requirements 6.5.4 Hardware-software interface specification (refined) 6.5.5 Hardware safety requirements verification report
7	Hardware design	The first objective of this clause is to design the hardware with respect to the system design specification and the hardware safety requirements. The second objective of this clause is to verify the hardware design against the system design specification and the hardware safety requirements.	— Hardware safety requirements specification (see 6.5.1) — Hardware software interface specification (refined) (see 6.5.5) — System design specification (ISO 26262-4:—, 7.5.2) — Overall project plan (refined) (see 5.5.1) — Safety plan (refined) (see 5.5.2)	7.5.1 Hardware design specification. 7.5.2 Hardware safety analysis report. 7.5.3 Hardware design verification report. 7.5.4 Requirements for production and operation.

Clause	Title	Objectives	Prerequisites	Work products
8	Hardware architectural metrics	The objective of this clause is to evaluate the hardware architecture of the item against the requirements for fault handling as represented by the hardware architectural metrics.	— Hardware safety requirements specification (see 6.5.1) — Hardware architectural metrics requirements (see 6.5.2) — Hardware design specification (see 7.5.1) — Hardware safety analysis report (see 7.5.2)	8.5.1 Assessment of the effectiveness of the system architecture to cope with the hardware random failures 8.5.2 Review report of assessment of the effectiveness of the system architecture to cope with the hardware random failures
9	Evaluation of violation of the safety goal due to random HW failures	The objective of the requirements in this clause is to make available criteria that can be used in a rationale that the residual risk of safety goal violation, due to random hardware failures of the item, is sufficiently low.	— Random hardware failure requirements (see 6.5.3) — Hardware design specification (see 7.5.1) — Hardware safety analysis report (see 7.5.2)	9.5.1 Evaluation of random hardware failures 9.5.2 Specification of dedicated measures 9.5.3 Review report of evaluation of violation of the safety goal due to random hardware failures
10	Hardware integration and testing	The objective of this clause is to ensure, by testing, the compliance of the developed hardware with the hardware safety requirements. The requirements in 10.4.1 to 10.4.7 apply to the hardware of an element.	— Overall project plan (refined) (see 5.5.1) — Safety plan (refined) (see 5.5.2) — Item integration and testing plan (refined) (see ISO 26262-4:—, 7.5.4) — Hardware safety requirements specification (see 6.5.1)	10.5.1 Hardware integration and verification report.

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 and using the flow diagram described in Figure B.2:

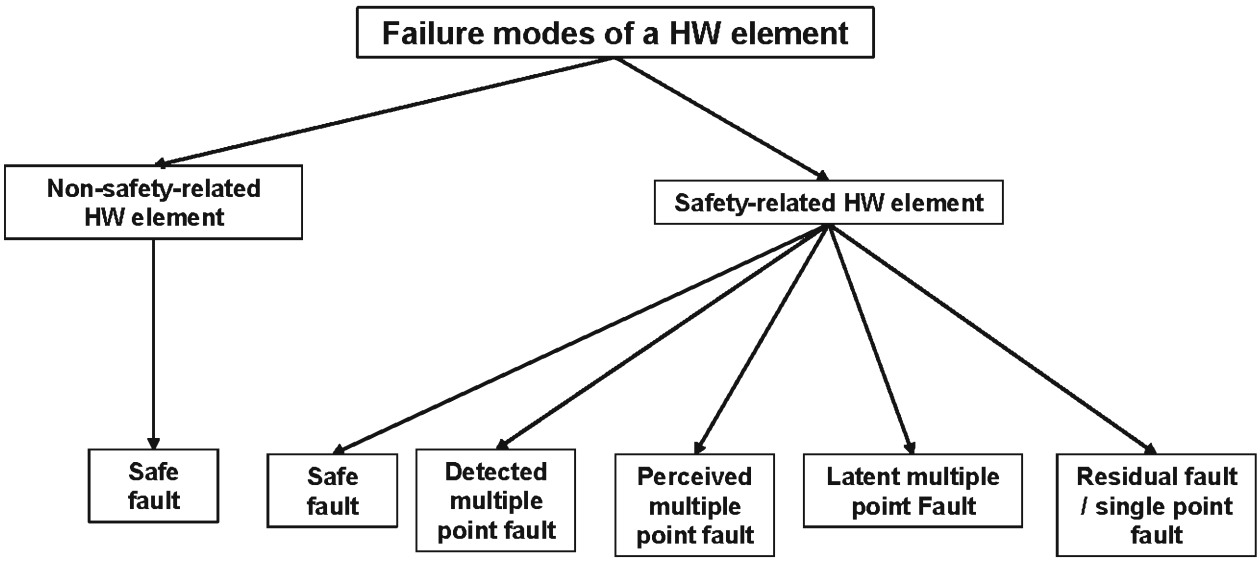


Figure B.1 — Failure modes classification of a hardware element

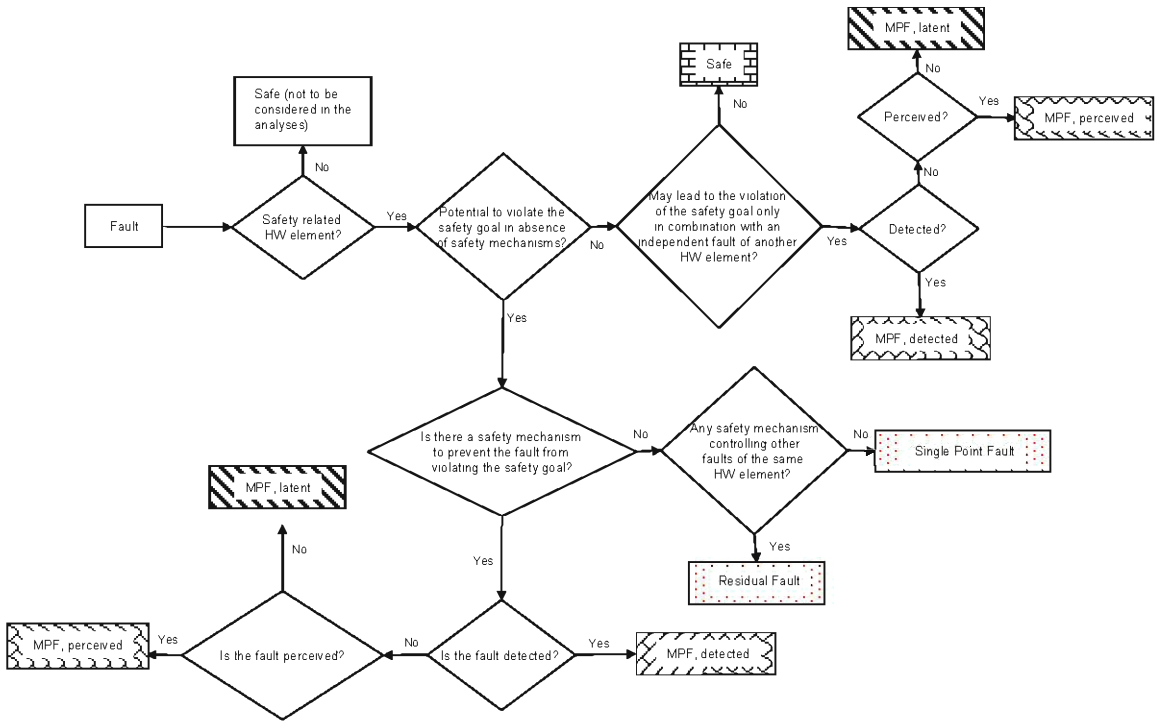


Figure B.2 — Example of flow diagram for fault classification

NOTE 1 Multiple point faults of distance strictly higher than n=2 are considered as safe faults unless shown relevant in the functional or technical safety concept.

NOTE 2 The background pattern of the boxes is related to the pattern of correspondent classes in the faults classification graphical representation of figure C.1 in Annex C.

Annex C
(normative)

Hardware architectural metrics

C.1 Diagnostic coverage

Hardware architectural metrics are defined for the overall hardware architecture of an item and address only safety-related elements.

Each fault occurring in a safety-related hardware element can be classified, as illustrated in Annex B, figure B.1, as:

— Single point fault: fault in an element which is not covered by a safety mechanism and where the fault leads directly to the violation of a safety goal

— Residual fault: portion of a fault which by itself leads to the violation of a safety goal, occurring in a hardware element, where that portion of the fault is not covered by existing safety mechanisms

— Multiple point fault: one fault of several independent faults that in combination, leads to a multiple point failure (either perceived, detected or latent)

— Safe fault: fault whose occurrence will not significantly increase the probability of violation of a safety goal

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

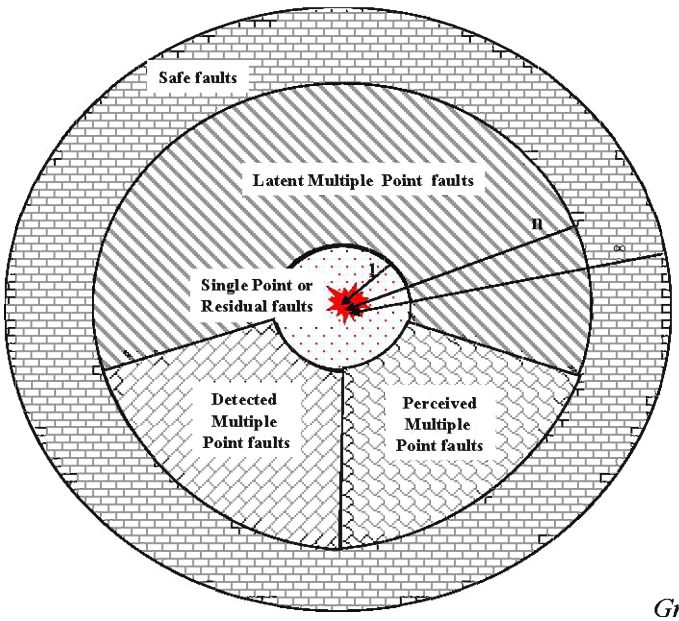


Figure C.1 — Faults classification of safety-related hardware elements of an item

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 are to be considered as safe faults unless shown relevant in the functional or technical safety concept.

NOTE 1 In 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 safe fault even if the driver is never informed about its existence.

The failure rate λ of each safety-related hardware element can therefore be split up as follows:

a) Failure rate associated to hardware element single point faults: λ_{SPF};

b) Failure rate associated to hardware element residual faults: λ_{RF};

c) Failure rate associated to hardware element multiple point faults: λ_{MPF};

1) Failure rate associated to hardware element perceived or detected multiple point faults: λ_{MPF_DP};

2) Failure rate associated to hardware element latent multiple point faults: λ_{MPF_L};

d) Failure rate associated to hardware element safe faults: λ_S

$$\text{with } \lambda = \lambda_{SPF} + \lambda_{RF} + \lambda_{MPF} + \lambda_S \text{ and } \lambda_{MPF} = \lambda_{MPF\_DP} + \lambda_{MPF\_L}.$$

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. The following equation gives therefore a conservative estimation of the failure rate associated to residual fault:

$$DC_{\text{with regard to residual faults}} : \text{Diagnostic Coverage in percentage}$$

$$DC_{\text{with regard to Residual Faults}} = \left(1 - \frac{\lambda_{RF}}{\lambda}\right) \times 100$$

$$\lambda_{RF} = \lambda \cdot \left(1 - \frac{DC_{\text{with regard to residual faults}}}{100}\right)$$

The failure rate assigned to latent multiple point faults can be determined using the diagnostic coverage of safety mechanisms that avoid latent multiple point faults of the hardware element. The following equation gives therefore a conservative estimation of the failure rate associated to latent multiple point faults:

$$DC_{\text{with regard to Latent Multiple Point Faults}} : \text{Diagnostic Coverage in percentage}$$

$$DC_{\text{with regard to Latent Multiple Point Faults}} = \left(1 - \frac{\lambda_{MPF\_L}}{\lambda}\right) \times 100$$

$$\lambda_{MPF\_L} = \lambda \cdot \left(1 - \frac{DC_{\text{with regard to Latent Multiple Point Faults}}}{100}\right)$$

NOTE 2 Annex D can be used to support evaluation of diagnostic coverage.

NOTE If the above estimations are considered too conservative, then a detailed analysis of the failure modes of the hardware element can classify each failure mode of the hardware element into one of the fault classes (single point faults, residual faults, multiple point faults (latent, detected or perceived) or safe faults) with regard to the considered safety goal and determine the failure rates apportioned to the failure modes. The flow diagram in annex B explains the flow that can be used to make the faults classification.

C.2 Single point faults metric

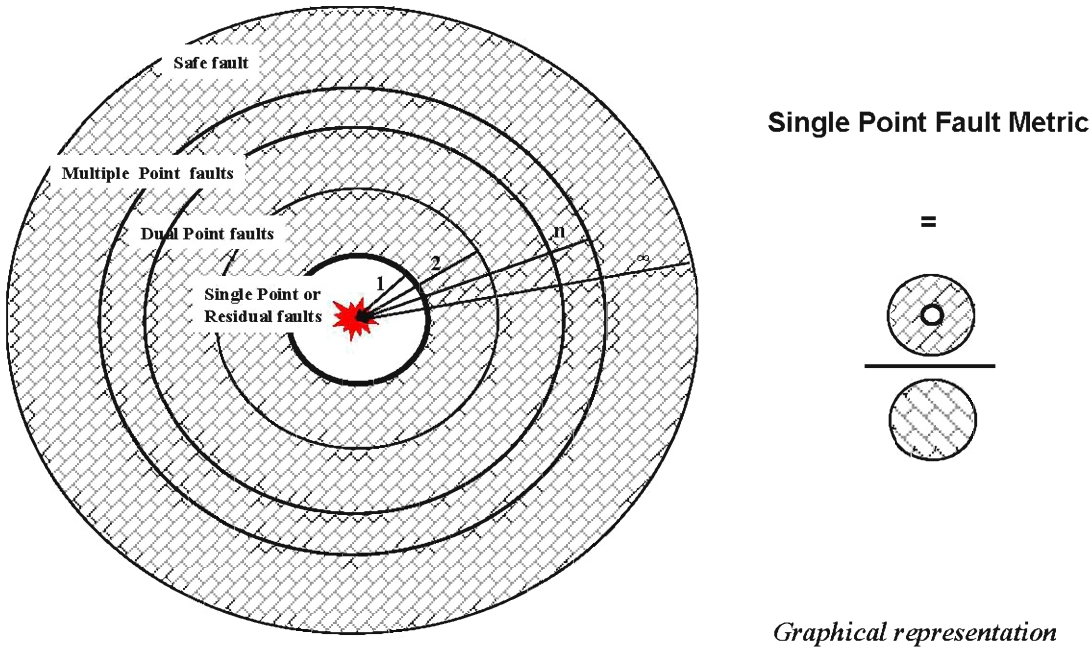
This metric reflects the robustness of the item to single point faults either by coverage from safety mechanisms or by design (primarily safe faults). A high single point faults metric implies that the proportion of single point faults and residual faults in the hardware is low. The definition is given by the following equation:

$$\text{Single Point Fault metric} = 1 - \frac{\sum_{\text{Safety related HW elements}} (\lambda_{SPF} + \lambda_{RF})}{\sum_{\text{Safety related HW elements}} \lambda} = \frac{\sum_{\text{Safety related HW elements}} (\lambda_{MPF} + \lambda_{S})}{\sum_{\text{Safety related HW elements}} \lambda}$$

where $\sum_{\text{safety related HW elements}} \lambda_x$ is the sum of λ_x of the safety-related hardware elements of the item.

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

NOTE 2 The following Figure C.2 gives a graphical representation of the single point faults metric.



Graphical representation

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

C.3 Latent faults metric

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 faults metric implies that the proportion of latent faults in the hardware is low.

The definition is given by the following equation:

$$\text{Latent Fault metric} = 1 - \frac{\sum_{\text{safety related HW elements}} (\lambda_{\text{MPF Latent}})}{\sum_{\text{safety related HW elements}} (\lambda - \lambda_{\text{SPF}} - \lambda_{\text{RF}})} = \frac{\sum_{\text{safety related HW elements}} (\lambda_{\text{MPF perceived or detected}} + \lambda_{\text{S}})}{\sum_{\text{safety related HW elements}} (\lambda - \lambda_{\text{SPF}} - \lambda_{\text{RF}})}$$

where $\sum_{\text{safety related HW elements}} \lambda_x$ is the sum of λ_x of the safety-related hardware elements of the item.

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

NOTE 2 The following Figure C.3 gives a graphical representation of the latent faults metric.

NOTE 3 An example of calculation of both metrics "single point faults metric" and "latent faults metric" is available in Annex F.

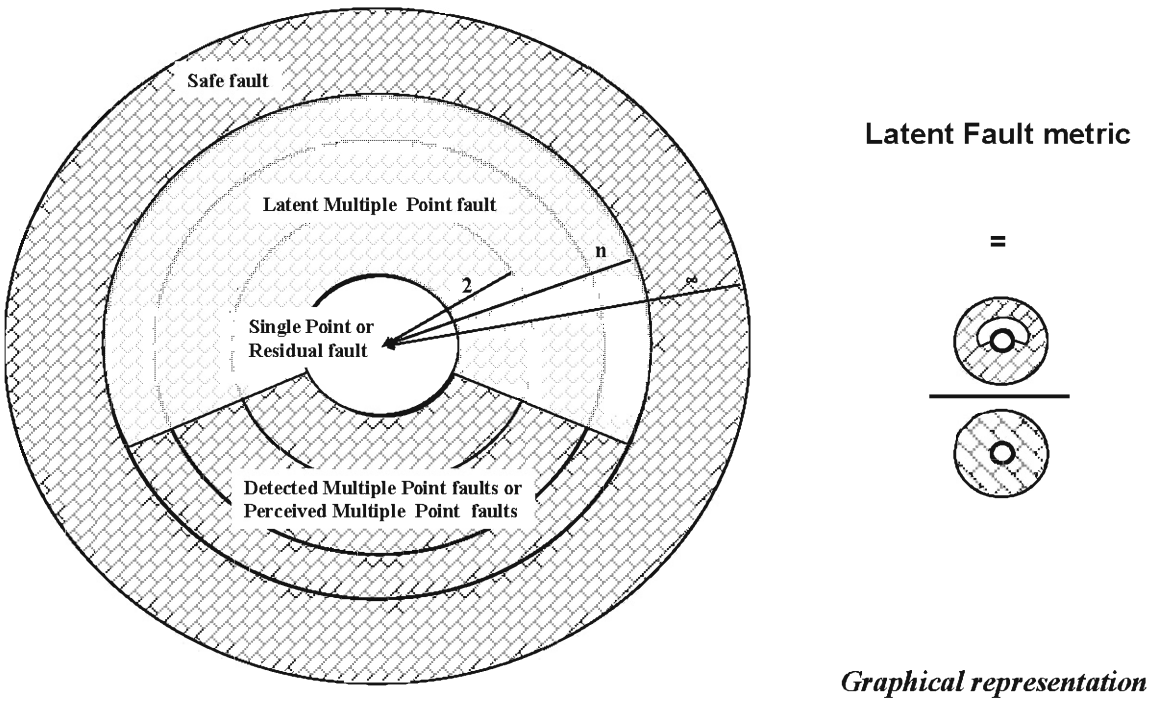


Figure C.3 — Graphical representation of the latent faults metric

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 make available a rationale for:

   1) The compliance with single point fault and latent fault metrics defined in Clause 8 of this part;

   2) The compliance with the evaluation of violation of the safety goal due to random hardware failures as defined in Clause 9.

b) A guideline in order to choose appropriate diagnostic techniques to be implemented in the E/E architecture to detect failures of elements.

NOTE This Annex does not deal with the complete effectiveness of the techniques mentioned in the tables (capacity to avoid the violation of the safety goals). This effectiveness is related both to diagnostic coverage, and to timing aspects associated with the item and its use (e.g. periodicity of diagnostic).

Table D.1 gives the recommendations for faults or failures that shall be detected in order to achieve the relevant level of diagnostic coverage. These recommendations are given considering the generic hardware of a system of the following Figure D.1. Table D.1 only gives information to help defining the faults or failures model, the model can depend on the application in which the component is used.

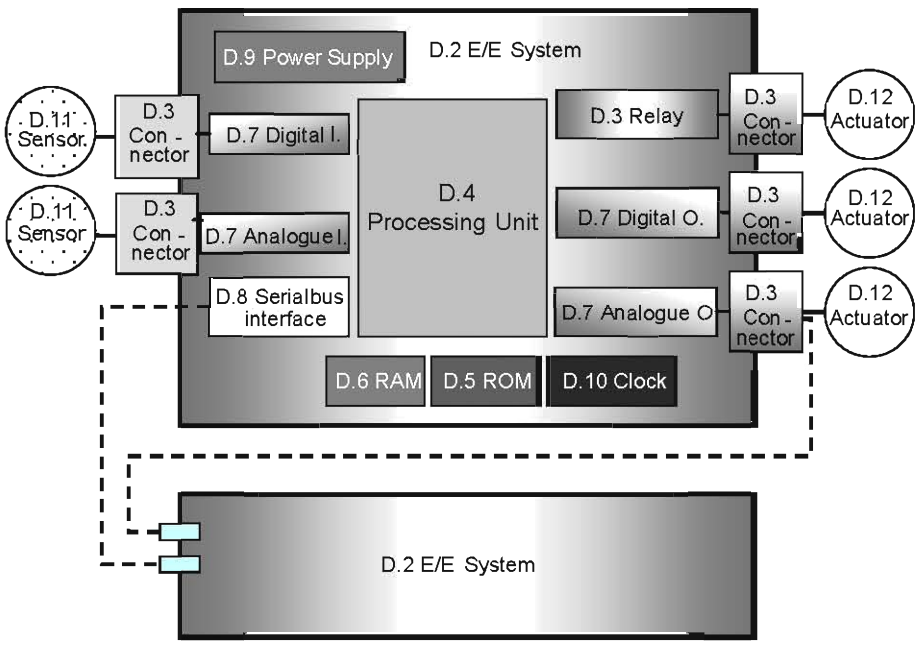


Figure D.1 — Generic hardware of a system

NOTE Every "D.x" label refers to the tables in which the diagnostic coverage recommendations for the specific type of component are given. Tables D.2 to D.12 support the information of Table D.1 by recommending techniques for diagnostic tests and indicate maximum levels of diagnostic coverage that can be claimed using them. The designations low, medium and high diagnostic coverage are quantified as 60 %, 90 % and 99 % respectively. Tables D.1 to D.12 are not exhaustive and other techniques can be used, provided evidence is produced to support the claimed diagnostic coverage. If justified, higher diagnostic coverage can be estimated, up to 100% for simple or complex components.

Table D.1 — Faults or failures to be analysed in the derivation of diagnostic coverage

Component	See Tables	Recommendations for diagnostic coverage		
		Low (60 %)	Medium (90 %)	High (99 %)
EE Systems	**D.2**	No generic fault model available. Detailed analysis necessary	No generic fault model available-Detailed analysis necessary	No generic fault model available Detailed analysis necessary
Electrical components	**D.3**			
Relays		Does not energize or de-energize Welded contacts	Does not energize or de-energize Individual contacts welded	Does not energize or de-energize Individual contacts welded
Connectors		Open Circuit	Open Circuit Short Circuit between pins	Open Circuit Short Circuit between pins Contact Resistance
Harnesses including splice		Open Circuit Short Circuit to Ground	Open Circuit Short Circuit to Ground Short Circuit to Vbat Short Circuit to other wires	Open Circuit Short Circuit to Ground Short Circuit to Vbat Short Circuit to other wires Resistive drift
Processing units	**D.4**			
Register, internal RAM		Stuck-at (see footnote) for data and addresses	d.c. fault model (see foot note)for data and addresses	d.c. fault model for data and addresses Dynamic cross-over for memory cells No, wrong or multiple addressing
Coding and execution including flag register		Wrong coding or no execution	Wrong coding or wrong execution	No generic fault model available. Detailed analysis necessary. Depends on CPU architecture
Address calculation		Stuck-at	d.c. fault model	No generic fault model available. Detailed analysis necessary.
Interrupt handling		No or continuous interrupts	No or continuous interrupts Cross-over of interrupts	No or continuous interrupts Cross-over of interrupts
Invariable memory range	**D.5**	Stuck-at for data and addresses	d.c. fault model for data and addresses	All faults which affect data in the memory
Variable memory range	**D.6**	Stuck-at for data and addresses	d.c. fault model for data and addresses	d.c. fault model for data and addresses Dynamic cross-over for memory cells No, wrong or multiple addressing
Analog I/O and Digital I/O	**D.7**			
Digital I/O		Stuck-at	d.c. fault model	d.c. fault model drift and oscillation
Analogue I/O		Stuck-at	d.c. fault model drift and oscillation	d.c. fault model drift and oscillation
Communication bus (serial, parallel)	**D.8**	-	-	-
General	-	Stuck-at of the addresses	Time out	Time out
Memory management unit	-	Stuck-at of data or addresses	Wrong address decoding	Wrong address decoding
Direct memory access	-	No or continuous access	d.c. fault model for data and addresses Wrong access time	All faults which affect data in the memory Wrong data or addresses Wrong access time
Bus-arbitration	-	Stuck-at of arbitration signals	No or continuous arbitration	No or continuous or wrong arbitration

Component	See Tables	Recommendations for diagnostic coverage		
		Low (60 %)	Medium (90 %)	High (99 %)
Data transmission (to be analysed with ISO 26262-6:—, Annex D)	-	Failure of communication peer Message corruption Message delay Message loss Unintended message repetition	Previous + Resequencing Insertion of message	Previous + Masquerading
Power supply Converter	D.9	-	-	-
Program sequence monitoring (Watchdog) / clock	D.10	-	-	-
Sensors including signal switches	D.11	No generic fault model available. Detailed analysis necessary.	No generic fault model available. Detailed analysis necessary.	No generic fault model available. Detailed analysis necessary.
Final elements (actuators, lamps, buzzer, screen...)	D.12	No generic fault model available. Detailed analysis necessary.	No generic fault model available. Detailed analysis necessary.	No generic fault model available. Detailed analysis necessary.
DEFINITIONS: - "Stuck-at": is a fault category that can be described with continuous "0" or "1" or "on" at the pins of a component. - "d.c. fault model" (d.c. = direct current): includes the following failure modes: stuck-at faults, stuck-open, open or high impedance outputs as well as short circuits between signal lines.				

Table D.2 — Systems

Diagnostic technique/measure	See overview of techniques	Maximum 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 signal monitoring in preference to digital on/off states	D.2.2.2	Low	-
Self-test by software cross exchange between two independent units)	D.2.3.5	Medium	Depends of the quality of the self test

Table D.3 — Electrical components

Diagnostic technique/measure	See overview of techniques	Maximum diagnostic coverage considered achievable	Notes
Failure detection by on-line monitoring	D.2.1.1	High	Depends on diagnostic coverage of failure detection

NOTE Table D.3 deals only with diagnostic techniques dedicated to electrical components. General techniques like “comparator” are also able to detect failures of electrical components but are not integrated in this Table (already included in Table D.2).

Table D.4 — Processing units

Diagnostic technique/measure	See overview of techniques	Maximum 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.2.3.5	Medium	Depends of the quality of the self test
Self-test by software: walking bit (one-channel)	D.2.3.2	Medium	Depends on the quality of the self test
Self-test supported by hardware (one-channel)	D.2.3.3	Medium	Depends on the quality of the self test
Coded processing (one-channel)	D.2.3.4	High	-
Software diversified redundancy (one hardware channel)	D.2.3.6	Medium	Depends of the quality of the diversification
Reciprocal comparison by software	D.2.3.7	High	Depends of the quality of the comparison

Table D.5 — Invariable memory ranges

Diagnostic technique/measure	See overview of techniques	Maximum diagnostic coverage considered achievable	Notes
Parity bit	-	Low	
Detection of memory data failures with error-detection-correction codes (EDC)	D.2.4.1	High	The effectiveness depends on the number of redundant bits.
Modified checksum	D.2.4.2	Low	-
Signature of one byte (8-bit) (CRC)	D.2.4.3	Medium	The effectiveness of the signature depends on the polynomial in relation to the block length of the information to be protected.
Signature of a double byte (16-bit) (CRC)	D.2.4.4	High	The effectiveness of the signature depends on the polynomial in relation to the block length of the information to be protected.
Block replication	D.2.4.5	High	-

Table D.6 — Variable memory ranges

Diagnostic technique/measure	See overview of techniques	Maximum diagnostic coverage considered achievable	Notes
RAM test "checkerboard" or "march"	D.2.5.1	Low	-
One bit redundancy	D.2.5.2	Low	-
Detection of RAM data failures with error-detection-correction codes (EDC)	D.2.4.1	High	-
Block replication	D.2.4.5	High	-

Table D.7 — Analogue and digital I/O

Diagnostic technique/measure	See overview of techniques	Maximum 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.2.6.1	High	Depends on type of pattern
Code protection for digital I/O	D.2.6.2	Medium	Depends on type of coding
Multi-channel parallel output	D.2.6.3	High	-
Monitored outputs	D.2.6.4	High	Only if dataflow changes within diagnostic test interval
Input comparison/voting (1oo2, 2oo3 or better redundancy)	D.2.6.5	High	Only if dataflow changes within diagnostic test interval
[a] Digital I/O can be periodical.			

Table D.8 — Communication bus (serial, parallel)

Diagnostic technique/measure	See overview of techniques	Maximum diagnostic coverage considered achievable	Notes
One-bit hardware redundancy	D.2.7.1	Low	-
Multi-bit hardware redundancy	D.2.7.2	Medium	-
Complete hardware redundancy	D.2.7.3	High	-
Inspection using test patterns	D.2.7.4	High	-
Transmission redundancy	D.2.7.5	Medium	Depends on type of redundancy. Effective only against transient faults
Information redundancy	D.2.7.6	Medium	Depends on type of redundancy.
Frame counter	D.2.7.7	Medium	-
Process counter	D.2.7.8	Medium	-
Combination of information redundancy and “process or frame” counter	-	High	-

Table D.9 — Power supply

Diagnostic technique/measure	See overview of techniques	Maximum diagnostic coverage considered achievable	Notes
Voltage or current control (primary)	D.2.8.1	Low	Recommended always to be used in addition to other techniques in this table
Voltage or current control (secondary)	D.2.8.2	High	-

Table D.10 — Programme sequence monitoring (Watchdog) / Clock

Diagnostic technique/measure	See overview of techniques	Maximum diagnostic coverage considered achievable	Notes
Watch-dog with separate time base without time-window	D.2.9.1	Low	-
Watch-dog with separate time base and time-window	D.2.9.2	Medium	Depends on time restriction for the time-window
Logical monitoring of program sequence	D.2.9.3	Medium	Only effective against clock failures if external temporal events influence the logical program flow
Temporal and logical monitoring of program sequence	D.2.9.4	High	-

Table D.11 — Sensors

Diagnostic technique/measure	See overview of techniques	Maximum 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.2.6.1	High	-
Input comparison/voting (1oo2, 2oo3 or better redundancy)	D.2.6.5	High	Only if dataflow changes within diagnostic test interval
Reference sensor	D.2.10.1	High	Depends on diagnostic coverage of failure detection

Table D.12 — Final elements (actuators)

Diagnostic technique/measure	See overview of techniques	Maximum 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.2.6.1	High	
Monitoring (i.e. coherence control)	D.2.11.1	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 components.

D.2.1.1 Failure detection by on-line monitoring

NOTE This technique/measure is referenced in Tables D.2, D.3, D.7, D.11 and D.12.

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 localise the failure.

NOTE Generally, there is no specific hardware component for the realisation of the on-line monitoring diagram. On-line monitoring detects an abnormal behaviour of the system with regard 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 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 Unlike the comparator, the majority voter technique allows increasing the availability by insuring 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 components.

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 components. This technique is often associated with electromechanical components.

D.2.2.2 Analogue signal monitoring in preference to digital on/off states

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

Aim: To improve confidence in measured signals.

Description: Wherever there is a choice, analogue signals are used in preference to digital on/off states. For example, trip or safe states are represented by analogue signal levels, usually with signal level tolerance monitoring. In case of digital signal, it is possible to monitor it with an analog input. The technique gives continuity monitoring and a higher level of confidence in the transmitter, reducing the required frequency of periodic test performed to detect failures of the transmitter sensing function.

D.2.3 Processing units

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

D.2.3.1 Self-test by software: limited number of patterns (one-channel)

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

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

Description: The hardware is built using standard techniques which do not consider any special safety requirements. The failure detection is realised entirely by additional software functions which perform self-tests using at least two complementary data patterns (for example 55hex and AAhex).

D.2.3.2 Self-test by software: walking bit (one-channel)

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

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

Description: The failure detection is realised entirely by additional software functions which perform self-tests using a data pattern (for example walking-bit pattern) which tests the physical storage (data and address registers) and the instruction decoder. However, the diagnostic coverage is only 90 %.

D.2.3.3 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, 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 failure. For example, this could be a hardware unit which cyclically monitors the output of a certain bit pattern in accordance with the watch-dog principle.

D.2.3.4 Coded processing (one-channel)

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

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

Description: Processing units can be designed with special failure-recognising or failure-correcting circuit techniques. So far, these techniques have been applied only to relatively simple circuits and are not widespread; however, future developments are not excluded.

D.2.3.5 Self-test by software (BIST) cross exchanged between two independent units

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

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

Description: The failure detection is realised entirely into two or more units by additional software functions which perform self-tests using a BIST model (for example walking-bit pattern) which tests the physical storage (data and address registers) and the instruction decoder. The processing units exchange the results.

D.2.3.6 Software diversified redundancy (one hardware channel)

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 redundant software implementations using different hardware resources (e.g. different RAM, ROM memory ranges). A comparison of the output data of the two redundant software implementations is carried out. Detected differences lead to a failure message. (see D.1.3)

D.2.3.7 Reciprocal comparison by software

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.

D.2.4 Invariable memory ranges

Global objective: The detection of information corruptions in the invariable memory.

D.2.4.1 Memory monitoring with a modified Hamming code, or detection of data failures with error-detection-correction codes (EDC)

NOTE This technique/measure is referenced in Tables D.5 and D.6.

Aim: To detect each single-bit failure, each two-bit failure, some three-bit failures, and some all-bit failures in a 16-bit word.

Description: Every word of memory is extended by several redundant bits to produce a modified Hamming code with a Hamming distance of at least 4. Every time a word is read, checking of the redundant bits can determine whether or not a corruption has taken place. If a difference is found, a failure message is produced. The procedure can also be used to detect addressing failures, by calculating the redundant bits for the concatenation of the data word and its address.

D.2.4.2 Modified checksum

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

Aim: To detect each single bit failure.

Description: A checksum is created by a suitable algorithm which uses each of the words in a block of memory. The checksum can be stored as an additional word in ROM, or an additional word can be added to the memory block to ensure that the checksum algorithm produces a predetermined value. In a later memory test, a checksum is created again using the same algorithm, and the result is compared with the stored or defined value. If a difference is found, a failure message is produced.

D.2.4.3 Signature of one byte (8-bit)

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

Aim: To detect each one-bit failure and each multi-bit failure within a byte.

Description: The contents of a memory block is compressed (using either hardware or software) using a cyclic redundancy check (CRC) algorithm into one memory byte. A typical CRC algorithm treats the whole contents

of the block as byte-serial or bit-serial data flow, on which a continuous polynomial division is carried out using a polynomial generator. The remainder of the division represents the compressed memory contents – it is the "signature" of the memory – and is stored. The signature is computed once again in later tests and compared with one already stored. A failure message is produced if there is a difference.

D.2.4.4 Signature of a double byte (16-bit)

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

Aim: To detect each one-bit failure and each multi-bit failure within a byte.

Description: This procedure calculates a signature using a cyclic redundancy check (CRC) algorithm, but the resulting value is at least two bytes in size. The extended signature is stored, recalculated and compared as in the single-byte case. A failure message is produced if there is a difference between the stored and recalculated signatures.

D.2.4.5 Block replication (for example double memory with hardware or software comparison)

NOTE This technique/measure is referenced in Table D.5 and D.6.

Aim: To detect each bit failure.

Description: The address space is duplicated in two memories. The first memory is operated in the normal manner. The second memory contains the same information and is accessed in parallel to the first. The outputs are compared and a failure message is produced if a difference is detected. In order to detect certain kinds of bit errors, the data is to be stored inversely in one of the two memories and inverted once again when read.

D.2.5 Variable memory ranges

Global objective: Detecting failures during addressing, writing, storing and reading.

D.2.5.1 RAM test "checkerboard" or "march"

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

Aim: To detect predominantly static bit failures.

Description: A checker-board type pattern of 0s and 1s is written into the cells of memory. The cells are then inspected in pairs to ensure that the contents are the same and correct. The address of the first cell of such a pair is variable and the address of the second cell of the pair is formed by inverting bitwise the first address. In the first run, the address range of the memory is run towards higher addresses from the variable address, and in a second run towards lower addresses. Both runs are then repeated with an inverted pre-assignment. A failure message is produced if any difference occurs.

In a RAM test "march" the cells of a bit-oriented memory are initialised by a uniform bit stream (i.e. all 0s or all 1s). In the first run, the cells are inspected in ascending order: each cell is checked for the correct contents and its contents are inverted. The background, which is created in the first run, is treated in a second run in descending order and in the same manner. Both first and second runs are repeated with an inverted pre-assignment in a third or fourth run. A failure message is produced if a difference occurs.

D.2.5.2 One-bit redundancy (for example RAM monitoring with a parity bit)

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

Aim: To detect 50 % of all possible bit failures in the memory range tested.

Description: Every word of the memory is extended by one bit (the parity bit) which completes each word to an even or odd number of logical 1s. The parity of the data word is checked each time it is read. If the wrong number of 1s is found, a failure message is produced. The choice of even or odd parity ought to be made such that, whichever of the zero word (nothing but 0s) and the one word (nothing but 1s) is the more unfavourable in the event of a failure, then that word is not a valid code. Parity can also be used to detect addressing failures, when the parity is calculated for the concatenation of the data word and its address.

D.2.6 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.2.6.1 Test pattern

NOTE This technique/measure is referenced in Tables D.7, D.11 and D.12.

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. The test pattern information, the test pattern reception, and test pattern evaluation are to be independent of each other. The functional behaviour of the system is not to be unacceptably influenced by the test pattern.

D.2.6.2 Code protection

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

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. 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.2.6.3 Multi-channel parallel output

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

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.2.6.4 Monitored outputs

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

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.2.6.5 Input comparison/voting

NOTE This technique/measure is referenced in Tables D.7, D.11.

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.2.7 Communication bus

Global objective: To detect failures caused by a defect in the information transfer.

D.2.7.1 One-bit hardware redundancy

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

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

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

D.2.7.2 Multi-bit hardware redundancy

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

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

Description: The bus is extended by two or more lines (bits) and these additional lines (bits) are used in order to detect failures by Hamming code techniques.

D.2.7.3 Complete hardware redundancy

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

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

Description: The bus is duplicated and the additional lines (bits) are used in order to detect failures.

D.2.7.4 Inspection using test patterns

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

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.

The test pattern information, the test pattern reception, and test pattern evaluation are to be independent of each other. The functional behaviour of the system is not to be unacceptably influenced by the test pattern.

D.2.7.5 Transmission redundancy

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

Aim: To detect transient failures in bus communication.

Description: The information is transferred several times in sequence. The repetition is effective only against transient failures.

D.2.7.6 Information redundancy

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

Aim: To detect failures in bus communication.

Description: Data is transmitted in blocks, together with a calculated checksum for each block. The receiver then re-calculates the checksum (and CRC eventually) of the received data and compares the result with the received checksum.

D.2.7.7 Frame counter

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

Aim: To detect frame losses.

Description: Frames include a frame number, which is transmitted on the bus. The receiver is then able to detect any frame loss.

D.2.7.8 Process counter

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

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

Description: Data (payload) is transmitted in a block which is labelled by an identifier. The identifier is sent on the bus together with the data block. The receiver is then able to detect any loss of data.

D.2.8 Power supply

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

D.2.8.1 Voltage or current control (primary)

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

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

Description: Monitoring of input voltage or current.

D.2.8.2 Voltage control (secondary)

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

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

Description: Monitoring of output voltage or current.

D.2.9 Temporal and logical program sequence monitoring

NOTE This group of techniques and measures is referenced in Table D.10.

Global objective: To detect a defective program sequence. A defective program sequence exists if the individual elements of a program (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.2.9.1 Watch-dog with separate time base without time-window

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

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

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

D.2.9.2 Watch-dog with separate time base and time-window

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

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

Description: External timing elements with a separate time base (for example watch-dog timers) are periodically triggered to monitor the computer's behaviour and the plausibility of the program sequence. It is important that the triggering points are correctly placed in the program. A lower and upper limit is given for the watch-dog timer. If the program sequence takes a longer or shorter time than expected, emergency action is taken.

D.2.9.3 Logical monitoring of program sequence

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

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

Description: The correct sequence of the individual program sections is monitored using software (counting procedure, key procedure) or using external monitoring facilities. It is important that the checking points are placed in the program correctly.

D.2.9.4 Combination of temporal and logical monitoring of program sequences

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

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

Description: A temporal facility (for example a watch-dog timer) monitoring the program sequence is retriggered only if the sequence of the program sections is also executed correctly.

D.2.10 Sensors

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

D.2.10.1 Reference sensor

NOTE This technique/measure is referenced in Table D.11.

Aim: To detect the incorrect operation of a sensor.

Description: An independent reference sensor is used to monitor the operation of a process sensor. Each input signal is checked at suitable time intervals by the reference sensor to detect failures of the process sensor.

D.2.11 Final elements (actuators)

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

D.2.11.1 Monitoring

NOTE This technique/measure is referenced in Table D.12.

Aim: To detect the incorrect operation of an actuator.

Description: The operation of the actuator is monitored.

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

Example 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.

Annex E
(informative)

Target values for hardware architectural metrics

Table E.1 — Single point faults metric and latent faults metric target values

	ASIL B	ASIL C	ASIL D
Single point faults metric	> 90 %	> 97 %	> 99 %
Latent faults metric	> 60 %	> 80 %	> 90 %

Annex F
(informative)

Example of calculation of the hardware architectural metrics: "single point faults metric" and "latent faults metric"

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

Function 1 has two inputs (wheel speed measured via sensors I1 and I2 generating pulses) and one output (valve1 controlled by I61) and consists in opening valve1 when vehicle speed is higher than 100 km/h.

The associated safety goal 1 is "valve1 shall not close when speed is higher than 100 km/h". It is ASIL C. Safe state is: valve1 open.

Sensors I1 and I2 pulses are acquired by the microcontroller and compared. In case of inconsistency, Out.1 is set to 0 (Safety Mechanism SM1 in the tables). This opens valve1 (0 voltage on transistor gate open it. 0 voltage on I61 opens valve1. Output stage controlling T61 is monitored by analogue input InADC2 (Safety mechanism SM2 in the tables). Wheel speed is computed using the mean value given by the sensors.

Function 2 has 1 input (temperature measured via sensor R3) and one output (valve2 controlled by I71) and consists in opening valve2 when temperature is higher than 100°C.

The associated safety goal 2 is "valve2 shall not close when temperature is higher than 100 °C". It is ASIL B. Safe state is: valve2 open.

Sensor R3 is acquired by the microcontroller ADC. R3 resistance decreases as temperature rises. There is no monitoring on this input. Output stage controlling T71 is monitored by analogue input InADC1 (Safety mechanism SM4 in the tables).

The microcontroller has no internal redundancy. In order to simplify the analysis, it is assumed that the microcontroller shows 50% safe and 50% multiple point failures. A global coverage of 90%, through internal self tests and the external watchdog (Safety Mechanism SM3 in the tables) is also assumed. Watchdog gets alive signal via output 0 of the microcontroller. When watchdog is no more refreshed, its output goes low. This switches both functions to safe state.

L1 is a LED on the dashboard, signalling to the driver a proportion of the multiple point failures as well as safe state activation.

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 The fault model of a resistor depends if the hardware part is used in a digital (such as R11, R12, R13...) or an analog 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 can use the failure mode coverage of only the safety mechanisms that aim at preventing the violation of the safety goal. The second metric can use the failure mode coverage of only the safety mechanisms that aim at preventing the failure mode from being latent.

Example The failure mode open of R21 has the potential to violate the safety goal 1 in absence of safety mechanism. Safety mechanism 1 detects this failure mode with a failure mode coverage of 99% and switch the system into a safe state. When detecting this failure mode, an alert is displayed; the failure mode coverage with regard to latent failures is 100%.

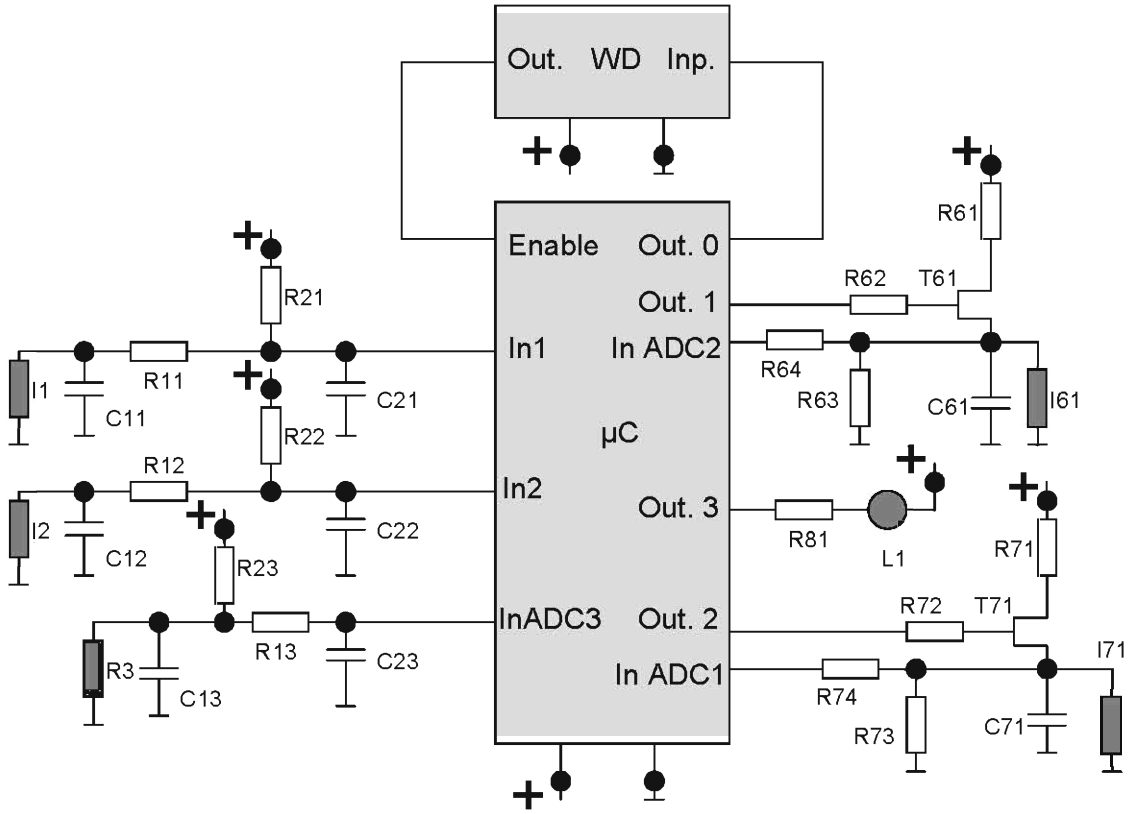


Figure F.1 — Example diagram

Table F.1 — Safety goal 1

Component Name	Failure rate / FIT	Safety Related Component ? No Safety Related Component ?	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	SR	open	90%	X	SM1	99%	0,018	X	SM1	100%	0
note1			closed	10%					X		0%	0,2
R12	2	SR	open	90%	X	SM1	99%	0,018	X	SM1	100%	0
note1			closed	10%					X		0%	0,2
R21	2	SR	open	90%	X	SM1	99%	0,018	X	SM1	100%	0
note2			closed	10%	X		99%	0,002	X		100%	0
R22	2	SR	open	90%	X	SM1	99%	0,018	X	SM1	100%	0
note2			closed	10%	X		99%	0,002	X		100%	0
C11	2	SR	open	20%		SM1			X	SM1	0%	0,4
note1			closed	80%	X		99%	0,016	X		100%	0
C12	2	SR	open	20%		SM1			X	SM1	0%	0,4
note1			closed	80%	X		99%	0,016	X		100%	0
C21	2	SR	open	20%		SM1				SM1		
			closed	80%	X		99%	0,016	X		100%	0
C22	2	SR	open	20%		SM1				SM1		
			closed	80%	X		99%	0,016	X		100%	0
I1	4	SR	open	70%	X	SM1	99%	0,028	X	SM1	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	SR	open	70%	X	SM1	99%	0,028	X	SM1	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	SR	Out. Stuck at 1	50%					X	none	0%	10
			Out. Stuck at 0	50%								
T61	5	SR	open gate	10%		SM2				SM2		
note3			closed gate	10%	X		90%	0,05	X		100%	0
			open drain source	20%								
			closed drain source	20%	X		90%	0,1	X		100%	0
			drift 0,5	20%	X		90%	0,1	X		100%	0
			drift 2	20%								
R61	2	SR	open	90%						none		
note4			closed	10%					X		0%	0,2
R62	2	SR	open	90%						none		
note4			closed	10%					X		0%	0,2
R63	2	SR	open	90%						none		
			closed	10%					X		0%	0,2
R64	2	SR	open	90%					X	none	0%	1,8
note1			closed	10%					X		0%	0,2
I61	5	SR	open	70%						SM2		
			closed	20%								
			drift 0,5	5%					X		90%	0,025
			drift 2	5%								
C61	2	SR	open	20%					X	none	0%	0,4
note5			closed	80%								
R81	2	NSR	open	90%								
			closed	10%								
L1	10	NSR	open	90%								
			closed	10%								
µC	100	SR	All	50%	X	SM3	90%	5	X	SM3	100%	0
			All	50%								
								Σ 5,47			Σ	14,23

Total failure rate	176	Single Point Faults Metric	96,7%
Total Safety Related	164		
Total Not Safety Related	12	Latent Faults Metric	91,0%

NOTE 1 The purpose of this 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. Both failure modes are multiple point faults.

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 wrong speed acquisition. The sensor is an open-collector sensor.

NOTE 3 A drift of 0.5 can cause unwanted command.

NOTE 4 The purpose of this part is electrical protection. The close failure mode means loss of protection, which is a multiple point fault.

NOTE 5 The purpose of this part is ESD protection. The open failure mode means loss of protection, which is a multiple point fault.

Table F.2 — Safety goal 2

Component Name	Failure rate / FIT	Safety Related Component ? No Safety Related Component ?	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
R3	3	SR	open	30%	X	none	0%	0,9				
			closed	10%								
			drift 0,5	30%								
			drift 2	30%	X		0%	0,9				
R13 note1	2	SR	open	90%	X	none	0%	1,8		none		
			closed	10%					X		0%	0,2
R23	2	SR	open	90%		none						
			closed	10%	X		0%	0,2				
C13 note2	2	SR	open	20%					X		0%	0,4
			closed	80%								
C23	2	NSR	open	20%								
			closed	80%								
WD	20	SR	Out. Stuck at 1	50%					X	none	0%	10
			Out. Stuck at 0	50%								
T71 note 3	5	SR	open gate	10%		SM4				SM4		
			closed gate	10%	X		90%	0,05	X		100%	0
			open drain source	20%								
			closed drain source	20%	X		90%	0,1	X		100%	0
			drift 0,5	20%	X		90%	0,1	X		100%	0
			drift 2	20%								
R71 note1	2	SR	open	90%						none		
			closed	10%					X		0%	0,2
R72 note1	2	SR	open	90%						none		
			closed	10%					X		0%	0,2
R73		SR	open	90%						none		
			closed	10%					X		0%	0,2
R74 note4	2	SR	open	90%					X	none	0%	1,8
			closed	10%					X		0%	0,2
I71	5	SR	open	70%						SM4		
			closed	20%								
			drift 0,5	5%					X		90%	0,025
			drift 2	5%								
C71 note2	2	SR	open	20%					X	none		0,4
			closed	80%								
R81	2	NSR	open	90%								
			closed	10%								
L1	10	NSR	open	90%								
			closed	10%								
µC	100	SR	All	50%	X	SM3	90%	5	X	SM3	100%	0
			All	50%								
							Σ	9,05			Σ	13,625
		Total failure rate	161		Single Point Faults Metric	93,8%			Latent Faults Metric	90,1%		
		Total Safety Related	147									
		Total Not Safety Related	14									

NOTE 1 The purpose of this part is electrical protection. The closed failure mode means loss of protection which is a multiple point fault.

NOTE 2 The purpose of this part is ESD protection. The open failure mode means loss of protection which is a multiple point fault.

NOTE 3 A drift of 0.5 can cause unwanted command.

NOTE 4 The purpose of this 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. Both failure modes are multiple point faults.

Annex G
(informative)

Target values for violation of a safety goal due to random hardware failures

Table G.1 — Random hardware failure target values

ASIL Level	Random hardware failure target values
D	$< 10^{-8}\ h^{-1}$
C	$< 10^{-7}\ h^{-1}$
B	$< 10^{-7}\ h^{-1}$
A	$< 10^{-6}\ h^{-1}$

Bibliography

[1] IEC 61508 (all parts), Functional safety of electrical/electronic/programmable electronic safety-related systems

[2] IEC 50-191, International Electrotechnical Vocabulary, Chapter 191: Dependability and quality of service

[3] ISO/IEC 2382-1, Information technology -- Vocabulary -- Part 1: Fundamental terms

[4] ISO/IEC 2382-20, Information technology -- Vocabulary -- Part 20: System development

[5] ISO 8402, Quality management and quality assurance -Vocabulary

[6] ISO 11452-2, Road vehicles — Component test methods for electrical disturbances from narrowband radiated electromagnetic energy — Part 2: Absorber-lined shielded enclosure

[7] ISO 11452-4, Road vehicles — Component test methods for electrical disturbances from narrowband radiated electromagnetic energy — Part 4: Bulk current injection (BCI)

[8] ISO 7637-2, Road vehicles — Electrical disturbances from conduction and coupling — Part 2: Electrical transient conduction along supply lines only

[9] ISO 10605, Road vehicles -- Test methods for electrical disturbances from electrostatic discharge

[10] ISO/IEC 15026, Information Technology - System and Software Integrity Levels

[11] ISO 16750-2, Road vehicles -- Environmental conditions and testing for electrical and electronic equipment -- Part 2: Electrical loads

[12] ISO 16750-4, Road vehicles -- Environmental conditions and testing for electrical and electronic equipment -- Part 4: Climatic loads

[13] ISO 16750-5, Road vehicles -- Environmental conditions and testing for electrical and electronic equipment -- Part 5: Chemical loads

[14] IEC 60300-3-9, Dependability management-Part 3: Application guide-Section 9: Risk analysis of technological systems

[15] IEC 60050(191), International Electrotechnical Vocabulary – Dependability and quality of service

[16] IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 1, NO. 1, JANUARY-MARCH 2004 - Basic Concepts and Taxonomy of Dependable and Secure Computing.

[17] ISO 17025, General Requirements for the Competence of Testing and Calibration Laboratories.