

Safety of machinery —
Functional safety of
safety-related
electrical, electronic
and programmable
electronic control
systems

ICS 13.110; 25.040.99; 29.020

National foreword

This British Standard is the UK implementation of EN 62061:2005+A2:2015, incorporating corrigendum February 2010. It is identical to IEC 62061:2005, incorporating amendments 1:2012 and 2:2015, and corrigenda July 2005 and April 2008. It supersedes BS EN 62061:2005+A1:2013, which is withdrawn.

The start and finish of text introduced or altered by amendments is indicated in the text by tags. Tags indicating changes to IEC text carry the number of the IEC amendment. For example, text altered by IEC amendment 1 is indicated by A1 tags \square_{A1} \triangleleft_{A1} .

The start and finish of text introduced or altered by corrigendum is indicated in the text by tags. Text altered by IEC corrigendum July 2005 is indicated in the text by \square_{AC1} \triangleleft_{AC1} , and text altered by IEC corrigendum April 2008 is indicated in the text by \square_{AC2} \triangleleft_{AC2} .

The UK participation in its preparation was entrusted to Technical Committee MCE/3, Safeguarding of machinery.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

Compliance with a British Standard cannot confer immunity from legal obligations.

Amendments/corrigenda issued since publication

Amd. No.	Date	Comments
15929 Corrigendum No. 1	July 2006	Implementation of IEC corrigendum July 2005
	28 February 2009	Implementation of IEC corrigendum April 2008
	31 May 2010	Implementation of CENELEC corrigendum February 2010. Replacement of EC Directive 98/37/EC with 2006/42/EC and deletion of the second dashed item in Annex ZZ
	30 June 2013	Implementation of IEC amendment 1:2012 with CENELEC endorsement A1:2013: Annex ZA and ZZ updated
	31 October 2015	Implementation of IEC amendment 2:2015 with CENELEC endorsement A2:2015: Annex ZA updated

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 26 April 2005

© The British Standards Institution 2015.
Published by BSI Standards Limited 2015

EUROPEAN STANDARD

EN 62061:2005+A2

NORME EUROPÉENNE

EUROPÄISCHE NORM

August 2015

ICS 13.110; 25.040.99; 29.020

Incorporates corrigendum February 2010

English version

**Safety of machinery –
Functional safety of safety-related electrical,
electronic and programmable electronic control systems**
(IEC 62061:2005)

Sécurité des machines –
Sécurité fonctionnelle des systèmes
de commande électriques, électroniques
et électroniques programmables relatifs
à la sécurité
(CEI 62061:2005)

Sicherheit von Maschinen –
Funktionale Sicherheit
sicherheitsbezogener elektrischer,
elektronischer und programmierbarer
elektronischer Steuerungssysteme
(IEC 62061:2005)

This European Standard was approved by CENELEC on 2004-12-01. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the Central Secretariat or to any CENELEC member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CENELEC member into its own language and notified to the Central Secretariat has the same status as the official versions.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Slovakia, Slovenia, Spain, Sweden, Switzerland and United Kingdom.

CENELEC

European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

Central Secretariat: rue de Stassart 35, B - 1050 Brussels

Foreword

The text of document 44/460/FDIS, future edition 1 of IEC 62061, prepared by IEC TC 44, Safety of machinery - Electrotechnical aspects, was submitted to the IEC-CENELEC parallel vote and was approved by CENELEC as EN 62061 on 2004-12-01.

The following dates were fixed:

- latest date by which the EN has to be implemented at national level by publication of an identical national standard or by endorsement (dop) 2005-11-01
- latest date by which the national standards conflicting with the EN have to be withdrawn (dow) 2007-12-01

This European Standard has been prepared under a mandate given to CENELEC by the European Commission and the European Free Trade Association and covers essential requirements of EC Directive 2006/42/EC. See Annex ZZ.

PROOF TEST INTERVAL AND LIFETIME

The following important information should be noted in relation to the requirements of this standard:

Where the probability of dangerous failure per hour (PFH_D) is highly dependent upon proof testing (i.e. tests intended to reveal faults not detected by diagnostic functions) then the proof test interval needs to be shown as realistic and practicable in the context of the expected use of the safety-related electrical control system (SRECS) (e.g. proof test intervals of less than 10 years can be unreasonably short for many machinery applications).

CEN/TC114/WG6 have used a proof test interval (mission time) of 20 years to support the estimation of mean time to dangerous failure ($MTTF_D$) for the realization of designated architectures in Annex B of prEN ISO 13849-1. Therefore, it is recommended that SRECS designers endeavour to use a 20 year proof test interval.

It is acknowledged that some subsystems and/or subsystem elements (e.g. electro-mechanical components with high duty cycles) will require replacement within the SRECS proof test interval.

Proof testing involves detailed and comprehensive checks that can, in practice, only be performed when the SRECS and/or its subsystems has been designed to facilitate proof testing (e.g. dedicated test ports) and provided with necessary information (e.g. proof test instructions).

To ensure the validity of the proof test interval specified by the designer it is important that any other necessary designated tests (e.g. functional tests) are also successfully performed at the SRECS.

Annexes ZA and ZZ have been added by CENELEC.

Endorsement notice

The text of the International Standard IEC 62061:2005 was approved by CENELEC as a European Standard without any modification.

The contents of the corrigendum of February 2010 have been included in this copy.

Foreword to amendment A1

The text of document 44/655/CDV, future edition 1 of IEC 62061:2005/A1, prepared by IEC TC 44 "Safety of machinery - Electrotechnical aspects" was submitted to the IEC-CENELEC parallel vote and approved by CENELEC as EN 62061:2005/A1:2013.

The following dates are fixed:

- latest date by which the document has (dop) 2013-09-30 to be implemented at national level by publication of an identical national standard or by endorsement
- latest date by which the national (dow) 2015-12-18 standards conflicting with the document have to be withdrawn

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CENELEC [and/or CEN] shall not be held responsible for identifying any or all such patent rights.

Endorsement notice

The text of the International Standard IEC 62061:2005/A1:2012 was approved by CENELEC as a European Standard without any modification.

Foreword to amendment A2

The text of document 44/718/CDV, future edition 1 of IEC 62061:2005/A2, prepared by IEC TC 44 "Safety of machinery - Electrotechnical aspects" was submitted to the IEC-CENELEC parallel vote and approved by CENELEC as EN 62061:2005/A2:2015.

The following dates are fixed:

- latest date by which the document has (dop) 2016-05-01 to be implemented at national level by publication of an identical national standard or by endorsement
- latest date by which the national (dow) 2018-07-31 standards conflicting with the document have to be withdrawn

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CENELEC [and/or CEN] shall not be held responsible for identifying any or all such patent rights.

For the relationship with EU Directive(s) see informative Annex ZZ, which is an integral part of EN 62061:2005.

Endorsement notice

The text of the International Standard IEC 62061:2005/A2:2015 was approved by CENELEC as a European Standard without any modification.

Annex ZA
(normative)

**Normative references to international publications
with their corresponding European publications**

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

NOTE 1 When an International Publication has been modified by common modifications, indicated by (mod), the relevant EN/HD applies.

NOTE 2 Up-to-date information on the latest versions of the European Standards listed in this annex is available here: www.cenelec.eu.

<u>Publication</u>	<u>Year</u>	<u>Title</u>	<u>EN/HD</u>	<u>Year</u>
IEC 60204-1	- ¹⁾	Safety of machinery - Electrical equipment of machines Part 1: General requirements	EN 60204-1 + corr. September	1997 ²⁾ 1998
IEC 61000-6-2, mod.	- ¹⁾	Electromagnetic compatibility (EMC) Part 6-2: Generic standards - Immunity for industrial environments	EN 61000-6-2	2001 ²⁾
IEC 61310	Series	Safety of machinery - Indication, marking and actuation	EN 61310	Series
IEC 61508-2	- ¹⁾	Functional safety of electrical/electronic/programmable electronic safety-related systems Part 2: Requirements for electrical/electronic/programmable electronic safety-related systems	EN 61508-2	2001 ²⁾
IEC 61508-3	- ¹⁾	Part 3: Software requirements	EN 61508-3	2001 ²⁾
ISO 12100	2010	Safety of machinery – General principles for design – Risk assessment and risk reduction	EN ISO 12100	2010
ISO 13849-1	2006	Safety of machinery – Safety-related parts of control systems – Part 1: General principles for design	EN ISO 13849-1	2008
ISO 13849-2	-	Safety of machinery – Safety-related parts of control systems – Part 2: Validation	EN ISO 13849-2	-

1) Undated reference.

2) Valid edition at date of issue.

Annex ZZ
(informative)

Coverage of Essential Requirements of EC Directives

This European Standard has been prepared under a mandate given to CENELEC by the European Commission and the European Free Trade Association and within its scope the Standard covers the following essential requirements out of those given in Annex I of the EC Directive 2006/42/EC

– 1.2.1

Compliance with this standard provides one means of conformity with the specified essential requirements of the Directive concerned.

WARNING: Other requirements, and other EC Directives may be applicable to the products falling within the scope of this standard.

CONTENTS

INTRODUCTION.....	9
1 Scope and object.....	11
2 Normative references	12
3 Terms, definitions and abbreviations	13
3.1 Alphabetical list of definitions	13
3.2 Terms and definitions	15
3.3 Abbreviations	23
4 Management of functional safety.....	24
4.1 Objective.....	24
4.2 Requirements.....	24
5 Requirements for the specification of Safety-Related Control Functions (SRCFs).....	25
5.1 Objective.....	25
5.2 Specification of requirements for SRCFs	25
6 Design and integration of the safety-related electrical control system (SRECS).....	28
6.1 Objective.....	28
6.2 General requirements.....	28
6.3 Requirements for behaviour (of the SRECS) on detection of a fault in the SRECS.....	29
6.4 Requirements for systematic safety integrity of the SRECS	30
6.5 Selection of safety-related electrical control system	32
6.6 Safety-related electrical control system (SRECS) design and development	32
6.7 Realisation of subsystems	37
6.8 Realisation of diagnostic functions	53
6.9 Hardware implementation of the SRECS	54
6.10 Software safety requirements specification.....	54
6.11 Software design and development.....	55
6.12 Safety-related electrical control system integration and testing.....	63
6.13 SRECS installation	64
7 Information for use of the SRECS.....	64
7.1 Objective.....	64
7.2 Documentation for installation, use and maintenance	64
8 Validation of the safety-related electrical control system.....	65
8.1 General requirements.....	65
8.2 Validation of SRECS systematic safety integrity	66
9 Modification.....	67
9.1 Objective.....	67
9.2 Modification procedure	67
9.3 Configuration management procedures	68
10 Documentation	70

Annex A (informative) SIL assignment	72
Annex B (informative) Example of safety-related electrical control system (SRECS) design using concepts and requirements of Clauses 5 and 6	80
Annex C (informative) Guide to embedded software design and development.....	87
Annex F (informative) Methodology for the estimation of susceptibility to common cause failures (CCF).....	97
Figure 1 – Relationship of IEC 62061 to other relevant standards	10
Figure 2 – Workflow of the SRECS design and development process	34
Figure 3 – Allocation of safety requirements to the function blocks to subsystems (see 6.6.2.1.1)	35
Figure 4 – Workflow for subsystem design and development (see box 6B of Figure 2).....	40
Figure 5 – Decomposition of function blocks to function block elements and their associated subsystem elements.....	41
Figure 6 – Subsystem A logical representation	47
Figure 7 – Subsystem B logical representation	48
Figure 8 – Subsystem C logical representation	48
Figure 9 – Subsystem D logical representation	50
Figure A.1 – Workflow of SIL assignment process.....	73
Figure A.2 – Parameters used in risk estimation	74
Figure A.3 – Example proforma for SIL assignment process	79
Figure B.1 – Terminology used in functional decomposition	80
Figure B.2 – Example machine	81
Figure B.3 – Specification of requirements for an SRCF	81
Figure B.4 – Decomposition to a structure of function blocks	82
Figure B.5 – Initial concept of an architecture for a SRECS	83
Figure B.6 – SRECS architecture with diagnostic functions embedded within each subsystem (SS1 to SS4)	84
Figure B.7 – SRECS architecture with diagnostic functions embedded within subsystem SS3.....	85
Figure B.8 – Estimation of PFH_D for a SRECS.....	86

Table 2 – Overview and objectives of IEC 62061	12
Table 3 – Safety integrity levels: target failure values for SRCFs	27
Table 4 – Characteristics of subsystems 1 and 2 used in this example.....	37
Table 5 – Architectural constraints on subsystems: maximum SIL that can be claimed for a SRCF using this subsystem	43
Table 8 – Information and documentation of a SRECS.....	70
Table A.1 – Severity (Se) classification.....	75
Table A.2– Frequency and duration of exposure (Fr) classification	75
Table A.3– Probability (Pr) classification.....	76
Table A.4– Probability of avoiding or limiting harm (Av) classification	77
Table A.5– Parameters used to determine class of probability of harm (Cl).....	77
Table A.6 – SIL assignment matrix.....	78
Table F.1 – Criteria for estimation of CCF.....	96
Table F.2 – Estimation of CCF factor (β).....	97

INTRODUCTION

As a result of automation, demand for increased production and reduced operator physical effort, Safety-Related Electrical Control Systems (referred to as SRECS) of machines play an increasing role in the achievement of overall machine safety. Furthermore, the SRECS themselves increasingly employ complex electronic technology.

Previously, in the absence of standards, there has been a reluctance to accept SRECS in safety-related functions for significant machine hazards because of uncertainty regarding the performance of such technology.

This International Standard is intended for use by machinery designers, control system manufacturers and integrators, and others involved in the specification, design and validation of a SRECS. It sets out an approach and provides requirements to achieve the necessary performance.

This standard is machine sector specific within the framework of IEC 61508. It is intended to facilitate the specification of the performance of safety-related electrical control systems in relation to the significant hazards (see 3.8 of ^{A2}ISO 12100:2010 ^{A2}) of machines.

This standard provides a machine sector specific framework for functional safety of a SRECS of machines. It only covers those aspects of the safety lifecycle that are related to safety requirements allocation through to safety validation. Requirements are provided for information for safe use of SRECS of machines that can also be relevant to later phases of the life of a SRECS.

There are many situations on machines where SRECS are employed as part of safety measures that have been provided to achieve risk reduction. A typical case is the use of an interlocking guard that, when it is opened to allow access to the danger zone, signals the electrical control system to stop hazardous machine operation. Also in automation, the electrical control system that is used to achieve correct operation of the machine process often contributes to safety by mitigating risks associated with hazards arising directly from control system failures. This standard gives a methodology and requirements to

- assign the required safety integrity level for each safety-related control function to be implemented by SRECS;
- enable the design of the SRECS appropriate to the assigned safety-related control function(s);
- integrate safety-related subsystems designed in accordance with ISO 13849 ;
- validate the SRECS.

This standard is intended to be used within the framework of systematic risk reduction described in ^{A2}ISO 12100 ^{A2} and in conjunction with risk assessment according to the principles described in ^{A2}ISO 12100 ^{A2}. A suggested methodology for safety integrity level (SIL) assignment is given in informative Annex A.

Measures are given to co-ordinate the performance of the SRECS with the intended risk reduction taking into account the probabilities and consequences of random or systematic faults within the electrical control system.

Figure 1 shows the relationship of this standard to other relevant standards.

^{A1} Text deleted ^{A1}

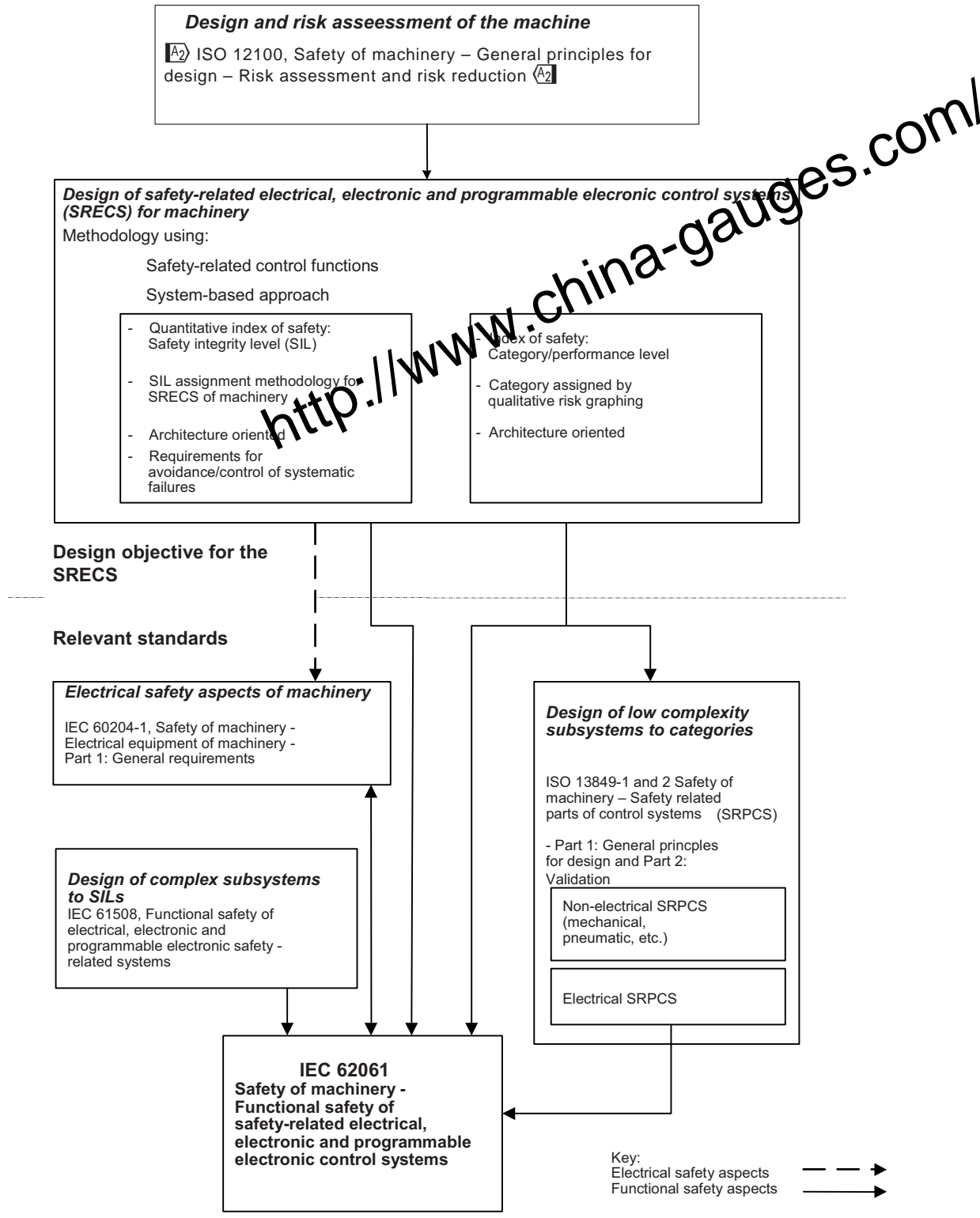


Figure 1 – Relationship of IEC 62061 to other relevant standards

[A1] Text deleted [A1]

[A1] IEC 62061 and ISO 13849-1 specify requirements for the design and implementation of safety-related control systems of machinery. The use of either of these standards, in accordance with their scopes, can be presumed to fulfil the relevant essential safety requirements. IEC/TR 62061-1 provides guidance on the application of IEC 62061 and ISO 13849-1 in the design of safety-related control systems for machinery. [A1]

[A1] Text deleted [A1]

SAFETY OF MACHINERY – FUNCTIONAL SAFETY OF SAFETY-RELATED ELECTRICAL, ELECTRONIC AND PROGRAMMABLE ELECTRONIC CONTROL SYSTEMS

1 Scope

This International Standard specifies requirements and makes recommendations for the design, integration and validation of safety-related electrical, electronic and programmable electronic control systems (SRECS) on machines (see Notes 1 and 2). It is applicable to control systems used, either singly or in combination, to carry out safety-related control functions on machines that are not portable by hand while working, including a group of machines working together in a co-ordinated manner.

NOTE 1 In this standard, the term “electrical control systems” is used to stand for “Electrical, Electronic and Programmable Electronic (E/E/PE) control systems” and “SRECS” is used to stand for “safety-related electrical, electronic and programmable electronic control systems”.

NOTE 2 ^{A1} In this standard, it is presumed that the design of complex programmable electronic subsystems or subsystem elements conforms to the relevant requirements of IEC 61508 and uses Route 1_H (see IEC 61508-2:2010, 7.4.4.2). It is considered that Route 2_H (see IEC 61508-2:2010, 7.4.4.3) is not suitable for general machinery. Therefore, this standard does not deal with Route 2_H. This standard provides a methodology for the use, rather than development, of such subsystems and subsystem elements as part of a SRECS. ^{A1}

This standard is an application standard and is not intended to limit or inhibit technological advancement. It does not cover all the requirements (e.g. guarding, non-electrical interlocking or non-electrical control) that are needed or required by other standards or regulations in order to safeguard persons from hazards. Each type of machine has unique requirements to be satisfied to provide adequate safety.

This standard:

- is concerned only with functional safety requirements intended to reduce the risk of injury or damage to the health of persons in the immediate vicinity of the machine and those directly involved in the use of the machine;
- is restricted to risks arising directly from the hazards of the machine itself or from a group of machines working together in a co-ordinated manner;

NOTE 3 Requirements to mitigate risks arising from other hazards are provided in relevant sector standards. For example, where a machine(s) is part of a process activity, the machine electrical control system functional safety requirements should, in addition, satisfy other requirements (e.g. IEC 61511) insofar as safety of the process is concerned.

- does not specify requirements for the performance of non-electrical (e.g. hydraulic, pneumatic) control elements for machines;

NOTE 4 Although the requirements of this standard are specific to electrical control systems, the framework and methodology specified can be applicable to safety-related parts of control systems employing other technologies.

- does not cover electrical hazards arising from the electrical control equipment itself (e.g. electric shock – see IEC 60204–1).

The objectives of specific Clauses in IEC 62061 are as given in Table 2.

Table 2 – Overview and objectives of IEC 62061

Clause	Objective
4: Management of functional safety	To specify the management and technical activities which are necessary for the achievement of the required functional safety of the SRECS.
5: Requirements for the specification of safety-related control functions	To set out the procedures to specify the requirements for safety-related control functions. These requirements are expressed in terms of functional requirements specification, and safety integrity requirements specification.
6: Design and integration of the safety-related electrical control system	To specify the selection criteria and/or the design and implementation methods of the SRECS to meet the functional safety requirements. This includes: selection of the system architecture, selection of the safety-related hardware and software, design of hardware and software, verification that the designed hardware and software meets the functional safety requirements.
7: Information for use of the machine	To specify requirements for the information for use of the SRECS, which has to be supplied with the machine. This includes: provision of the user manual and procedures, provision of the maintenance manual and procedures.
8: Validation of the safety-related electrical control system	To specify the requirements for the validation process to be applied to the SRECS. This includes inspection and testing of the SRECS to ensure that it achieves the requirements stated in the safety requirements specification.
9: Modification of the safety-related electrical control system	To specify the requirements for the modification procedure that has to be applied when modifying the SRECS. This includes: modifications to any SRECS are properly planned and verified prior to making the change; the safety requirements specification of the SRECS is satisfied after any modifications have taken place.

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.

IEC 60204–1, *Safety of machinery – Electrical equipment of machines – Part 1: General requirements*

IEC 61000-6-2, *Electromagnetic compatibility (EMC) – Part 6-2: Generic standards – Immunity for industrial environments*

IEC 61310 (all parts), *Safety of machinery – Indication, marking and actuation*

IEC 61508-2, *Functional safety of electrical/electronic/ programmable electronic safety-related systems – Part 2: Requirements for electrical/electronic/programmable electronic safety-related systems*

IEC 61508-3, *Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 3: Software requirements*

ISO 12100:2010, *Safety of machinery – General principles for design – Risk assessment and risk reduction*

ISO 13849-1:2006, *Safety of machinery – Safety-related parts of control systems – Part 1: General principles for design*

ISO 13849-2:2012, *Safety of machinery – Safety-related parts of control systems – Part 2: Validation*

Text deleted

3 Terms, definitions and abbreviations

3.1 Alphabetical list of definitions

Term	Definition number
application software	3.2.46
architectural constraint	3.2.36
architecture	3.2.35
common cause failure	3.2.43
complex component	3.2.8
control function	3.2.14
dangerous failure	3.2.40
demand	3.2.25
diagnostic coverage	3.2.38
electrical control system	3.2.3
embedded software	3.2.47
failure	3.2.39
fault	3.2.30
fault tolerance	3.2.31
full variability language (FVL)	3.2.48
function block	3.2.32
function block element	3.2.33

functional safety	3.2.9
hardware safety integrity	3.2.20
hazard (from machinery)	3.2.10
hazardous situation	3.2.3
high demand or continuous mode	3.2.27
limited variability language (LVL)	3.2.49
low complexity component	3.2.7
low demand mode	3.2.26
machine control system	3.2.2
machinery (machine)	3.2.1
mean time to failure (MTTF)	3.2.34
probability of dangerous failure per hour (PFH _D)	3.2.28
proof test	3.2.37
protective measure	3.2.12
random hardware failure	3.2.44
risk	3.2.13
safe failure	3.2.41
safe failure fraction	3.2.42
safety function	3.2.15
safety integrity	3.2.19
safety integrity level (SIL)	3.2.23
safety-related control function (SRCF)	3.2.16
safety-related electrical control system (SRECS)	3.2.4
safety-related software	3.2.50
SIL claim limit	3.2.24
software safety integrity	3.2.21
SRECS diagnostic function	3.2.17
SRECS fault reaction function	3.2.18
subsystem	3.2.5
subsystem element	3.2.6
systematic failure	3.2.45
systematic safety integrity	3.2.22
target failure value	3.2.29
validation	3.2.52
verification	3.2.51

3.2 Terms and definitions

For the purposes of this standard, the following terms and definitions apply.

3.2.1

machinery

assembly of linked parts or components, at least one of which moves, with the appropriate machine actuators, control and power circuits, joined together for a specific application, in particular for the processing, treatment, moving or packaging of a material.

The terms “machinery” and “machine” also cover an assembly of machines which, in order to achieve the same end, are arranged and controlled so that they function as an integral whole.

^{A2} [ISO 12100:2010, 3.1] ^{A2}

3.2.2

machine control system

system which responds to an input from, for example, the process, other machine elements, an operator, external control equipment, and generates an output(s) causing the machine to behave in the intended manner

3.2.3

electrical control system

all the electrical, electronic and programmable electronic parts of the machine control system used to provide, for example, operational control, monitoring, interlocking, communications, protection and safety-related control functions

NOTE Safety-related control functions can be performed by an electrical control system that is either integral to or independent of those parts of a machine's control system that perform non-safety-related functions.

3.2.4

Safety-Related Electrical Control System

SRECS

electrical control system of a machine whose failure can result in an immediate increase of the risk(s)

NOTE A SRECS includes all parts of an electrical control system whose failure may result in a reduction or loss of functional safety and this can comprise both electrical power circuits and control circuits.

3.2.5

subsystem

^{A1} entity of the top-level architectural design of the SRECS where a dangerous failure of any subsystem will result in a dangerous failure of a safety-related control function

^{A2} [IEC 61508-4:2010, 3.2.7] ^{A2}

NOTE 1 A complete subsystem can be made up from a number of identifiable and separate subsystem elements, which when put together implement the function blocks allocated to the subsystem.

NOTE 2 This differs from common language where “subsystem” may mean any sub-divided part of an entity, the term “subsystem” is used in this standard within a strongly defined hierarchy of terminology: “subsystem” is the first level subdivision of a system. The parts resulting from further subdivision of a subsystem are called “subsystem elements”. ^{A1}

3.2.6

subsystem element



part of a subsystem, comprising a single component or any group of components

3.2.7

low complexity component

component in which

- the failure modes are well-defined; and
- the behaviour under fault conditions can be completely defined

 [IEC 61508-4:2010, 3.4.3] 

NOTE 1 Behaviour of the low complexity component under fault conditions shall be determined by analytical and/or test methods.

NOTE 2 A subsystem or subsystem element comprising one or more limit switches, operating, possibly via interposing electro-mechanical relays, one or more contactors to re-energise an electric motor is an example of a low complexity component.

3.2.8

complex component



component in which

- the failure modes are not well-defined; or
- the behaviour under fault conditions cannot be completely defined

3.2.9

functional safety

part of the safety of the machine and the machine control system which depends on the correct functioning of the SRECS, other technology safety-related systems and external risk reduction facilities

 [IEC 61508-4:2010, 3.1.12] 



NOTE 1 This standard only considers the functional safety that depends on the correct functioning of the SRECS in machinery applications.

NOTE 2 ISO/IEC Guide 51 defines safety as freedom from unacceptable risk.

3.2.10

hazard (from machinery)

potential source of physical injury or damage to health

 [ISO 12100:2010, 3.6] 

NOTE The term hazard can be qualified in order to define its origin or the nature of the expected harm (e.g. electric shock hazard, crushing hazard, cutting hazard, toxic hazard, fire hazard).

3.2.11

hazardous situation

circumstance in which a person is exposed to a hazard(s)

 [ISO 12100:2010, 3.10] 

3.2.12

protective measure

measure intended to achieve risk reduction



 [ISO 12100:2010, 3.19] 

<http://www.china-gauges.com/>

3.2.13

risk

combination of the probability of occurrence of harm and the severity of that harm

 [ISO 12100:2010, 3.12] 

3.2.14



control function

function that evaluates input information or signals and produces output information or activities

3.2.15

safety function

function of a machine whose failure can result in an immediate increase of the risk(s)

 [ISO 12100:2010, 3.30] 

NOTE This definition differs from the definitions in IEC 61508-4 and ISO 13849-1.

3.2.16

Safety-Related Control Function

SRCF

control function implemented by a SRECS with a specified integrity level that is intended to maintain the safe condition of the machine or prevent an immediate increase of the risk(s)

3.2.17

SRECS diagnostic function

function intended to detect faults in the SRECS and produce a specified output information or activity when a fault is detected

NOTE This function is intended to detect faults that could lead to a dangerous failure of a SRCF and initiate a specified fault reaction function.

3.2.18



SRECS fault reaction function

function that is initiated when a fault within a SRECS is detected by the SRECS diagnostic function

3.2.19

safety integrity

probability of a SRECS or its subsystem satisfactorily performing the required safety-related control functions under all stated conditions

 [IEC 61508-4:2010, 3.5.4] 



NOTE 1 The higher the level of safety integrity of the item, the lower the probability that the item will fail to carry out the required safety-related control function.

NOTE 2 Safety integrity comprises hardware safety integrity (see 3.2.20) and systematic safety integrity (see 3.2.22).

3.2.20



hardware safety integrity

part of the safety integrity of a SRECS or its subsystems comprising requirements for both the probability of dangerous random hardware failures and architectural constraints

 [IEC 61508-4:2010, 3.5.7] 

3.2.21**software safety integrity**



part of the systematic safety integrity of a SRECS or its subsystems related to the capability of software in a programmable electronic system performing its safety-related control functions under all stated conditions during a stated period of time

 [IEC 61508-4:2010, 3.5.5] 

NOTE Software safety integrity cannot usually be quantified precisely.

3.2.22**systematic safety integrity**

part of the safety integrity of a SRECS or its subsystems relating to its resistance to systematic failures (see 3.2.45) in a dangerous mode.



 [IEC 61508-4:2010, 3.5.6] 

NOTE 1 Systematic safety integrity cannot usually be quantified precisely.

NOTE 2 Requirements for systematic safety integrity apply to both hardware and software aspects of a SRECS or its subsystems.

3.2.23**Safety Integrity Level****SIL**

discrete level (one out of a possible three) for specifying the safety integrity requirements of the safety-related control functions to be allocated to the SRECS, where safety integrity level three has the highest level of safety integrity and safety integrity level one has the lowest

 [IEC 61508-4:2010, 3.5.8] 

NOTE SIL 4 is not considered in this standard, as it is not relevant to the risk reduction requirements normally associated with machinery. For requirements applicable to SIL 4, see IEC 61508-1 and IEC 61508-2.



3.2.24**SIL Claim Limit (for a subsystem)****SILCL**

maximum SIL that can be claimed for a SRECS subsystem in relation to architectural constraints and systematic safety integrity

3.2.25**demand**



event that causes the SRECS to perform its SRCF



3.2.26**low demand mode**

 mode of operation in which the frequency of demands on a SRECS is no greater than one per year 

NOTE Equipment that is only designed in accordance with requirements for the low demand mode of operation described in IEC 61508-1 and IEC 61508-2 can be unsuitable for use as part of a SRECS in this standard. Low demand mode of operation is not considered to be relevant for SRECS applications at machinery.

3.2.27**high demand or continuous mode**

 mode of operation in which the frequency of demands on a SRECS is greater than one per year or the SRCF retains the machine in a safe state as part of normal operation 

 [IEC 61508-4:2010, 3.5.16] 

NOTE 1 Low demand mode of operation is not considered to be relevant for SRECS applications at machinery. Therefore, in this standard SRECS are only considered to operate in the high demand or continuous mode.

NOTE 2 Demand mode means that a safety-related control function is only performed on request (demand) in order to transfer the machine into a specified state. The SRECS does not influence the machine until there is a demand on the safety-related control function.

NOTE 3 Continuous mode means that a safety-related control function is performed perpetually (continuously), i.e. the SRECS is continuously controlling the machine and a (dangerous) failure of its function can result in a hazard.

3.2.28

Probability of dangerous Failure per Hour

PFH_D

$\overline{A_1}$ average probability of a dangerous failure per hour of a safety related system/subsystem to perform the specified safety function over a given period of time

NOTE 1 PFH_D should not be confused with probability of dangerous failure on demand (PDF). $\overline{A_1}$

$\overline{A_2}$ NOTE 2 Within this standard it is expressed as the constant failure rate with respect to 1 hour. $\overline{A_2}$

3.2.29

target failure value

intended PFH_D to be achieved to meet a specific safety integrity requirement(s)

NOTE Target failure value is specified in terms of the probability of dangerous failure per hour.

$\overline{A_2}$ [IEC 61508-4:2010, 3.5.17] $\overline{A_2}$

3.2.30

fault

abnormal condition that may cause a reduction in or loss of, the capability of a SRECS, a subsystem, or a subsystem element to perform a required function

$\overline{A_2}$ [IEC 61508-4:2010, 3.6.1] $\overline{A_2}$

3.2.31

fault tolerance

ability of a SRECS, a subsystem, or subsystem element to continue to perform a required function in the presence of faults or failures

$\overline{A_2}$ [IEC 61508-4:2010, 3.6.3] $\overline{A_2}$

3.2.32

function block

smallest element of a SRCF whose failure can result in a failure of the SRCF

NOTE 1 In this standard, a SRCF (F) may be seen as a logical AND of the function blocks (FB), i.e. $F = FB_1 \text{ AND } FB_2 \text{ AND } FB_n$.

NOTE 2 This definition of a function block differs from those used in IEC 61131-3 and other standards.

3.2.33

function block element

part of a function block

3.2.34

Mean Time To Failure

MTTF

expectation of the mean time to failure

[IEV 191-12-07, modified]

NOTE MTTF is normally expressed as an average value of expectation of the time to failure.

3.2.35**architecture**

specific configuration of hardware and software elements in a SRECS

\square_{A2} [IEC 61508-4:2010, 3.3.4] \square_{A2}

3.2.36**architectural constraint**

set of architectural requirements that limit the SIL that can be claimed for a subsystem

NOTE Requirements for architectural constraints are given in 6.7.1.

3.2.37**proof test**

\square_{A1} periodic test performed to detect dangerous hidden failures and degradation in a SRECS and its subsystems so that, if necessary, the SRECS and its subsystems can be restored to an "as new" condition or as close as practical to this condition \square_{A1}

\square_{A2} [IEC 61508-4:2010, 3.8.5] \square_{A2}

NOTE A proof test is intended to confirm that the SRECS is in a condition that assures the specified safety integrity.

3.2.38**diagnostic coverage**

\square_{A1} fraction of dangerous failures detected by automatic on-line diagnostic test \square_{A1}

\square_{A2} [IEC 61508-4:2010, 3.8.6] \square_{A2}

\square_{A1} NOTE 1 \square_{A1} Diagnostic coverage (DC) can be calculated using the following equation:

$$DC = \Sigma \lambda_{DD} / \lambda_{Dtotal}$$

where λ_{DD} is the rate of detected dangerous hardware failures and λ_{Dtotal} is the rate of total dangerous hardware failures.

\square_{A1} NOTE 2 The fraction of detected dangerous failures is computed to be the rate of dangerous failures that are detected by automatic on-line diagnostic tests divided by the rate of total dangerous failures. \square_{A1}

3.2.39**failure**

termination of the ability of a SRECS, a subsystem, or a subsystem element to perform a required function

\square_{A2} [IEC 61508-4:2010, 3.6.4 modified and ISO 12100:2010, 3.34] \square_{A2}

NOTE Failures are either random (in hardware) or systematic (in hardware or software).

3.2.40**dangerous failure**

failure of a SRECS, a subsystem, or a subsystem element that has the potential to cause a hazard or non-functional state

\square_{A1} Text deleted \square_{A1}

NOTE 1 Whether or not the potential is realised can depend on the channel architecture of the system; for example, in systems with multiple channels to improve safety, a dangerous hardware failure is less likely to lead to the overall dangerous or fail-to function state.

NOTE 2 In a subsystem with multiple channels, the probability of dangerous failure of the subsystem can be smaller than the dangerous failure rate of a channel that constitutes the subsystem. The probability of dangerous failure of a SRECS cannot be smaller than that of any subsystem constituting the SRECS. (This comes from the particular definition of "subsystem" in this standard.)

NOTE 3 A dangerous failure normally results in a failure or potential failure to perform the SRCF.

3.2.41

safe failure

failure of a SRECS, a subsystem of a SRECS, or a subsystem element of a SRECS that does not have the potential to cause a hazard

$\overline{A_1}$ Text deleted $\overline{A_1}$

$\overline{AC_2}$ Note deleted $\overline{AC_2}$

3.2.42

Safe Failure Fraction

SFF

fraction of the overall failure rate of a subsystem that does not result in a dangerous failure

NOTE Safe Failure Fraction (SFF) can be calculated using the following equation:

$$\frac{\Sigma \lambda_S + \Sigma \lambda_{DD}}{\Sigma \lambda_S + \Sigma \lambda_D}$$

where

λ_S is the rate of safe failure,

$\Sigma \lambda_S + \Sigma \lambda_D$ is the overall failure rate,

λ_{DD} is the rate of dangerous failure which is detected by the diagnostic functions, and

λ_D is the rate of dangerous failure.

The diagnostic coverage (if any) of each subsystem in SRECS is taken into account in the calculation of the probability of random hardware failures. The safe failure fraction is taken into account when determining the architectural constraints on hardware safety integrity (see 6.7.7).

3.2.43

Common Cause Failure

CCF

failure, which is the result of one or more events, causing $\overline{A_1}$ concurrent $\overline{A_1}$ failures of two or more separate channels in a multiple channel (redundant architecture) subsystem, leading to failure of a SRCF

$\overline{A_2}$ [IEC 61508-4:2010, 3.6.10] $\overline{A_2}$

NOTE This definition differs from that given in $\overline{A_2}$ ISO 12100 $\overline{A_2}$ and IECV 191-04-23.

3.2.44

random hardware failure

failure occurring at a random time, which results from one or more of the possible degradation mechanisms in the hardware

$\overline{A_2}$ [IEC 61508-4:2010, 3.6.5] $\overline{A_2}$

3.2.45

systematic failure

failure related in a deterministic way to a certain cause, which can only be eliminated by a modification of the design or of the manufacturing process, operational procedures, documentation or other relevant factors

$\overline{A_2}$ [IEC 61508-4:2010, 3.6.6] $\overline{A_2}$

NOTE 1 Corrective maintenance without modification will usually not eliminate the failure cause.

NOTE 2 A systematic failure can be induced by simulating the failure cause.

NOTE 3 Examples of causes of systematic failures include human error in

- the safety requirements specification;
- the design, manufacture, installation and/or operation of the hardware;
- the design and/or implementation of the software.

3.2.46

application software

software specific to the application, that is implemented by the designer of the SRECS, generally containing logic sequences, limits and expressions that control the appropriate input, output, calculations, and decisions necessary to meet the SRECS functional requirements

3.2.47

embedded software

software, supplied by the manufacturer, that is part of the SRECS and that is not normally accessible for modification

NOTE Firmware and system software are examples of embedded software.

3.2.48

Full Variability Language

FVL

type of language that provides the capability to implement a wide variety of functions and applications

 [IEC 61511-1:2003, 3.2.81.1.3] 

NOTE 1 Typical example of systems using FVL are general-purpose computers.

NOTE 2 FVL is normally found in embedded software and is rarely used in application software.

NOTE 3 FVL examples include: Ada, C, Pascal, Instruction List, assembler languages, C++, Java, SQL.

3.2.49

Limited Variability Language

LVL

type of language that provides the capability to combine predefined, application specific, library functions to implement the safety requirements specifications

 [IEC 61511-1:2003, 3.2.81.1.2] 

NOTE 1 A LVL provides a close functional correspondence with the functions required to achieve the application.

NOTE 2 Typical examples of LVL are given in IEC 61131-3. They include ladder diagram, function block diagram and sequential function chart. Instruction lists and structured text are not considered to be LVL.

NOTE 3 Typical example of systems using LVL: Programmable Logic Controller (PLC) configured for machine control.

3.2.50

safety-related software

software that is used to implement safety-related control functions in a safety-related system

3.2.51

verification

confirmation by examination (e.g. tests, analysis) that the SRECS, its subsystems or subsystem elements meet the requirements set by the relevant specification

 [IEC 61508-4:2010, 3.8.1 and IEC 61511-1:2003, 3.2.92] 

NOTE The verification results should provide documented objective evidence.

EXAMPLE: Verification activities include:

- reviews on outputs (documents from all phases) to ensure compliance with the objectives and requirements of the phase, taking into account the specific inputs to that phase;
- design reviews;
- tests performed on the designed products to ensure that they perform according to their specification;
- integration tests performed where different parts of a system are put together in a step-by-step manner and by the performance of environmental tests to ensure that all the parts work together in the specified manner.

**3.2.52
 validation**

confirmation by examination (e.g. tests, analysis) that the SRECS meets the functional safety requirements of the specific application

A2 [IEC 61508-4:2010, 3.8.2] A2

3.3 Abbreviations

The following abbreviations are used in this standard.

CCF	Common Cause Failure(s)
DC	Diagnostic Coverage
EMC	Electromagnetic Compatibility
FB	Function Block
FVL	Full Variability Language
I/O	Input/Output
LVL	Limited Variability Language
PFH_D	Probability of dangerous Failure per Hour
MTTF	Mean Time To Failure
MTTR	Mean Time To Restoration
P_{TE}	Probability of dangerous Transmission Error
SFF	Safe Failure Fraction
SIL	Safety Integrity Level
SILCL	Safety Integrity Level (SIL) Claim Limit (for subsystems)
S-R	Safety Related
SRECS	Safety-Related Electrical Control System
SRCF	Safety-Related Control Function
SRS	Safety Requirements Specification
SYS	System

<http://www.china-gauges.com/>

4 Management of functional safety

4.1 Objective

This Clause specifies management and technical activities that are necessary for the achievement of the required functional safety of the SRECS.

4.2 Requirements

4.2.1 A functional safety plan shall be drawn up and documented for each SRECS design project, and shall be updated as necessary. The plan shall include procedures for control of the activities specified in Clauses 5 to 9.

NOTE 1 The content of the functional safety plan should depend upon the specific circumstances, which can include:

- size of project;
- degree of complexity;
- degree of novelty of design and technology;
- degree of standardization of design features;
- possible consequence(s) in the event of failure.

In particular the plan shall:

- a) identify the relevant activities specified in Clauses 5 to 9.
- b) describe the policy and strategy to fulfil the specified functional safety requirements.
- c) describe the strategy to achieve functional safety for the application software, development, integration, verification and validation.
- d) identify persons, departments or other units and resources that are responsible for carrying out and reviewing each of the activities specified in Clauses 5 to 9.
- e) identify or establish the procedures and resources to record and maintain information relevant to the functional safety of a SRECS.

NOTE 2 The following should be considered:

- the results of the hazard identification and risk assessment;
 - the equipment used for safety-related functions together with its safety requirements;
 - the organization responsible for maintaining functional safety;
 - the procedures necessary to achieve and maintain functional safety (including SRECS modifications).
- f) describe the strategy for configuration management (see 9.3) taking into account relevant organizational issues, such as authorized persons and internal structures of the organization.
 - g) establish a verification plan that shall include:
 - details of when the verification shall take place;
 - details of the persons, departments or units who shall carry out the verification;
 - the selection of verification strategies and techniques;
 - the selection and utilization of test equipment;
 - the selection of verification activities;
 - acceptance criteria; and
 - the means to be used for the evaluation of verification results.

h) establish a validation plan comprising:

- details of when the validation shall take place;
- identification of the relevant modes of operation of the machine (e.g. normal operation, setting);
- requirements against which the SRECS is to be validated;
- the technical strategy for validation, for example analytical methods or statistical tests;
- acceptance criteria; and
- actions to be taken in the event of failure to meet the acceptance criteria.

NOTE 3 The validation plan should indicate whether the SRECS and its subsystems are to be subject to routine testing, type testing and/or sample testing.

4.2.2 The functional safety plan shall be implemented to ensure prompt follow-up and satisfactory resolution of issues relevant to a SRECS arising from:

- activities specified in Clauses 5 to 9;
- verification activities; and
- validation activities.

5 Requirements for the specification of Safety-Related Control Functions (SRCFs)

5.1 Objective

This Clause sets out the procedures to specify the requirements of SRCF(s) to be implemented by the SRECS.

5.2 Specification of requirements for SRCFs

5.2.1 General

5.2.1.1 From the risk reduction strategy, as outlined in **ISO 12100**, any need for safety functions will be determined.

5.2.1.2 Where safety functions are selected to be implemented (in whole or in part) by SRECS, then the associated SRCF(s) (see 3.2.16) shall be specified.

5.2.1.3 Specifications of each SRCF shall comprise:

- functional requirements specification (see 5.2.3);
- safety integrity requirements specification (see 5.2.4).

and these shall be documented in the safety requirements specification (SRS).

NOTE 1 Where non-electrical equipment contributes towards the performance of a safety function in combination with electrical means, the target failure value(s) applicable to the non-electrical equipment is not considered within this standard. Electrical means covers any and all devices or systems operating on electrical principles, including:

- electro-mechanical devices;
- non-programmable electronic devices;
- programmable electronic devices.

NOTE 2 The SRS needs to be subject to version control as part of the configuration management procedures (see 9.3).

5.2.1.4 The safety requirements specification shall be verified to ensure consistency and completeness for its intended use.

NOTE For example this may be achieved by inspection, analysis, check-lists. See also B.2.6 on [A2](#) IEC 61508-7:2010 [A2](#).

5.2.2 Information to be available

The following information shall be used to produce both the functional requirements specification and safety integrity requirements specification of each SRCF:

- results of the risk assessment for the machine including all safety functions determined to be necessary for the risk reduction process for each specific hazard;
- machine operating characteristics, including:
 - modes of operation,
 - cycle time,
 - response time performance,
 - environmental conditions,
 - interaction of person(s) with the machine (e.g. repairing, setting, cleaning);
- all information relevant to the SRCFs which can have an influence on the SRECS design including, for example:
 - a description of the behaviour of the machine that a SRCF is intended to achieve or to prevent;
 - all interfaces between the SRCFs, and between SRCFs and any other function (either within or outside the machine);
 - required fault reaction functions of the SRCF.

NOTE Some of the information might not be available or sufficiently defined before starting the iterative design process of SRECS, so the SRECS safety requirements specifications can be required to be updated during the design process.

5.2.3 Functional requirements specification for SRCFs

[A1](#) The functional requirements specification for SRCFs shall describe details of each SRCF to be performed including, as applicable:

- the condition(s) (e.g. operating mode) of the machine in which the SRCF shall be active or disabled;
- the priority of those functions that can be simultaneously active and that can cause conflicting action;
- the frequency of operation of each SRCF;
- the required response time of each SRCF;
- the interface(s) of the SRCFs to other machine functions;
- the required response times (e.g. input and output devices);
- a description of each SRCF;
- a description of fault reaction function(s) and any constraints on, for example, re-starting or continued operation of the machine in cases where the initial fault reaction is to stop the machine;
- a description of the operating environment (e.g. temperature, humidity, dust, chemical substances, mechanical vibration and shock); [A1](#)

- Ⓐ) – tests and any associated facilities (e.g. test equipment, test access ports);
- rate of operating cycles, duty cycle, and/or utilisation category, for electromechanical devices intended for use in the SRCF.

NOTE 1 In addition to the requirements of IEC 61000-6-2, when a SRECS is intended for use in an industrial environment, electromagnetic (EM) immunity levels are given in IEC 61326-3-1. SRECS intended for use in another EM environment (e.g. residential) should have immunity levels based on those specified in different EMC standards (e.g., for a residential environment, IEC 61000-6-1).

NOTE 2 When specifying EM immunity levels it is necessary to consider whether the levels used in different EMC standards cover cases which can occur in a SRECS application even with a low probability of occurrence.

NOTE 3 EM immunity performance criterion for functional safety of a SRECS is given in 6.4.3. Ⓐ)

5.2.4 Safety integrity requirements specification for SRCFs

5.2.4.1 The safety integrity requirements for each SRCF shall be derived from the risk assessment to ensure the necessary risk reduction can be achieved. In this standard, a safety integrity requirement is expressed as a target failure value for the probability of dangerous failure per hour of each SRCF.

5.2.4.2 The safety integrity requirements for each SRCF shall be specified in terms of a SIL in accordance with Table 3 and documented. An example of a methodology is given in Annex A.

Table 3 – Safety integrity levels: target failure values for SRCFs

Safety integrity level	Probability of a dangerous Failure per Hour (PFH_D)
3	$\geq 10^{-8}$ to $< 10^{-7}$
2	$\geq 10^{-7}$ to $< 10^{-6}$
1	$\geq 10^{-6}$ to $< 10^{-5}$

NOTE Where the required safety integrity of a SRCF is less than SIL 1, as a minimum the requirements of category B of ISO 13849-1 should be met.

5.2.4.3 Where a product standard specifies a SIL for a SRCF then this shall take precedence over Annex A.

6 Design and integration of the safety-related electrical control system (SRECS)

6.1 Objective

This Clause specifies requirements for the selection or design of a SRECS to meet the functional and safety integrity requirements specified in the safety requirements specification (see 5.2).

6.2 General requirements

6.2.1 The SRECS shall be selected or designed to meet the safety requirements specification (see 5.2) and where relevant the software safety requirements specification (see 6.10) taking into account the appropriate requirements of this standard.

6.2.2 The selection or design of the SRECS (including the overall hardware and software architecture, sensors, actuators, programmable electronics, embedded software, application software, etc.) shall comply with either 6.5 or 6.6. Whichever method is used, the SRECS shall meet the following requirements:

- a) the requirements for hardware safety integrity comprising:
 - the architectural constraints on hardware safety integrity (see 6.6.3.3); and
 - the requirements for the probability of dangerous random hardware failures (see 6.6.3.2);
- b) the requirements for systematic safety integrity (see 6.4) comprising:
 - the requirements for the avoidance of failures, and
 - the requirements for the control of systematic faults;
- c) the requirements for SRECS behaviour on detection of a fault (see 6.3);
- d) the requirements for the design and development of safety-related software (see 6.10 and 6.11).

6.2.3 The design of the SRECS shall take into account human capabilities and limitations (including reasonably foreseeable misuse) and be suitable for the actions assigned to operators, maintenance staff and others who might interact with the SRECS. The design of all operator interfaces shall follow good human-factor practice (see the IEC 61310 series) and shall accommodate the likely level of training or awareness of operators, in particular, for mass-produced subsystems where the operator can be a member of the public.

NOTE The design goal should be that reasonably foreseeable mistakes made by operators or maintenance staff are prevented or eliminated by design. Where this is not possible, other means should also be applied (e.g. manual action with secondary confirmation before completion) to minimize the possibility of operator errors and ensure that foreseeable mistakes do not lead to increased risk.

6.2.4 Maintainability and testability shall be considered during the design and integration to facilitate the implementation of these properties in the SRECS.

6.2.5 The SRECS design, including its diagnostic and fault reaction functions, shall be documented. This documentation shall:

- be accurate, complete and concise;
- be suitable for its intended purpose;
- be accessible and maintainable;
- be version controlled.

6.2.6 The outcome of the activities performed during SRECS design, development and implementation shall be verified at appropriate stages.

6.3 Requirements for behaviour (of the SRECS) on detection of a fault in the SRECS

6.3.1 The detection of a dangerous fault in any subsystem that has a hardware fault tolerance of more than zero shall result in the performance of the specified fault reaction function.

The specification may allow isolation of the faulty part of the subsystem to continue safe operation of the machine while the faulty part is repaired. In this case, if the faulty part is not repaired within the estimated maximum time as assumed in the calculation of the probability of random hardware failure (see 6.7.8), then a second fault reaction shall be performed to maintain a safe condition.

Where the SRECS is designed for online repair, isolation of a faulty part shall only be applicable where this does not increase the probability of dangerous random hardware failure of the SRECS above that specified in the SRS.

After the occurrence of faults that reduce the hardware fault tolerance to zero, the requirements of 6.3.2 apply.

NOTE The mean time to restoration (see IEC 60050-191-13-08) that is considered in the reliability model will need to take into account the diagnostic test interval, the repair time and any other delays prior to restoration.

6.3.2 Where a diagnostic function(s) is necessary to achieve the required probability of dangerous random hardware failure and the subsystem has a hardware fault tolerance of zero, then the fault detection and specified fault reaction shall be performed before the hazardous situation addressed by the SRCF can occur.

EXCEPTION to 6.3.2: In the case of a subsystem implementing a particular SRCF where the hardware fault tolerance is zero and the ratio of the diagnostic test rate to the demand rate exceeds 100, then the diagnostic test interval of that subsystem shall be such as to enable the subsystem to meet the requirement for the probability of dangerous random hardware failure.

6.3.3 Where performance of a fault reaction function as part of a SRCF that is specified as SIL 3 has resulted in the machine being stopped, subsequent normal operation of the machine via the SRECS (e.g. enabling re-start of the machine) shall not be possible until the fault has been repaired or rectified. For SRCFs with a specified safety integrity of less than SIL 3, the behaviour of the machine after performance of a fault reaction function (e.g. re-starting normal operation) shall depend on the specification of relevant fault reaction functions (see 5.2.3).

6.4 Requirements for systematic safety integrity of the SRECS

A1 Note deleted **A1**

6.4.1 Requirements for the avoidance of systematic hardware failures

6.4.1.1 The following measures shall be applied:

- a) the SRECS shall be designed and implemented in accordance with the functional safety plan (see 4.2);
- b) proper selection, combination, arrangements, assembly and installation of subsystems, including cabling, wiring and any interconnections;
- c) use of the SRECS within the manufacturer's specification;
- d) use of manufacturer's application notes, for example catalogue sheets, installation instructions, and use of good engineering practice (see also **A2** ISO 13849-2:2012 **A2**, Clause D.1);
- e) use of subsystems that have compatible operating characteristics (see also **A2** ISO 13849-2:2012 **A2**, Clause D.1);
- f) the SRECS shall be protected in accordance with IEC 60204-1;
- g) prevention of the loss of functional earth connection(s) in accordance with IEC 60204-1;
- h) undocumented modes of component operation shall not be used (e.g. 'reserved' registers of programmable equipment); and
- i) consideration of foreseeable misuse, environmental changes or modification(s).

6.4.1.2 In addition, at least one of the following techniques and/or measures shall be applied taking into account the complexity of the SRECS and the SIL(s) for those functions to be implemented by the SRECS:

- a) SRECS hardware design review (e.g. by inspection or walk-through): to establish by reviews and/or analysis any discrepancies between the specification and implementation;

NOTE 1 In order to reveal discrepancies between the specification and implementation, any points of doubt or potential weak points concerning the realisation, the implementation and the use of the product are documented so they can be resolved; taking into account that on an inspection procedure the author is passive and the inspector is active whilst on a walk-through procedure the author is active and the inspector is passive.

- b) advisory tools such as computer-aided design packages capable of simulation or analysis, and/or the use of computer-aided design tools to perform the design procedures systematically with the use of pre-designed elements that are already available and tested;

NOTE 2 The integrity of these tools can be demonstrated by specific testing, or by an extensive history of satisfactory use, or by independent verification of their output for the particular SRECS that is being designed. See 6.11.3.4.

- c) simulation: perform a systematic and complete assimilation of a SRECS design in terms of both functional performance and the correct dimensioning and interaction of its subsystems.

EXAMPLE The function of the SRECS can be simulated on a computer via a software behavioural model (see 6.11.3.4) where individual subsystems or subsystem elements each have their own simulated behaviour, and the response of the circuit in which they are connected is examined by looking at the marginal data of each subsystem or subsystem element.

6.4.2 Requirements for the control of systematic faults

The following measures shall be applied:

- a) use of de-energization: the SRECS shall be designed so that with loss of its electrical supply a safe state of the machine is achieved or maintained;
- b) measures to control the effect of temporary subsystem failures: the SRECS shall be designed so that, for example:

- voltage variation (e.g. interruptions, dips) to an individual subsystem or a part of a subsystem does not lead to a hazard (e.g. a voltage interruption that affects a motor circuit shall not cause an unexpected start-up when the supply is restored), and

NOTE 1 See also relevant requirements of IEC 61010-1. In particular:

overvoltage or undervoltage should be detected early enough so that all outputs can be switched to a safe condition by the power-down routine or a switch-over to a second power unit; and/or

where necessary, overvoltage or undervoltage should be detected early enough so that the internal state can be saved in non-volatile memory, so that all outputs can be set to a safe condition by the power-down routine, or all outputs can be switched to a safe condition by the power-down routine or a switch-over to a second power unit.

- the effects of electromagnetic interference from the physical environment or a subsystem(s) do not lead to a hazard;
- c) measures to control the effects of errors and other effects arising from any data communication process, including transmission errors, repetitions, deletion, insertion, re-sequencing, corruption, delay and masquerade;

NOTE 2 ^{A1} Further information can be found in IEC 61784-3 and IEC 61508-2. ^{A1}

NOTE 3 The term 'masquerade' means that the true contents of a message are not correctly identified. For example, a message from a non-safety component is incorrectly identified as a message from a safety component.

- d) when a dangerous fault occurs at an interface, the fault reaction function shall be performed before the hazard due to this fault can occur. When a fault that reduces the hardware fault tolerance to zero occurs, this fault reaction shall take place before the estimated MTTR (see 6.7.4.4.2 g) is exceeded.

The requirements of item d) apply to interfaces that are inputs and outputs of subsystems and all other parts of subsystems that include or require cabling during integration (for example output signal switching devices of a light curtain, output of a guard position sensor).

NOTE 4 This does not require that a subsystem or subsystem element on its own has to detect a fault on its outputs(s). The fault reaction function may also be initiated by any subsequent subsystem after a diagnostic test is performed.

6.4.3 Electromagnetic (EM) immunity

In addition to the requirements of IEC 61000-6-2 and the EM phenomena given in ^{A1} IEC 61326-3-1 ^{A1}, the following performance criterion for functional safety shall be satisfied by a SRECS:

- unsafe conditions or hazards shall not be introduced; and
- no loss of the SRCF(s); or

- the SRCF(s) implemented by the SRECS may be disturbed temporarily or permanently provided that a safe state of the machine is maintained or achieved before a hazard can occur. Where the EM phenomena can result in the destruction of components, it shall be ensured (e.g. by analysis) that functional safety is not affected, including by lower value(s) of EM phenomena that can cause partial destruction.

NOTE Consideration should be given to the behaviour of the SRECS in response to EM phenomena at all value(s) up to those given in A_1 IEC 61326-3-1 A_1 .

6.5 Selection of safety-related electrical control system

Where a supplier provides a SRECS for a specific function referenced in the safety requirements specification, a pre-designed SRECS may be selected instead of a custom design providing that it meets the requirements of the safety requirements specification and 6.3, 6.4 and 6.6.1.

NOTE Selection of a pre-designed SRECS is an alternative to the design and development of a specific SRECS in accordance with 6.6.

6.6 Safety-related electrical control system (SRECS) design and development

6.6.1 General requirements

6.6.1.1 The SRECS shall be designed and developed in accordance with the SRECS safety requirements specification (see 5.2).

6.6.1.2 A clearly structured design process shall be followed and documented (see 6.6.2).

6.6.1.3 Where the use of diagnostics is necessary to achieve the required safety integrity when a fault is detected, the SRECS shall perform the specified fault reaction function (see 5.2 and 6.3).

6.6.1.4 Where a SRECS or part of a SRECS (i.e. its subsystem(s)) is to implement both SRCFs and other functions, then all its hardware and software shall be treated as safety-related unless it can be shown that the implementation of the SRCFs and other functions is sufficiently independent (i.e. that the normal operation or failure of any other functions do not affect the SRCFs).

NOTE Sufficient independence of implementation can be established by showing that the probability of a dependent failure between the non-safety and safety-related parts is equivalent to that of the safety integrity level of the SRECS.

6.6.1.5 For a SRECS or its subsystems that implements safety-related control functions of different safety integrity levels, its hardware and software shall be treated as requiring the highest safety integrity level unless it can be shown that the implementation of the safety-related control functions of the different safety integrity levels is sufficiently independent.

NOTE Sufficient independence of implementation can be established by showing that the probability of a dependent failure between the parts implementing SRCFs of different integrity levels is equivalent to that of the safety integrity level achieved by the SRECS.

6.6.1.6 Interconnections (e.g. wiring, cabling) other than digital data communication shall be considered to be part of one of the subsystems to which they are connected (see also item d) of 6.4.2).

6.6.1.7 Where digital data communication is used as a part of a SRECS implementation it shall satisfy the relevant requirements of IEC 61508-2 in accordance with the SIL target(s) of the SRCF(s).

6.6.1.8 The information for use of the SRECS shall specify those techniques and measures necessary during the design life of the SRECS to maintain the safety integrity level.

6.6.2 Design and development process

The design and development shall follow a clearly defined process that shall take into account all aspects covered by the process shown in Figure 4.

NOTE The approach of this standard is to apply a structured design process to the SRECS beginning from the requirements that are specified in the Safety Requirements Specification. Figure 3 shows the workflow of the design process and the terminology that applies to the different levels.

6.6.2.1 System architecture design

6.6.2.1.1 Each SRCF as specified in the SRECS safety requirements specification shall be decomposed to a structure of function blocks, for example as shown in Figure 3. This structure shall be documented comprising:

- the description of the structure;
- the safety requirements (functional, integrity,) for each function block;
- definition of inputs and outputs of each function block.

NOTE 1 The decomposition process should lead to a structure of function blocks that fully describes the functional and integrity requirements of the SRCF. This process should be applied down to that level that permits the functional and integrity requirements determined for each function block to be allocated to subsystems, where the allocation to a subsystem of the complete functional requirements of a function block is possible. However, it is possible to allocate more than one function block to a single subsystem, but it is not possible to allocate one function block to several subsystems where it is intended that these subsystems have separate functional and integrity requirements. Where the intention is to allocate the functional requirements of one function block to redundant subsystem elements, refer to 6.7.4.

NOTE 2 The inputs and outputs of each function block are the information that is transferred, for example speed, position, mode of operation, etc.

NOTE 3 The function blocks are a representation of functions of the SRCF (see 3.2.16) and do not include SRECS diagnostic functions (see 3.2.17). For the purposes of this standard, the diagnostic functions are considered as separate functions that may have a different structure to the SRCF (see 6.8).

6.6.2.1.2 An initial concept for an architecture of the SRECS shall be created in accordance with the structure of the function blocks.

NOTE There should be ongoing collaboration between the developer of the safety-related control architecture, the organization responsible for configuration of the devices and the developer of the software. As the software safety requirements and the possible software architecture become more precise, there may be an impact on the SRECS hardware architecture, and for this reason close co-operation between the SRECS architecture designer, the subsystem supplier(s), software developer and, as necessary, the machinery designer or the user can help to reduce the potential for systematic failure(s).

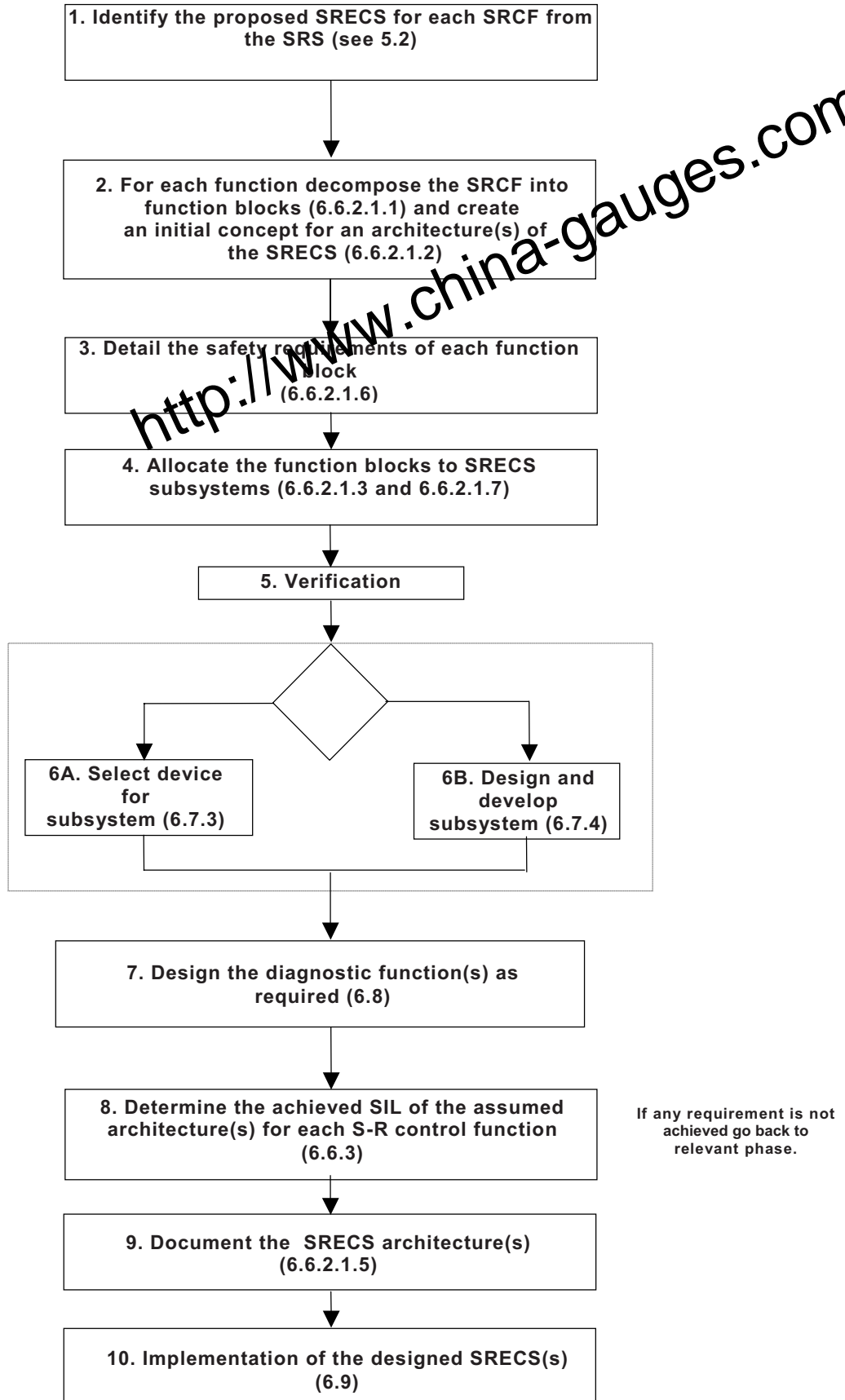


Figure 2 – Workflow of the SRECS design and development process

6.6.2.1.3 Each function block shall be allocated to a subsystem within the architecture of the SRECS. More than one function block may be allocated to one subsystem.

6.6.2.1.4 Each subsystem and the function blocks allocated to it shall be clearly identified.

6.6.2.1.5 The architecture shall be documented describing its subsystems and their interrelationship.

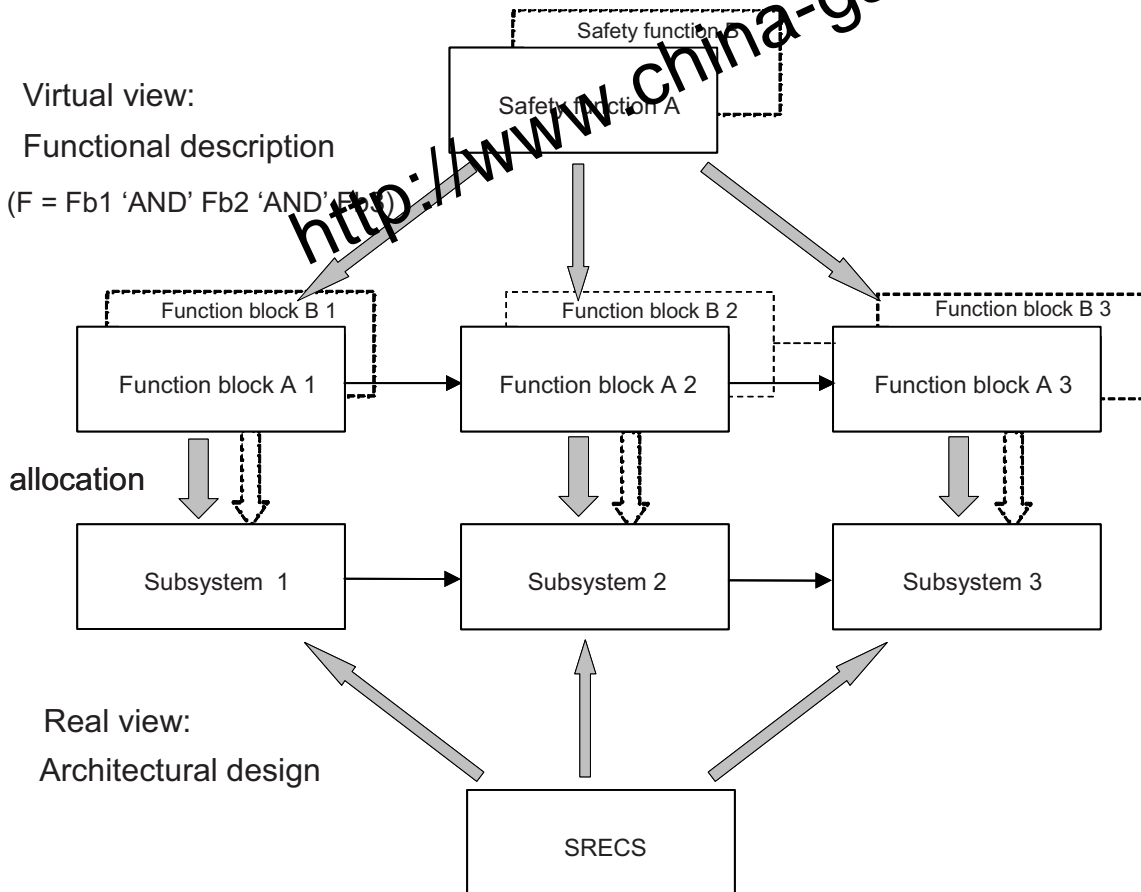


Figure 3 – Allocation of safety requirements of the function blocks to subsystems (see 6.6.2.1.1)

6.6.2.1.6 The safety requirements for each function block shall be as specified in the safety requirements specification of the corresponding SRCF in terms of

- functional requirements (e.g. input information, internal operation (logic) and output of the function block);
- safety integrity requirements.

6.6.2.1.7 The safety requirements for a subsystem shall be those of the function block(s) allocated to it. If more than one function block is allocated to a subsystem, then the highest integrity requirement applies (see 6.6.3). These requirements shall be documented as the subsystem safety requirements specification.

6.6.3 Requirements for the estimation of the safety integrity achieved by a SRECS

6.6.3.1 General

The SIL that can be achieved by the SRECS shall be considered separately for each SRCF to be performed by the SRECS.

The SIL that can be achieved by the SRECS shall be determined from the probability of dangerous random hardware failure, architectural constraints, and systematic safety integrity of the subsystems that comprise the SRECS. \square_{A1} The SIL that can be achieved by the SRECS is less than or equal to the lowest SILCLs of any of the subsystems that comprise the SRECS. \square_{A1}

6.6.3.2 Hardware safety integrity

6.6.3.2.1 The probability of dangerous failure of each SRCF due to dangerous random hardware failures shall be equal to or less than the target failure value as specified in the safety requirements specification.

NOTE The target values associated with SILs are given in Table 3.

6.6.3.2.2 The probability of dangerous failure of each SRCF due to dangerous random hardware failures shall be estimated taking into account:

a) the architecture of the SRECS as it relates to each SRCF under consideration;

NOTE This involves deciding which failure modes of the subsystems are in a series configuration (i.e. any failure causes failure of the relevant SRCF to be carried out) and which are in a parallel (redundant) configuration (i.e. co-incident failures are necessary for the relevant SRCF to fail).

b) the estimated rate of failure of each subsystem to perform its allocated function block(s) in any modes which would cause a dangerous failure of the SRECS.

6.6.3.2.3 The estimation of the probability of dangerous failure shall be based on the probability of dangerous random hardware failure of each relevant subsystem as derived using the information required in 6.7.2.2 including, where appropriate 6.7.2.2 (k), for digital data communication processes between subsystems. The probability of dangerous random hardware failure of the SRECS is the sum of the probabilities of dangerous random hardware failure of all subsystems involved in the performance of the SRCF and shall include, where appropriate, the probability of dangerous transmission errors for digital data communication processes:

$$PFH_D = PFH_{D1} + \dots + PFH_{Dn} + P_{TE}$$

NOTE 1 This approach is based on the definition of a function block which states that a failure of any function block will result in a failure of the SRCF (see 3.2.16).

NOTE 2 Interconnections other than digital data communication are considered to be a part of the subsystems.

6.6.3.3 Architectural constraints

The SIL achieved by the SRECS according to the architectural constraints is less than or equal to the lowest SILCL of any subsystem (see 6.7.6) involved in the performance of the SRCF.

NOTE For example, a SRECS comprises two series connected subsystems (subsystem 1 and subsystem 2) where the SFF and fault tolerance of each subsystem are assumed to be as shown in Table 4. The estimated PFH_D for the SRECS is 8×10^{-8} , which corresponds to SIL 3. However, according to Table 5 the architectural constraint of subsystem 2 limits the SIL that can be achieved by the SRECS to SIL 2.

Table 4 – Characteristics of subsystems 1 and 2 used in this example (see Note above)

Subsystem	Hardware fault tolerance	SFF	SIL claim limit according to architectural constraints (see Table 5)
1	1	95 %	SIL 3
2	1	80 %	SIL 2

A1 Text deleted **A1**

6.7 Realisation of subsystems

6.7.1 Objective

The objective is to realise a subsystem that fulfils all safety requirements of the allocated function blocks (see Figure 1). Two approaches are considered.

- selection of a device that is sufficient to fulfil the requirements for that subsystem, i.e. it shall fulfil the safety requirements specification of each of its allocated function blocks and the requirements of this standard; or
- design and development of a subsystem by combining function block elements and specifying how they are arranged and how they interact.

6.7.2 General requirements for subsystem realisation

A1 **6.7.2.1** The subsystem shall be realised by either selection (see 6.7.3) or design (see 6.7.4) in accordance with its safety requirements specification (see 6.6.2.1.7), taking into account all the requirements of 6.2. Subsystem(s) incorporating complex components shall comply with IEC 61508-2 and IEC 61508-3 as appropriate for the required SIL and the design shall use Route 1H (see IEC 61508-2:2010, 7.4.4.2).

EXCEPTION: Where a subsystem design includes a complex component as a subsystem element, 6.7.4.2.3 is applicable.

NOTE In this standard, it is presumed that the design of complex programmable electronic subsystems or subsystem elements conforms to the relevant requirements of IEC 61508 and uses Route 1_H (see IEC 61508-2:2010, 7.4.4.2). It is considered that Route 2_H (see IEC 61508-2:2010, 7.4.4.3) is not suitable for general machinery. Therefore, this standard does not deal with Route 2_H. This standard provides a methodology for the use, rather than development, of such subsystems and subsystem elements as part of a SRECS. **A1**

6.7.2.2 The following information shall be available for each subsystem:

- a) a functional specification of those functions and interfaces of the subsystem which can be used by SRCFs;
- b) the estimated rates of failure (due to random hardware failures) declared in any modes which could cause a dangerous failure of the SRECS;

NOTE 1 **A1** For electromechanical subsystems, the probability of failure should be estimated taking into account the number of operating cycles declared by the manufacturer and the duty cycle (see 5.2.3). This information should be based upon a B10 value (see IEC 61649) under the operating conditions stated by the manufacturer. See for example **A2** IEC 60947-4-1:2009+A1:2012 **A2**, Annex K. **A1**

c) constraints on the subsystem for

- the environment and operating conditions which should be observed in order to maintain the validity of the estimated rates of failure due to random hardware failures; and

¹ To be published.

- the lifetime of the subsystem which should not be exceeded in order to maintain the validity of the estimated rates of failure due to random hardware failures;
- d) any test and/or maintenance requirements;
- e) the diagnostic coverage and the diagnostic test interval (when required, see Note 2)
NOTE 2 Item e) relates to diagnostic functions that are external to the subsystem. This information is only required when credit is claimed in the reliability model of the SRECS for the action of the diagnostic functions performed in the subsystem.
- f) any additional information (e.g. repair times) which is necessary to allow the derivation of a mean time to restoration (MTTR) following detection of a fault by the diagnostics.
NOTE 3 Items b) to f) are needed to allow the probability of failure per hour of the SRCF to be estimated.
- g) the SILCL due to architectural constraints (see 6.7.6) or:
 - all information which is necessary to enable the derivation of the safe failure fraction (SFF) of the subsystem as applied in the SRECS; and
NOTE 4 The required information is the possible failure modes of the subsystem. Based on the failure modes of the subsystem, it can be decided whether the subsystem failure causes a safe or a dangerous failure of the SRECS.
NOTE 5 For details on estimation of the SFF see 6.7.7.
 - the hardware fault tolerance of the subsystem;
- h) any limits on the application of the subsystem which should be observed in order to avoid systematic failures;
- i) the highest safety integrity level that can be claimed for a SRCF which uses the subsystem on the basis of:
 - measures and techniques used to prevent systematic faults being introduced during the design and implementation of the hardware and software of the subsystem;
 - the design features that make the subsystem tolerant against systematic faults.
NOTE 6 Items h) and i) are needed to determine the highest safety integrity level that can be claimed for a SRCF according to the architectural constraints. Also, these items can be used to provide a link (see Tables 4 and 5) to the category requirements of ISO 13849-1 in terms of both fault detection and hardware fault tolerance.
- j) any information which is required to identify the hardware and software configuration of the subsystem in order to enable the configuration management of a SRECS in accordance with 6.11.3.2;
- k) the probability of dangerous transmission errors for digital data communication processes, where applicable.

6.7.3 Requirements for selection of existing (pre-designed) subsystems

6.7.3.1 Where a supplier provides a subsystem for a specific SRCF referenced in the safety requirements specification, such a pre-designed subsystem may be selected instead of a custom design providing that it satisfies the safety requirements specification for the subsystem, 6.4.3 and 6.7.3.2 or 6.7.3.3.

6.7.3.2 Subsystems incorporating complex components shall comply with IEC 61508-2 and IEC 61508-3 as appropriate for the required SIL A_1 and the design shall use Route 1_H (see IEC 61508-2:2010, 7.4.4.2). A_1

EXCEPTION: Where a subsystem design includes a complex component as a subsystem element, 6.7.4.2.3 is applicable.

A_1 NOTE In this standard, it is presumed that the design of complex programmable electronic subsystems or subsystem elements conforms to the relevant requirements of IEC 61508 and uses Route 1_H (see IEC 61508-2:2010, 7.4.4.2). It is considered that Route 2_H (see IEC 61508-2:2010, 7.4.4.3) is not suitable for general machinery. Therefore, this standard does not deal with Route 2_H. This standard provides a methodology for the use, rather than development, of such subsystems and subsystem elements as part of a SRECS. A_1

6.7.3.3 Subsystems comprising low complexity components only shall comply with 6.7.4.4.1, 6.7.6.2, 6.7.6.3, 6.7.7, 6.7.8 and 6.8 of this standard.

6.7.4 Design and development of subsystems

6.7.4.1 Objectives

6.7.4.1.1 The first objective is to design a subsystem that fulfils the safety requirements of the allocated function block(s).

6.7.4.1.2 The second objective is to create an architecture in terms of subsystem elements that work together in combination to fulfil the functional and safety integrity requirements of all function blocks allocated to the subsystem.

6.7.4.2 General requirements

6.7.4.2.1 The subsystem shall be designed in accordance with its safety requirements specification.

6.7.4.2.2 The subsystem shall be such as to meet all of the requirements a) to c) as follows:

- a) the requirements for hardware safety integrity comprising:
 - the architectural constraints on hardware safety integrity (see 6.7.6), and
 - the requirements for the probability of dangerous random hardware failures (see 6.7.8);
- b) the requirements for systematic safety integrity comprising:
 - the requirements for the avoidance of failures (see 6.7.9.1), and the requirements for the control of systematic faults (see 6.7.9.2), or
 - evidence that the equipment is 'proven-in-use'. In this case, the subsystem shall fulfil the relevant requirements of IEC 61508-2 (see IEC 61508-2, **A1** 7.4.10 **A1**).
- c) the requirements for subsystem behaviour on detection of a fault (fault reaction)(see 6.3).

A1 6.7.4.2.3 Where the design of a subsystem incorporates a complex component (as a subsystem element) which satisfies all relevant requirements of IEC 61508-2 and IEC 61508-3 in relation to the SILCL and uses Route 1_H (see IEC 61508-2:2010, 7.4.4.2), it can be considered as a low complexity component in the context of a subsystem design since its relevant failure modes, behaviour on detection of a fault, rate of failure, and other safety-related information are known. Such components shall only be used in accordance with their specification and the relevant information for use provided by their supplier.

NOTE In this standard, it is presumed that the design of complex programmable electronic subsystems or subsystem elements conforms to the relevant requirements of IEC 61508 and uses Route 1_H (see IEC 61508-2:2010, 7.4.4.2). It is considered that Route 2_H (see IEC 61508-2:2010, 7.4.4.3) is not suitable for general machinery. Therefore, this standard does not deal with Route 2_H. This standard provides a methodology for the use, rather than development, of such subsystems and subsystem elements as part of a SRECS. **A1**

6.7.4.3 Subsystem design and development process

The subsystem design and development shall follow a clearly defined process that shall take into account all aspects covered by the process shown in Figure 4.

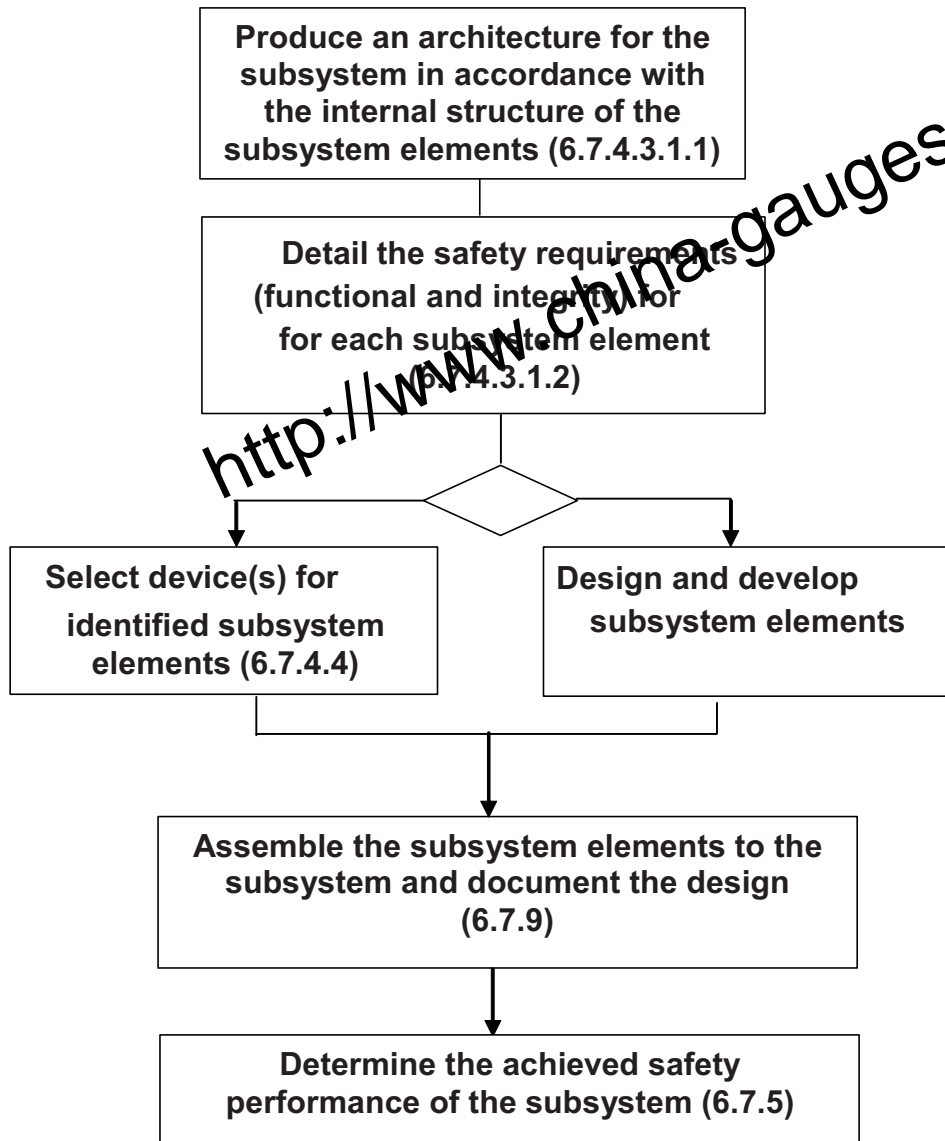


Figure 4 – Workflow for subsystem design and development (see box 6B of Figure 2)

6.7.4.3.1 Subsystem architecture design

6.7.4.3.1.1 During subsystem architecture design, the decomposition process should lead to a structure of function block elements that fully represent the functional requirements of the function block. This process should be applied down to that level which permits the functional requirements determined for each function block element to be allocated to subsystem elements (see example in Figure 5).

NOTE The workflow of the design process is shown in Figure 4.

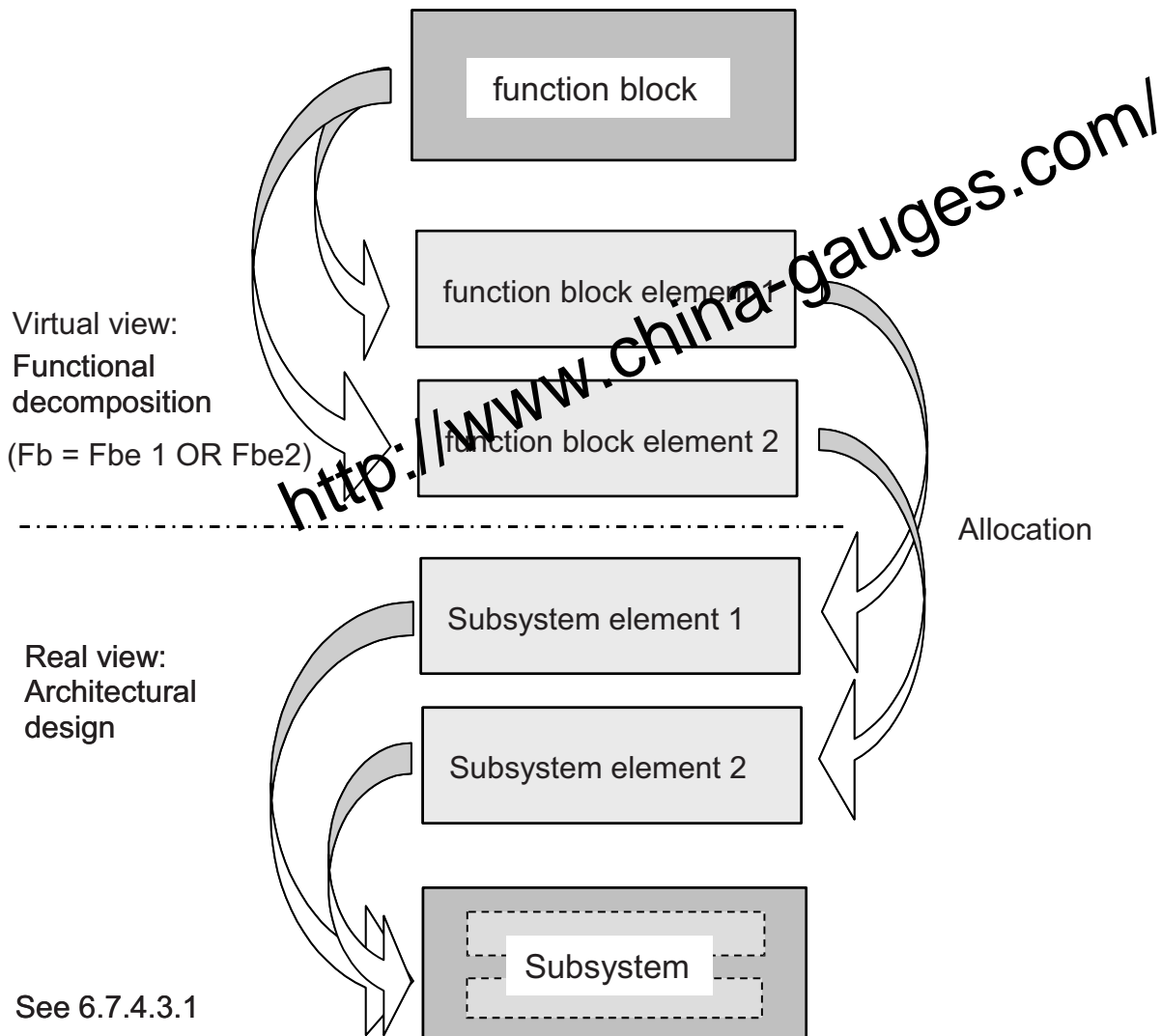


Figure 5 – Decomposition of a function block into redundant function block elements and their associated subsystem elements

6.7.4.3.1.2 The subsystem architecture shall be documented in terms of its elements and their interrelationships. Where necessary this shall also include information relating to function block elements that are allocated to subsystem elements.

6.7.4.4 Requirements for the selection and design of subsystem elements

6.7.4.4.1 Subsystem elements shall be suitable for their intended use and shall conform to relevant international standards where such exist.

6.7.4.4.2 The following information shall be available for each subsystem element:

- a) a functional specification of the subsystem element;
- b) specification of the interfaces of the subsystem element (e.g. electrical characteristics);

- c) each failure mode and its probability of occurrence, and, where relevant (e.g. complex components used in accordance with 6.7.4.2.3), the diagnostic coverage and probability of dangerous failure.

NOTE ^{A1} For electromechanical subsystems, the probability of failure should be estimated taking into account the number of operating cycles declared by the manufacturer and the duty cycle (see 5.2.3). This information should be based upon a B10 value (see IEC 61649) under the operating conditions stated by the manufacturer. See for example ^{A2} IEC 60947-4-1:2009+A1:2012 ^{A2}, Annex K. ^{A1}

- d) constraints on the subsystem element for
- the environment and operating conditions which should be observed in order to maintain the validity of the information given in item (c); and
 - the lifetime of the subsystem element which should not be exceeded in order to maintain the validity of the information given in item (c);
- e) any periodic proof test and/or maintenance requirements;
- f) features that can contribute to diagnostics (e.g. mechanically linked contacts);
- g) any additional information (e.g. repair times) which is necessary to allow the derivation of a mean time to restoration (MTTR) following detection of a fault by the diagnostics;
- h) any limits on the application of the subsystem element which should be observed in order to avoid systematic failures;
- i) hardware fault tolerance.

6.7.5 Determination of the safety performance of the subsystem

The safety performance of a subsystem is characterized by the SILCL determined by its architectural constraints (6.7.6), its SILCL due to systematic integrity (6.7.9) and its probability of dangerous random hardware failure (6.7.8).

NOTE 1 The SILCL of a subsystem sets a limit for the maximum safety integrity level that can be claimed for a safety-related control function using this subsystem.

NOTE 2 Information about all three aspects is necessary to determine the SIL achieved by the safety-related control system implementing the allocated SRCF.

6.7.6 Architectural constraints on hardware safety integrity of subsystems

6.7.6.1 In the context of hardware safety integrity, the highest safety integrity level that can be claimed for a SRCF is limited by the hardware fault tolerances and safe failure fractions of the subsystems that carry out that SRCF. Table 5 specifies the highest safety integrity level that can be claimed for a SRCF that uses a subsystem taking into account the hardware fault tolerance and safe failure fraction of that subsystem. The architectural constraints given in Table 5 shall be applied to each subsystem. With respect to these architectural constraints:

- a) a hardware fault tolerance of N means that $N+1$ faults could cause a loss of the SRCF. In determining the hardware fault tolerance, no account is taken of other measures that can control the effects of faults such as diagnostics; and
- b) where one fault directly leads to the occurrence of one or more subsequent faults, these shall be considered as a single fault;
- c) in determining hardware fault tolerance, certain faults may be excluded, provided that the likelihood of them occurring is very low in relation to the safety integrity requirements of the subsystem. Any such fault exclusions shall be justified and documented (see also 6.7.7).

6.7.6.2 The architectural constraints of Table 5 shall apply to each subsystem implementing a function block of an SRCF.

6.7.6.3 A subsystem that comprises only a single subsystem element shall satisfy the requirements of Table 5. In particular, for such a subsystem that has a hardware fault tolerance of zero (i.e. $N = 0$) then a SFF of greater than 99 % shall be achieved by a SRECS diagnostic function(s).

NOTE This requirement is necessary to ensure an appropriate form of the architectural constraints is applied to subsystems that comprise only a single subsystem element in order to justify a SILCL of SIL 3.

Table 5 – Architectural constraints on subsystems, maximum SIL that can be claimed for a SRCF using this subsystem

Safe failure fraction	Hardware fault tolerance (see Note 1)		
	0	1	2
< 60 %	Not allowed (for exceptions see Note 3)	SIL1	SIL2
60 % – < 90 %	SIL1	SIL2	SIL3
90 % – < 99 %	SIL2	SIL3	SIL3 (see Note 2)
≥ 99 %	SIL3	SIL3 (see Note 2)	SIL3 (see Note 2)

NOTE 1 A hardware fault tolerance of N means that $N+1$ faults could cause a loss of the safety-related control function.

NOTE 2 A SIL 4 claim limit is not considered in this standard. For SIL 4 see IEC 61508-1.

NOTE 3 See 6.7.6.4 or for subsystems where fault exclusions have been applied to faults that could lead to a dangerous failure, see 6.7.7.

6.7.6.4 Electromechanical subsystems, which have a safe failure fraction of less than 60 % and zero hardware fault tolerance, that use well-tried components (see Note) in accordance with ISO 13849-1:2006 Category 1 $\text{PL} = \text{c}$ shall be considered to achieve a SILCL of SIL1.

NOTE A well-tried component for a safety-related application is a component which has been:

- widely used in the past with successful results in similar applications, or
- made and verified using principles which demonstrate its suitability and reliability for safety-related applications.

A_1 Text deleted A_1

6.7.7 Estimation of safe failure fraction (SFF)

6.7.7.1 The SFF shall be estimated where it is required to determine the SILCL due to architectural constraints.

6.7.7.2 To estimate the SFF, an analysis (e.g. fault tree analysis, failure mode and effects analysis) of each subsystem shall be performed to determine all relevant faults and their corresponding failure modes. Whether a failure is a safe or a dangerous failure depends on the SRECS and the intended safety-related control functions, including fault reaction function. The probability of each failure mode shall be determined based on the probability of the associated fault(s) taking into account the intended use and may be derived from sources such as:

- dependable failure rate data collected from field experience by the manufacturer and relevant to the intended use;
- component failure data from a recognised industry source A_1 Text deleted A_1 and relevant to the intended use;

A_1 Text deleted A_1

- A_1 c) A_1 failure rate data derived from the results of testing and analysis.

EXCEPTION: For a subsystem with a hardware fault tolerance of zero and where fault exclusions have been applied to faults that could lead to a dangerous failure, then the SILCL due to architectural constraints of that subsystem is constrained to a maximum of SIL 2.

[A1] NOTE 1 Information of the failure mode ratios for electrical/electronic component can be found in several sources including:

- MIL-HDBK 217F(Notice 2) Reliability Prediction of Electronic Equipment (28-02-95), Parts Stress Analysis
- MIL-HDBK 217F(Notice 2) Reliability Prediction of Electronic Equipment (28-02-95), Appendix A, Parts Count Reliability Prediction
- SN 29500 Part 7, Failure Rates of Components, Expected Values for Relays, April 1992
- SN 29500 Part 11, Failure Rates of Components, Expected Values for Contactors, August 1990
- The documents in the SN 29500 series are publicly available and can be obtained from:
 - Siemens AG, CT SR SI
Otto-Hahn-Ring 6
D-81739 München:
- UTE C 80-810 RDF 2000: Reliability data handbook – A universal model for reliability prediction of electronic components, PCBs and equipment
- Failure mode/mechanism distributions FMD-91, RAC 1991.

NOTE 2 It is recommended to use failure rate data and failure mode ratio data provided by manufacturers.

NOTE 3 Some component standards have relevant data (e.g. Annex K of **[A2]** IEC 60947-4-1:2009+A1:2012 **[A2]**).

NOTE 4 Where a detailed analysis of each failure mode is not practically possible, a division of failures into 50 % safe, 50 % dangerous is generally accepted.

NOTE 5 Lists of faults to be considered for mechanical, pneumatic, hydraulic and electrical technologies are given in Annexes A, B, C and D of **[A2]** ISO 13849-2:2012 **[A2]**, **[A1]**.

6.7.7.3 The use of fault exclusions shall be justified (e.g. by analysis) and documented.

[AC2] NOTE It is permissible to exclude faults in accordance with **[A2]** 4.4 and D.2 of ISO 13849-2:2012 **[A2]**, **[AC2]**.

6.7.8 Requirements for the probability of dangerous random hardware failures of subsystems

6.7.8.1 General requirements

6.7.8.1.1 The probability of dangerous random hardware failure shall be equal to or less than the target failure value as specified in the subsystem safety requirements specification (see 6.6.2.1.7).

6.7.8.1.2 The probability of dangerous failure of each subsystem due to random hardware failures to perform the allocated function blocks shall be estimated taking into account:

- a) the architecture of the subsystem as it relates to the allocated function blocks under consideration;

NOTE 1 This involves deciding whether there is hardware fault tolerance or not.

- b) the rate of failure of each subsystem element in any modes which would cause a dangerous failure of the subsystem but which are detected by diagnostic tests (see 6.3);
- c) the rate of failure of each subsystem element in any modes which would cause a dangerous failure of the subsystem which are undetected by the diagnostic tests (see 6.3);

- d) the susceptibility of the subsystem to common cause failures which would cause a dangerous failure of the subsystem (see Notes 2 and 3);

NOTE 2 Where comparison of redundant components is used for fault detection, failure of the fault detection means can occur when the redundant components fail at the same time in the same mode. This can occur due to a common cause referred to as a common cause failure (CCF) that is expressed as a beta (β) factor. A simplified approach to estimate the susceptibility to common cause failures is given in 6.7.8.3. For further guidance on quantifying the effect of hardware-related common cause failures, see also IEC 61508-6:2010 (A2), Annex D.

- e) the diagnostic coverage of the diagnostic tests (see 3.2.38) and the associated diagnostic test interval;
- f) the intervals at which proof tests are undertaken to reveal dangerous faults which are undetected by diagnostic tests and/or the useful lifetime (A1) of the subsystem element(s) which should not be exceeded in order to maintain the validity of the information given in items b) and c);
- g) the repair times for detected faults where the subsystem is designed for online repair.

NOTE 3 The maximum repair time will constitute one part of the time to restoration (see IECV 191-10-05), including also the time taken to detect a fault and any time period during which repair is not possible (see IEC 61508-6, Annex B for an example of how the mean time to restoration can be used to calculate the probability of failure). For situations where the repair can only be carried out during a specific period of time, while the machine is shut down and in a safe state, it is particularly important that full account is taken of the time period when no repair can be carried out, especially when this is relatively large.

NOTE 4 A simplified approach for the estimation of the probability of dangerous random hardware failure of subsystems is given in 6.7.8.2. Other methods are available and the most appropriate method will depend on the circumstances. Available methods include:

- a) fault tree analysis (see B.6.6.5 of IEC 61508-7:2010 (A2) and IEC 61025);
- (A1) b) Markov models (see B.6.6.6 of IEC 61508-7:2010 (A2) and IEC 61165);
- c) reliability block diagrams (see B.6.6.7 of IEC 61508-7:2010 (A2) and IEC 61078). (A1)

NOTE 5 Failures due to common cause effects and data communication processes can result from effects other than actual failures of hardware components (e.g. electromagnetic interference, software errors, etc.). See 6.7.9.

6.7.8.1.3 For subsystems or subsystem elements where the probability of failure is given in relation to a number of operating cycles, these values shall be transformed into time-related values by using the specified duty cycle for the relevant SRCFs (see 5.2.3).

6.7.8.1.4 The diagnostic test interval of any subsystem having a hardware fault tolerance of more than zero shall be such as to enable the subsystem to meet the requirement for the probability of random hardware failure (see 6.3.1).

NOTE This diagnostic test interval should be such that a fault is detected before the occurrence of a subsequent fault that may lead to dangerous failure of the subsystem and exceeds the target failure measure.

6.7.8.1.5 The diagnostic test interval of any subsystem having a hardware fault tolerance of zero shall be such that the requirements of 6.3.2 are fulfilled.

(A1) Text deleted (A1)

6.7.8.2 Simplified approach for the estimation of probability of dangerous random hardware failures of subsystems

6.7.8.2.1 General

This subclause describes a simplified approach to the estimation of probability of dangerous random hardware failures for a number of basic subsystem architectures and gives formulae that can be used for subsystems assembled from either low complexity subsystem elements or complex subsystem elements. The formulae are in themselves a simplification of reliability analysis theory and are intended to provide estimates that are biased towards the safe direction. The precondition for the validity for all formulae given in this subclause is that $1 \gg \lambda \times T_1$, where T_1 is the smaller of the proof test interval or the lifetime, and the subsystem is operating in the "high demand or continuous mode" (see 3.2.27). See also 6.8.6.

NOTE 1 The results that are obtained represent a limitation upon probability of dangerous random hardware failures of subsystems and where this is unacceptable, it is possible to apply more accurate modelling techniques (see 6.7.8.1.1).

AC2 NOTE 2 For equations (A) to (D) given in 6.7.8.2 constant and sufficiently low ($1 \gg \lambda \times T$) failure rates (λ) of the subsystem elements are assumed (this means that the mean time to dangerous failure has to be much greater than the proof test interval or the lifetime of the subsystem). Therefore, the following basic equations can be used:

- $\lambda = 1/\text{MTTF}$, where MTTF is expressed in hours.

For electromechanical devices the failure rate is determined using the B_{10} value and the number of operating cycles C (expressed as the number of operating cycles per hour) of the application as specified (see 5.2.3).

- $\lambda = 0,1 \times C/B_{10}$ **AC2**

A2 Throughout this standard λ is expressed as the constant failure rate with respect to 1 hour. **A2**

NOTE 3 Terms used are as follows:

- $\lambda = \lambda_S + \lambda_D$; where λ_S is the rate of safe failure and λ_D is the rate of dangerous failure.
- $PFH_D = \lambda_D$ **A2** *Text deleted* **A2**; average probability of dangerous failure within one hour.
- T_2 : diagnostic test interval.
- T_1 : proof test interval or lifetime whichever is the smaller.

6.7.8.2.2 Basic subsystem architecture A: zero fault tolerance without a diagnostic function

In this architecture, any dangerous failure of a subsystem element causes a failure of the SRCF. For architecture A, the probability of dangerous failure of the subsystem is the sum of the probabilities of dangerous failure of all subsystems elements:

$$\lambda_{DssA} = \lambda_{De1} + \dots + \lambda_{Den} \quad (A)$$

$$\text{A2} \text{ } PFH_{DssA} = \lambda_{DssA} \text{A2}$$

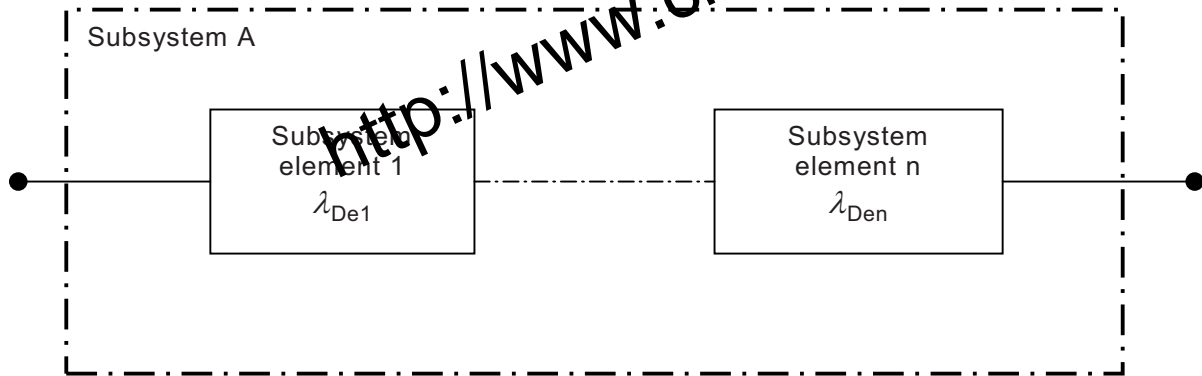


Figure 6 – Subsystem A logical representation

NOTE Figure 6 is a logical representation of the subsystem A architecture and should not be interpreted as its physical implementation.

6.7.8.2.3 Basic subsystem architecture B: single fault tolerance without a diagnostic function

This architecture is such that a single failure of any subsystem element does not cause a loss of the SRCF. Thus, there would have to be a dangerous failure in more than one element before failure of the SRCF can occur. For architecture B, the probability of dangerous failure of the subsystem is:

$$\lambda_{DssB} = (1 - \beta)^2 \times \lambda_{De1} \times \lambda_{De2} \times T_1 + \beta \times (\lambda_{De1} + \lambda_{De2}) / 2 \quad (B)$$

$$\text{A2} \text{ } PFH_{DssB} = \lambda_{DssB} \text{A2}$$

where

T_1 is the proof test interval or lifetime whichever is the smaller.

β is the susceptibility to common cause failures.

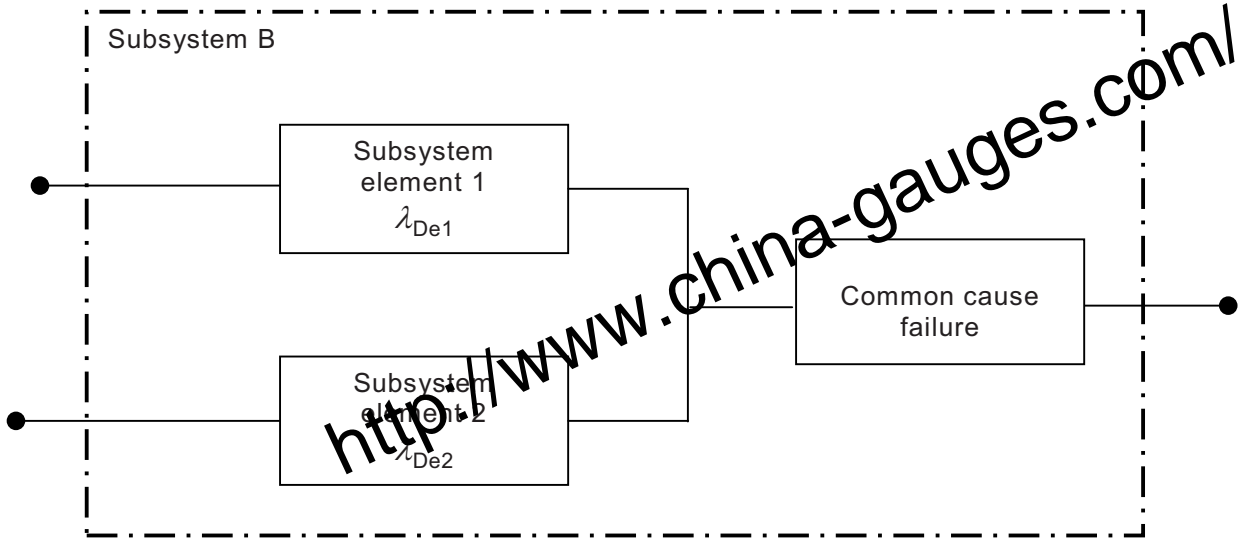


Figure 7 – Subsystem B logical representation

NOTE Figure 7 is a logical representation of the subsystem B architecture and should not be interpreted as its physical implementation.

6.7.8.2.4 Basic subsystem architecture C: zero fault tolerance with a diagnostic function

Any undetected dangerous fault of the subsystem element leads to a dangerous failure of the SRCF. Where a fault of a subsystem element is detected, the diagnostic function(s) initiates a fault reaction function (see 6.3.2). For architecture C, the probability of dangerous failure of the subsystem is:

$$\lambda_{DssC} = \lambda_{De1} (1 - DC_1) + \dots + \lambda_{Den} (1 - DC_n) \quad (C)$$

$$\boxed{A_2} PFH_{DssC} = \lambda_{DssC} \boxed{A_2}$$

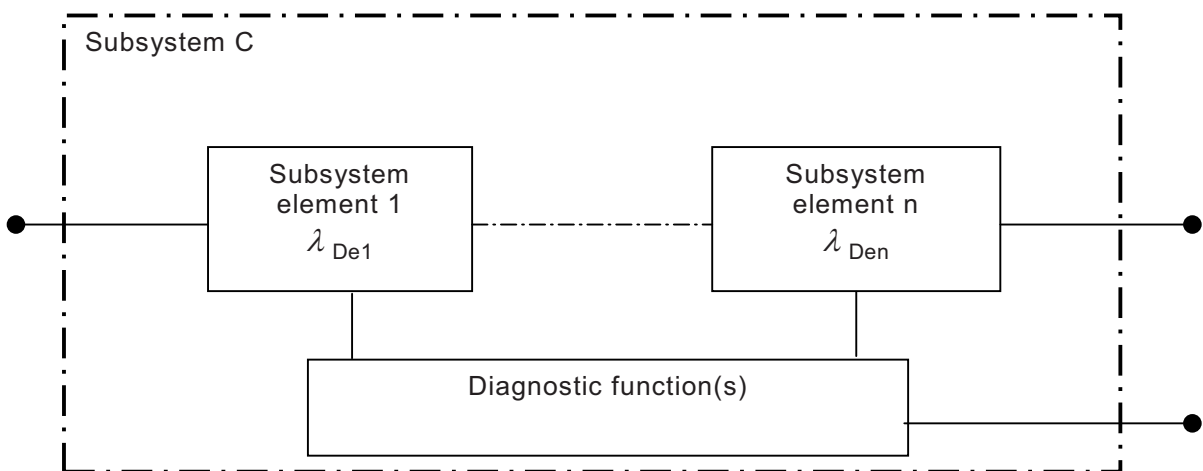


Figure 8 – Subsystem C logical representation

NOTE Figure 8 is a logical representation of the subsystem C architecture and should not be interpreted as its physical implementation. The diagnostic function shown may be carried out by

- the subsystem which requires diagnostics; or
- other subsystems of the SRECS; or
- subsystems not involved in the performance of the safety-related control function.

6.7.8.2.5 Basic subsystem architecture D: single fault tolerance with a diagnostic function(s)

This architecture is such that a single failure of any subsystem element does not cause a loss of the SRCF, where

T_2 is the diagnostic test interval;

T_1 is the proof test interval or T_2 whichever is the smaller.

β is the susceptibility to common cause failures; $\lambda_D = \lambda_{DD} + \lambda_{DU}$; where λ_{DD} is the rate of detectable dangerous failures and λ_{DU} is the rate of undetectable dangerous failure.

$$\lambda_{DD} = \lambda_D \times DC$$

$$\lambda_{DU} = \lambda_D \times (1 - DC)$$

For subsystem elements of different design:

λ_{De1} is the dangerous failure rate of subsystem element 1;

DC_1 is the diagnostic coverage of subsystem element 1;

λ_{De2} is the dangerous failure rate of subsystem element 2;

DC_2 is the diagnostic coverage of subsystem element 2.

$$\lambda_{DssD} = (1 - \beta)^2 \{ [\lambda_{De1} \times \lambda_{De2} \times (DC_1 + DC_2)] \times T_2/2 + [\lambda_{De1} \times \lambda_{De2} \times (2 - DC_1 - DC_2)] \times T_1/2 \} + \beta \times (\lambda_{De1} + \lambda_{De2})/2 \quad (D.1)$$

$$\boxed{A2} \text{ } PFH_{DssD} = \lambda_{DssD} \boxed{A2}$$

For subsystem elements of the same design:

λ_{De} is the dangerous failure rate of subsystem element 1 or 2;

DC is the diagnostic coverage of subsystem element 1 or 2.

$$\lambda_{DssD} = (1 - \beta)^2 \{ [\lambda_{De}^2 \times 2 \times DC] \times T_2/2 + [\lambda_{De}^2 \times (1 - DC)] \times T_1 \} + \beta \times \lambda_{De} \quad (D.2)$$

$$\boxed{A2} \text{ } PFH_{DssD} = \lambda_{DssD} \boxed{A2}$$

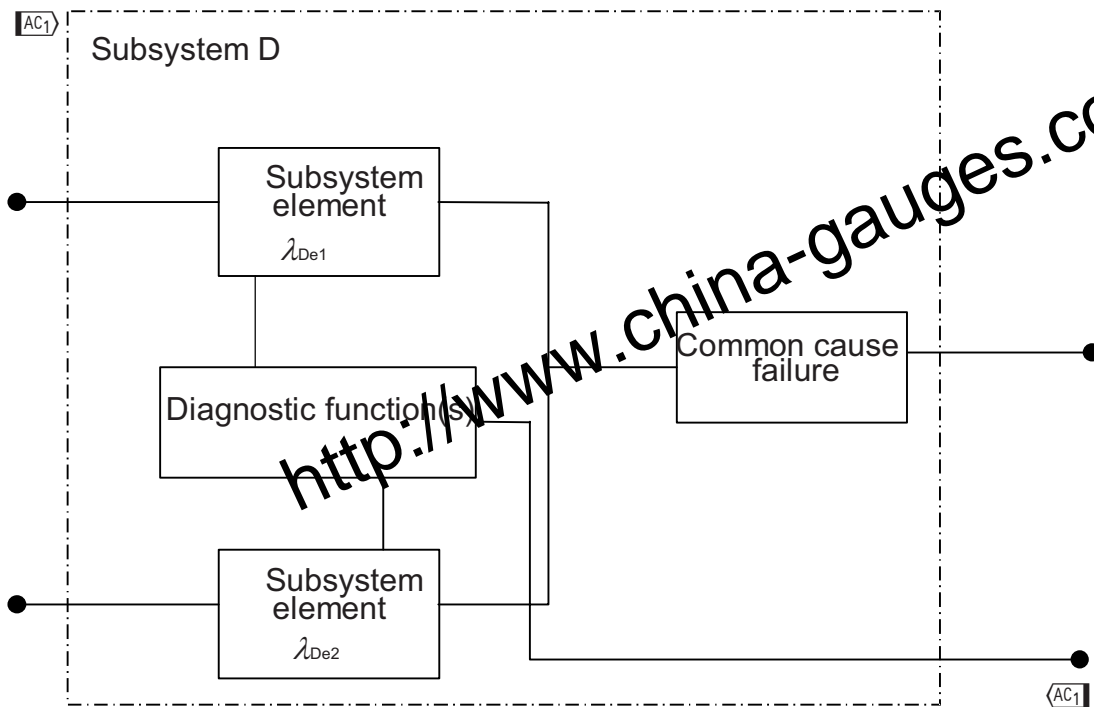


Figure 9 – Subsystem D logical representation

NOTE 1 Figure 9 is a logical representation of the subsystem D architecture and should not be interpreted as its physical implementation. The diagnostic function(s) shown may be carried out by

- the subsystem which requires diagnostics; or
- other subsystems of the SRECS; or
- subsystems not involved in the performance of the safety-related control function.

NOTE 2 The fault reaction for this subsystem is assumed to be termination of the relevant operation as required in 6.3.1. When an online repair is incorporated in the design where the fault reaction is to report the fault but not to terminate the relevant operation, a new PFH_D for the subsystem after occurrence of a first fault should be determined for the remaining architecture.

6.7.8.3 Simplified approach to estimation of contribution of common cause failure (CCF)

6.7.8.3.1 Knowledge of the susceptibility of a subsystem to CCF is required to contribute to the estimation of the probability of dangerous random hardware failure of a subsystem (see 6.7.8.1).

6.7.8.3.2 Where a redundant architecture is used to achieve the required probability of dangerous random hardware failure of a subsystem and a CCF(s) can remove the effect of that redundancy, the probability of dangerous random hardware failure based on the probability of occurrence of the common cause shall be added to the probability of dangerous random hardware failure of a subsystem based on the use of redundancy.

6.7.8.3.3 The probability of occurrence of the CCF will usually be dependent upon a combination of technology, architecture, application and environment. The use of Annex F will be effective in avoiding many types of CCF.

6.7.8.3.4 Annex F contains a scoring table and an associated methodology that can be used to estimate the effectiveness of measures applied in the design of a subsystem to limit susceptibility to CCF.

6.7.9 Requirements for systematic safety integrity of subsystems

The SILCL due to systematic safety integrity of a subsystem is up to SIL 3 when the requirements in 6.7.9.1 and 6.7.9.2 are fulfilled.

A1 Note deleted **A1**

6.7.9.1 Requirements for the avoidance of systematic failures

6.7.9.1.1 The following measures shall be applied:

- a) proper selection, combination, arrangements, assembly and installation of components, including cabling, wiring and any interconnections: apply manufacturer's application notes and use of good engineering practice;
- b) use of the subsystem and subsystem elements within the manufacturer's specification and installation instructions;
- c) compatibility: use components with compatible operating characteristics;
- d) withstanding specified environmental conditions: design the subsystem so that it is capable of working in all expected environments and in any foreseeable adverse conditions, for example temperature, humidity, vibration and electromagnetic interference (EMI) (see **A2** ISO 13849-2:2012 **A2**, Clause D.1);
- e) use of components that are in accordance with an appropriate standard and have their failure modes well-defined: to reduce the risk of undetected faults by the use of components with specific characteristics;
- f) use of suitable materials and adequate manufacturing: selection of material, manufacturing methods and treatment in relation to, for example stress, durability, elasticity, friction, wear, corrosion, temperature, conductivity, dielectric strength;
- g) correct dimensioning and shaping: consider the effects of, for example, stress, strain, fatigue, temperature, surface roughness, manufacturing tolerances.

6.7.9.1.2 In addition, one or more of the following measures shall be applied taking into account the complexity of the subsystem:

- a) hardware design review (e.g. by inspection or walk-through): to reveal by reviews and/or analysis discrepancies between the specification and implementation;

NOTE 1 In order to reveal discrepancies between the specification and implementation, any points of doubt or potential weak points concerning the realisation, the implementation and the use of the product are documented so they may be resolved; taking into account that on an inspection procedure the author is passive and the inspector is active whilst on a walk-through procedure the author is active and the inspector is passive.

- b) computer-aided design tools capable of simulation or analysis: perform the design procedure systematically and include appropriate automatic construction elements that are already available and tested;

NOTE 2 The integrity of these tools can be demonstrated by specific testing, or by an extensive history of satisfactory use, or by independent verification of their output for the particular subsystem that is being designed. See 6.11.3.4.

- c) simulation: perform a systematic simulation of a subsystem design in terms of both the functional performance and the correct dimensioning of their components.

NOTE 3 The function of the subsystem can be simulated on a computer via a software behavioural model (see 6.11.3.4) where individual components of the circuit each have their own simulated behaviour, and the response of the subsystem in which they are connected is examined by looking at the marginal data of each component.

6.7.9.2 Requirements for the control of systematic failures

6.7.9.2.1 The following measures shall be applied:

- a) measures to control the effects of insulation breakdown, voltage variations and interruptions, overvoltage and undervoltage: subsystem behaviour in response to insulation breakdown, voltage variations and interruptions, overvoltage and undervoltage conditions shall be pre-determined so that the subsystem can achieve or maintain a safe state of the SRECS;

NOTE 1 See also relevant requirements of IEC 60204-1. In particular:

- overvoltage should be detected early enough so that all outputs can be switched to a safe condition by the power-down routine or a switch-over to a second power unit; and/or
- the control circuit voltage should be monitored and a power-down initiated, or a switch-over to a second power unit, if it is not within its specified range; and/or
- overvoltage or undervoltage should be detected early enough so that the internal state can be saved in non-volatile memory (if necessary), so that all outputs can be set to a safe condition by the power-down routine or a switch-over to a second power unit.

- b) measures to control or avoid the effects of the physical environment (for example, temperature, humidity, water, vibration, dust, corrosive substances, electromagnetic interference and its effects): subsystem behaviour in response to the effects of the physical environment shall be pre-determined so that the SRECS can achieve or maintain a safe state of the machine. See also IEC 60204-1;

- c) measures to control or avoid the effects of temperature increase or decrease, if temperature variations can occur: the subsystem should be designed so that, for example, over-temperature can be detected before it begins to operate outside specification.

NOTE 2 Further information can be found in [A2](#) IEC 61508-7:2010 [A2](#), Clause A.10.

6.7.9.2.2 In addition, the following measures, as appropriate, shall be applied for the control of systematic failures:

- failure detection by on-line monitoring;
- tests by comparison of redundant hardware;
- diverse hardware;
- operation in the positive mode (e.g. a limit switch is pushed when a guard is opened);
- oriented mode of failure;
- over-dimensioning by a suitable factor, where the manufacturer can demonstrate that derating will improve reliability.

NOTE 1 Where over-dimensioning is appropriate, an over-dimensioning factor of at least 1,5 should be used.

NOTE 2 Further information can be found in [A2](#) ISO 13849-2:2012, Table D.2 [A2](#).

6.7.10 Subsystem assembly

The subsystem elements shall be combined to form the subsystem in accordance with 6.7.4.3.1.2 and the detailed design documented.

6.8 Realisation of diagnostic functions

6.8.1 Each subsystem shall be provided with associated diagnostic functions that are necessary to fulfil the requirements for architectural constraints (6.7.6) and the probability of dangerous random hardware failures (6.7.8).

6.8.2 The diagnostic functions are considered as separate functions that may have a different structure than the SRCF and may be performed by

- the same subsystem which requires diagnostics; or
- other subsystems of the SRECS; or
- subsystems of the SRECS not performing the SRCF.

NOTE See also Note 3 of 6.6.2.1.

6.8.3 Diagnostic functions shall satisfy the following that are applicable to their associated SRCFs:

- requirements for the avoidance of systematic failures (see 6.7.9.1); and
- requirements for the control of systematic failures (see 6.7.9.2).

6.8.4 The probability of failure of the SRECS diagnostic function(s) shall be taken into account when estimating the probability of dangerous failure of the SRCF.

NOTE 1 See also Note 3 of 6.6.2.1.

NOTE 2 Timing constraints applicable to the testing of the subsystem performing a diagnostic function may differ from those applicable to SRCFs and, in general, the test interval should meet requirements applicable to a subsystem with a hardware fault tolerance of 1.

NOTE 3 Failure of a diagnostic function(s) should be detected and an appropriate reaction should be taken to ensure that the contribution of the diagnostic function to the safety integrity of the SRCF is maintained. The failure of a diagnostic function(s) may be detected by on-line testing, cross-checking by redundant hardware, etc.

6.8.5 A clear description of the SRECS diagnostic function(s), their failure detection/reaction, and an analysis of their contribution towards the safety integrity of the associated SRCFs shall be provided.

6.8.6 To apply the simplified approach for the estimation of probability of dangerous random hardware failures of subsystems (6.7.8.2), the following shall apply:

- where a SRECS diagnostic function(s) is necessary to achieve the required probability of dangerous random hardware failure and the subsystem has a hardware fault tolerance of zero, then the fault detection and specified fault reaction shall be performed before the hazard due to this fault can occur; and
- SRECS diagnostic function(s) shall as a minimum be implemented so that the probability of random hardware failure and the systematic safety integrity are the same as those specified for the corresponding SRCF(s); or

NOTE 1 Architectural constraints on hardware safety integrity need not apply to the realisation of diagnostic function(s).

- where the probability of dangerous random hardware failure is of an order of magnitude greater than that specified for the SRCF, then a test shall be performed to determine whether diagnostic function(s) or diagnosing device(s) remain operational. It is assumed that such a test of the diagnostic function(s) or diagnosing device(s) be carried out at a minimum of 10 times during the interval between proof tests applied to the subsystem.

NOTE 2 A test of the diagnostic function(s) should as far as practicable cover 100 % of those parts implementing the diagnostic function(s).

NOTE 3 Where a diagnostic function is implemented by the logic solver of the SRECS it can be unnecessary to perform a separate test of the diagnostic function as its failure can be revealed as a failure of the SRCF.

NOTE 4 A test can be performed by either external means (e.g. test equipment) or internal dynamic checks (e.g., embedded within the logic solver) of the SRECS.

6.9 Hardware implementation of the SRECS

The SRECS shall be implemented in accordance with the documented SRECS design.

6.9.1 SRECS interconnection

6.9.1.1 The SRECS shall be interconnected so as to satisfy appropriate parts of the SRECS safety requirements specification and those requirements relevant to conductors, cabling and wiring practices in IEC 60204.

6.9.1.2 Measures for avoiding and controlling failures of interconnecting conductors and cables shall be realised in accordance with 6.4.1 and 6.4.2.

6.10 Software safety requirements specification

6.10.1 General

Where software is to be used in any part of a SRECS implementing a safety-related control function(s), a software safety requirements specification shall be developed and documented.

6.10.2 Requirements

6.10.2.1 A software safety requirements specification shall be developed for each subsystem on the basis of the SRECS specification and architecture.

6.10.2.2 The specification of the requirements for software safety for each subsystem shall be derived from (1) the specified safety requirements of the SRCF, (2) the requirements resulting from the SRECS architecture and (3) any requirements of the functional safety plan (see 4.2). This information shall be made available to the application software developer.

6.10.2.3 The specification of the requirements for application software safety shall be sufficiently detailed to allow the design and implementation of the SRECS to achieve the required safety integrity, and to allow verification.

6.10.2.4 The application software developer shall review the information in the specification to ensure that the requirements are adequately specified. In particular, the software developer shall conform to this standard by including the following:

- SRCFs;
- configuration or architecture of the system;
- capacity and response time performance;
- equipment and operator interfaces;
- all relevant modes of operation of the machine as specified in the safety requirements specification;
- diagnostic tests of external devices (e.g. sensors and final elements).

6.10.2.5 The specified requirements for software safety shall be expressed and structured so that they are:

- clear, verifiable, testable, maintainable and operable, commensurate with the safety integrity level;
- traceable back to the specification of the safety requirements of the SRECS;
- free of ambiguous terminology and descriptions.

6.10.2.6 The software safety requirements specification shall express the required properties of each subsystem by providing information allowing proper equipment selection. The requirements for the following software-based SRCFs shall be specified:

- the logic (i.e. the functionality) of all function blocks assigned to each subsystem;
- input and output interfaces assigned for each function block;
- format and value range of input and output data and their relation to function blocks;
- relevant data to describe any limits of each function block, for example maximum response time, limit values for plausibility checks;
- diagnostic functions of other devices within the SRECS (e.g. sensors and final elements) to be implemented by that subsystem;
- functions that enable the machine to achieve or maintain a safe state;
- functions related to the detection, annunciation and handling of faults;
- functions related to the periodic testing of SRCFs on-line and off-line;
- functions that prevent unauthorized modification of the SRECS;
- interfaces to non SRCFs; and
- capacity and response time performance.

NOTE Interfaces include both off-line and online programming facilities.

6.10.2.7 Where appropriate, semi-formal methods such as logic, function-block, or sequence diagrams shall be used in the documentation.

NOTE Guidance on software documentation is given in IEC 61506,  ISO/IEC/IEEE 26512:2011 and ISO/IEC/IEEE 26511:2011. 

6.11 Software design and development

6.11.1 Embedded software design and development

Embedded software incorporated into subsystems shall comply with IEC 61508-3 as appropriate for the required SIL.

NOTE 1 See also 6.7.3.2

NOTE 2 Annex C is provided to assist in the design and development of embedded software used to implement SRCFs within a SRECS.

6.11.2 Software based parameterization

6.11.2.1 Software based parameterization of safety-related parameters shall be considered as a safety-related aspect of SRECS design that is described in the software safety requirements specification (see 6.10). Parameterization shall be carried out using a dedicated tool provided by the supplier of the SRECS or the related subsystem(s). This tool shall have its own identification (name, version, etc.). The parameterization tool shall prevent unauthorized modification, for example by using a password.

6.11.2.2 The integrity of all data used for parameterization shall be maintained. This shall be achieved by applying measures to

- control the range of valid inputs;
- control data corruption before transmission;
- control the effects of errors from the parameter transmission process;
- control the effects of incomplete parameter transmission; and
- control the effects of faults and failures of hardware and software of the tool used for parameterization.

6.11.2.3 The tool used for parameterization shall fulfil the following requirements:

- all relevant requirements for a subsystem according to this standard to ensure correct parameterization; or
- a special procedure shall be used for setting the safety-related parameters. This procedure shall include confirmation of input parameters to the SRECS by either:
 - retransmitting the modified parameters to the parameterization tool; or
 - other means to confirm the integrity of the parameters;

and subsequent confirmation (e.g. by a suitably skilled person and an automatic check by a parameterization tool);

NOTE This is of particular importance where parameterization is carried out using a device not specifically intended for this purpose (e.g. personal computer or equivalent).

- the software modules used for encoding/decoding within the transmission/retransmission process and software modules used for visualization of the safety-related parameters to the user shall as a minimum use diversity in function(s) to avoid systematic failures.

6.11.2.4 Documentation of software based parameterization shall indicate data used (e.g. pre-defined parameter sets) and information necessary to identify the parameters associated with the SRECS, the person(s) carrying out the parameterization together with other relevant information such as date of parameterization.

6.11.2.5 The following verification activities shall be applied for software based parameterization:

- verification of the correct setting for each safety-related parameter (minimum, maximum and representative values);
- verification that the safety-related parameters are checked for plausibility, for example by detection of invalid values, etc.;
- verification that unauthorized modification of safety-related parameters is prevented;

- verification that the data/signals for parameterization are generated and processed in such a way that faults cannot lead to a loss of SRCF(s).

NOTE This is of particular importance where the parameterization is carried out using a device not specifically intended for this purpose (e.g. personal computer or equivalent),

6.11.3 Application software design and development

NOTE This subclause is based on IEC 61508-3.

6.11.3.1 General requirements

6.11.3.1.1 The requirements of IEC 61508-3 apply to Full Variability Languages (FVL). The following requirements shall be applied to applications software based upon Limited Variability Languages (LVL).

6.11.3.1.2 The outcome of the activities performed during the application software development shall be verified at appropriate stages.

6.11.3.1.3 The design method and application language chosen to satisfy the required SIL of the SRCF shall possess features relevant for the application that facilitate:

- a) abstraction, modularity and other features which control complexity; wherever possible, the software shall be based on well-proven logic functions which may include user library functions and well-defined rules for linking logic functions;
- b) expression of
 - functionality, ideally as a logical description or as algorithmic functions;
 - information flow between modular elements;
 - sequencing and time related requirements;
 - timing constraints;
 - data structures and their properties, including data types, validity of data ranges;
- c) comprehension by developers and others who need to understand the design, both from a functional understanding of the application and from a knowledge of the constraints of the SRECS technology;
- d) verification and validation, including structural testing (white box) of the application software, functional testing (black box) of the integrated application program and interface testing (grey box) of the interaction with the SRECS and its application specific hardware configuration;
- e) safe modification.

6.11.3.1.4 Testing shall be the main verification method used for the application software. Test planning shall address the following:

- the policy for verification of the integration of software and hardware;
- test cases and test results;
- types of tests to be performed;
- test equipment including tools, support software and configuration description;
- test criteria on which the completion of the test shall be judged;

- physical location(s) (e.g. factory or site);
- dependence on external functionality;
- the amount of test cases necessary; and
- completeness with respect to the related functions or requirements.

6.11.3.1.5 Where the application software is to implement both non-safety and safety-related control functions, then all of the application software shall be treated as safety-related, unless adequate independence between the functions can be demonstrated in the design.

6.11.3.1.6 The design shall include data integrity checks and reasonableness checks at the application layer (e.g. checks in communication links, bounds checking on sensor inputs, bounds checking on data parameters).

6.11.3.1.7 The application software design shall include self-monitoring of control flow and data flow unless such functions are included in the embedded software. On failure detection, appropriate actions shall be performed to achieve or maintain a safe state.

6.11.3.1.8 Where previously developed software library functions are to be used as part of the design, their suitability in satisfying the specification of requirements for software safety shall be justified. Suitability shall be based upon evidence of satisfactory operation in similar applications that have been demonstrated to have similar functionality, or shall be subject to the same verification and validation procedures as would be expected for any newly developed safety-related software. Constraints from the previous software environment (for example operating system and compiler dependencies) shall be evaluated.

6.11.3.1.9 Any modifications or changes to application software shall be subject to an impact analysis that identifies all software modules affected and the necessary re-verification activities to confirm that the software safety requirements specification is still satisfied.

6.11.3.2 Software configuration management

6.11.3.2.1 The functional safety plan shall define the strategy for the development, integration, verification and validation of the software.

6.11.3.2.2 Software configuration management shall:

- ensure that all necessary operations have been carried out to demonstrate that the required software safety integrity has been achieved;
- maintain accurately and with unique identification all documents related to configuration items that are necessary to maintain the integrity of the SRECS. Configuration items shall include at least the following:
 - safety analysis and requirements;
 - software specification and design documents;
 - software source code modules;
 - test plans and results;
 - pre-existing software modules and packages which are to be incorporated into the SRECS;
 - all tools and development environments that are used to create or test, or carry out any action on the application software;

- apply change-control procedures to:
 - prevent unauthorized modifications;
 - document modification requests;
 - analyze the impact of proposed modifications, and to approve or reject the request(s);
 - document the details of, and the authorization for, all approved modifications;
 - document the software configuration at appropriate points in the software development;
- document the following information to permit a subsequent audit: release status, the justification for and approval of all modifications, and the details of the modification;
- formally document the release of the application software. Master copies of the software and all associated documentation shall be kept to permit maintenance and modification throughout the operational lifetime of the released software.

6.11.3.3 Requirements for software architecture

NOTE 1 The software architecture defines the major components and subsystems of system and application software, how they are interconnected, and how the required attributes should be achieved. Examples of application software modules include application functions that are replicated throughout the machine, machine input/output, override and inhibit components, data validity checking and range checks, etc.

NOTE 2 The software architecture is also affected by the underlying architecture of the subsystem provided by the supplier.

6.11.3.3.1 The software architecture design shall be based on the required SRECS safety specification within the constraints of the system architecture of the SRECS and the subsystem design.

6.11.3.3.2 The software architecture design shall:

- a) provide a comprehensive description of the internal structure and of the operation of the SRECS and of its components (see Note);
- b) include the specification of all identified software components, and the description of connection and interactions between identified components (software and hardware);
- c) include the internal design and architecture of all identified components that are not black boxes;
- d) identify the software modules included in the SRECS but not used in any mode of safety-related operation.

NOTE It is of particular importance that the architecture documentation be up-to-date and complete with respect to the SRECS.

6.11.3.3.3 A set of techniques and measures necessary during design of the application software to satisfy the specification shall be described and justified. These techniques and measures shall aim at ensuring the predictability of the behaviour of the SRECS and shall be consistent with any constraints identified in the SRECS documentation.

6.11.3.3.4 Measures used for maintaining the integrity of all data shall be described and justified. Such data may include machine input-output data, communications data, operation interface data, maintenance data and internal database data.

6.11.3.4 Requirements for support tools, user manual and application languages

6.11.3.4.1 A suitable set of tools, including configuration management, simulation, and test harness tools shall be selected. The availability of suitable tools (not necessarily those used during initial system development) to supply the relevant services over the lifetime of the SRECS shall be considered. The suitability of the tools shall be explained and documented.

NOTE The selection of development tools depends on the nature of the software development activities, the embedded software and the software architecture. Verification and validation tools such as code analyzers, and simulators may be needed.

6.11.3.4.2 Wherever necessary a sub-set of the application programming language shall be defined.

6.11.3.4.3 Application software shall be designed taking into account constraints and known weaknesses included in the SRECS and subsystem(s) user manuals.

6.11.3.4.4 The application language selected shall either:

- be processed using a translator/compiler which shall be assessed to establish its fitness for purpose;
 - be completely and unambiguously defined or restricted to unambiguously defined features;
 - correspond to the characteristics of the application;
- NOTE An application's characteristics refer, for example to any performance constraints.
- contain features that facilitate the detection of programming mistakes; and
 - support features that match the design method;

or, the deficiencies of the language used shall be documented in the software architecture design description and the fitness for purpose of the language shall be explained including additional measures necessary to address the identified shortcomings of the language.

6.11.3.4.5 The procedures for use of the application language shall specify good configuration practice, proscribe unsafe generic software features (for example, undefined language features, unstructured designs, etc), identify checks that can be used to detect errors in the configuration and specify procedures for documentation of the application program. As a minimum, the following information shall be contained in the application program documentation:

- a) legal entity (for example company, author(s), etc);
- b) description;
- c) traceability to application functional requirements;
- d) traceability to standard library function;
- e) inputs and outputs; and
- f) configuration management.

6.11.3.5 Requirements for application software design

6.11.3.5.1 The following information shall be available prior to the start of detailed application software design:

- the software safety requirements specification;
- the description of the software architecture design including identification of the application logic and fault tolerant functionality, a list of input and output data, the generic software modules and support tools to be used and the procedures for configuring the application software with the available materials to provide the application functionality for the defined I/O; and
- the plan for validating the software safety.

6.11.3.5.2 The application software shall be produced in a structured way to achieve:

- modularity of application functionality and of I/O control data;
- testability of functionality (including fault tolerant features) and of internal structure;
- the capacity for safe modification through provision of adequate traceability and explanation of application functions and associated constraints.

6.11.3.5.3 For each major component/subsystem in the description of the application software architecture design (see 6.11.3.5.1), refinement of the design shall be based on:

- functions which are used in a recurring fashion throughout the design;
- mapping of the input/output information of application software modules;
- realisation of the application functions from the generic software functions and I/O mapping.

6.11.3.5.4 The design of each application software module and the structural tests to be applied to each application software module shall be specified.

6.11.3.5.5 Appropriate software and SRECS integration tests shall be specified to ensure that the application program satisfies the specified requirements for application software safety. The following shall be considered:

- the division of the application software into manageable integration sets;
- test cases;
- types of tests to be performed;
- test environment, tools, configuration and programs;
- test criteria on which the completion of the test shall be judged; and
- procedures for corrective action on failure of test.

6.11.3.6 Requirements for application code development

6.11.3.6.1 The application software shall:

- be readable, understandable and testable;
- satisfy the relevant design principles;
- satisfy the relevant requirements specified during safety planning.

6.11.3.6.2 The application software shall be reviewed to ensure conformance to the specified design, the coding rules and the requirements of safety planning.

NOTE Application software review includes such techniques as software inspections or walk-throughs, code analysis or mathematical proof. These techniques should be used in conjunction with testing and/or simulation to provide assurance that the application software satisfies its associated specification.

6.11.3.7 Requirements for application module testing

NOTE Testing that the application software correctly satisfies its test specification is a verification activity. It is the combination of code review and structural testing that provides assurance that an application software module satisfies its associated specification, i.e. it is verified.

6.11.3.7.1 The configuration of each input and output point shall be checked through review, testing, or simulation to confirm that the I/O data is mapped to the correct application logic.

6.11.3.7.2 Each software module shall be checked through a process of review, simulation and testing to determine that the intended function is correctly executed and unintended functions are not executed.

6.11.3.7.3 The tests shall be suitable for the specific module being tested and shall:

- ensure each branch of any application software modules is exercised;
- ensure boundary data is exercised;
- ensure sequences are correctly implemented, including relevant synchronisation conditions.

6.11.3.7.4 The results of the application software module testing shall be documented.

6.11.3.7.5 Where software has already been assessed or when a significant amount of positive operating experience is available, the amount of testing may be reduced.

6.11.3.8 Requirements for application software integration testing

NOTE Testing that the software is correctly integrated is a verification activity.

6.11.3.8.1 The application software tests shall verify that all application software modules and components/subsystems interact correctly with each other and with the underlying embedded software to perform their intended function and do not perform unintended functions that could jeopardize any safety function.

6.11.3.8.2 The results of application software integration testing shall be documented, stating:

- the test results; and
- whether the objectives of the test criteria have been met.

6.11.3.8.3 If there is a failure, the reasons for the failure and corrective action taken shall be included in the test results documentation.

6.11.3.8.4 During application software integration, any modification or change to the software shall be subject to a safety impact analysis that shall determine:

- all software modules impacted; and
- all necessary re-verification and re-design activities.

6.12 Safety-related electrical control system integration and testing

NOTE SRECS integration is usually carried out prior to installation but, in some cases, the SRECS integration cannot be carried out until after installation (for example, when the application software development is not finalized until after installation).

6.12.1 General requirements

6.12.1.1 The SRECS shall be integrated according to the specified SRECS design. As part of the integration of all subsystems and subsystem elements into the SRECS, the SRECS shall be tested according to the specified integration tests. These tests shall verify that all modules interact correctly to perform their intended function and not perform unintended functions.

6.12.1.2 The integration of safety-related application software into the SRECS shall include tests that are specified during the design and development phase to ensure the compatibility of the application software with the hardware and embedded software platform such that the functional and safety performance requirements are satisfied.

NOTE 1 This does not imply testing of all input combinations. Testing all equivalence classes (see **A1**) B.5.2 **A1** and C.5.7 of **A2**) IEC 61508-7:2010 **A2**) can suffice. Static analysis, dynamic analysis or failure analysis can reduce the number of test cases to an acceptable level. Use of structured design or semi-formal methods can facilitate testing and verification.

NOTE 2 Use of structured design or semi-formal methods can permit a reduced depth and number of test cases.

NOTE 3 Statistical evidence may also be used to permit a reduced depth and number of test cases.

6.12.1.3 Appropriate documentation of the integration testing of the SRECS shall be produced, stating the test results and whether the objectives and criteria specified during the design and development phase have been met. If there is a failure, the reasons for the failure shall be documented, corrective action taken and re-testing carried out.

6.12.1.4 During the integration and testing, any modification or change to the SRECS shall be subject to an impact analysis that shall identify all components affected and additional verification.

6.12.1.5 During SRECS integration testing, the following shall be documented:

- a) the version of the test specification used;
- b) the criteria for acceptance of the integration tests;
- c) the version of the SRECS being tested;
- d) the tools and equipment used along with calibration data;
- e) the results of each test;
- f) all discrepancies between expected and actual results;
- g) the analysis made and the decisions taken on whether to continue the test or issue a change request, in the case where discrepancies occur.

6.12.2 Tests to determine systematic safety integrity during SRECS integration

6.12.2.1 Testing to reveal faults and to avoid failures during integration of the application software and hardware shall be applied. During the tests, reviews shall be carried out to see whether the specified characteristics of the SRECS have been achieved.

6.12.2.2 The following tests shall be applied:

- a) functional tests where data that adequately characterizes the operation are applied to the SRECS. The outputs shall be observed and their response is compared with that given by the specification. Deviations from the specification and indications of an incomplete specification shall be documented; and
- b) dynamic tests to verify the dynamic behaviour under realistic functioning conditions and reveal failures to meet the SRECS functional specification, and to assess utility and robustness of the SRECS.

NOTE The functions of a system or program are executed in a specified environment with specified test data that has been derived systematically from the SRECS SRS according to established criteria. This exposes the behaviour of the SRECS and permits a comparison with the specification. The aim is to determine whether the SRECS and/or its subsystems carries out correctly all the functions required by the specification. The technique of forming equivalence classes is an example of the criteria for black-box test data. The input data space is subdivided into specific input value ranges (equivalence classes) with the aid of the specification. Test cases are then formed from the:

- data from permissible ranges;
- data from inadmissible ranges;
- data from the range limits;
- extreme values;
- and combinations of the above classes.

Other criteria can be effective in order to select test cases in the various test activities (module test, integration test and system test).

6.13 SRECS installation

6.13.1 Objective

The objectives of the requirements of this subclause are for installation of a SRECS to ensure that it is suitable for its intended use and that it is ready for validation.

6.13.2 Requirements

6.13.2.1 A SRECS shall be installed in accordance with the functional safety plan for the final system validation (see item h) of 4.2.1).

6.13.2.2 Appropriate records of the installation of the SRECS shall be produced, stating any test results. If there is a failure, the reasons for the failure shall be recorded.

7 Information for use of the SRECS

7.1 Objective

Information on the SRECS shall be provided to enable the user to develop procedures to ensure that the required functional safety of the SRECS is maintained during use and maintenance of the machine.

7.2 Documentation for installation, use and maintenance

NOTE 1 See also A_2 Clause 6.4 of ISO 12100:2010 A_2 that provides general information that should be considered during drafting of accompanying documents.

NOTE 2 One or more items of the documentation described in this subclause may have been developed in order to satisfy other aspects of this standard.

The documentation shall provide information for installation, use and maintenance of the SRECS. This shall include:

- a) a comprehensive description of the equipment, installation and mounting.
- b) a statement of the intended use of the SRECS and any measures that can be necessary to prevent reasonably foreseeable misuse.
- c) information on the physical environment (e.g. lighting, vibration, noise levels, atmospheric contaminants) where appropriate.
- d) overview (block) diagram(s) where appropriate.
- e) circuit diagram(s).
- f) proof test interval or lifetime.
- g) a description (including interconnection diagrams) of the interaction (if any) between the SRECS function (s) and the machine electrical control system function(s).
- h) a description of the necessary measures to ensure the separation of the SRECS function(s) from the machine electrical control system function(s).
- i) a description of the safeguarding and of the means provided to maintain safety where it is necessary to suspend the SRCF(s) (e.g. for manual programming, program verification).
- j) programming information, where relevant.
- k) description of the maintenance requirements applicable to the SRECS including:
 - 1) a log for recording the maintenance history of the machine;
 - 2) the routine actions which need to be carried out to maintain the functional safety of the SRECS, including routine replacement of components with a pre-defined life;
 - 3) the maintenance procedures to be followed when faults or failures occur in the SRECS, including:
 - procedures for fault diagnosis and repair;
 - procedures for confirming correct operation subsequent to repairs;
 - maintenance recording requirements.
 - 4) the tools necessary for maintenance and re-commissioning, and the procedures for maintaining the tools and equipment.
 - 5) a specification for periodic testing, preventive maintenance and corrective maintenance.

NOTE 3 Periodic tests are those functional tests necessary to confirm correct operation and to detect faults.

NOTE 4 Preventive maintenance are the measures taken to maintain the required performance of the SRECS.

NOTE 5 Corrective maintenance includes the measures taken after the occurrence of specific fault(s) that bring the SRECS back into the as-designed state.

8 Validation of the safety-related electrical control system

NOTE Validation of the SRECS may form a part of the validation activities applied to the overall machine design.

8.1 Objective

This Clause specifies the requirements for the validation process to be applied to the SRECS. This includes inspection and testing of the SRECS to ensure that it achieves the requirements stated in the safety requirements specification.

8.2 General requirements

8.2.1 The validation of the SRECS shall be carried out in accordance with a prepared plan (see 4.2).

NOTE 1 In some cases, the safety validation cannot be completed until after installation (for example when the application software development is not finalized until after installation).

NOTE 2 Validation of a programmable SRECS comprises validation of both hardware and software. The requirements for validation of software are contained in 6.11.3.

8.2.2 Each SRCF specified in the SRECS requirements specification (see 5.2), and all the SRECS operation and maintenance procedures shall be validated by test and/or analysis.

8.2.3 Appropriate documentation of the SRECS safety validation testing shall be produced, which shall state for each SRCF:

- a) the version of the SRECS safety validation plan being used and the version of the SRECS tested;
- b) the SRCF under test (or analysis), along with the specific reference to the requirement specified during the SRECS safety validation planning;
- c) tools and equipment used, along with calibration data;
- d) the results of each test;
- e) discrepancies between expected and actual results.

8.2.4 When discrepancies occur, corrective action and re-testing shall be carried out as necessary and documented.

8.3 Validation of SRECS systematic safety integrity

8.3.1 The following shall be applied:

- a) functional testing to reveal failures during the specification, design and integration phases, and to avoid failures during validation of SRECS software and hardware shall be applied. This shall include verification (e.g., by inspection or test) to assess whether the SRECS is protected against adverse environmental influences and shall be based upon the safety requirements specification;

NOTE 1 See also 6.12.2.1.

- b) interference immunity testing to ensure that the SRECS is able to satisfy 5.2.3. Testing for immunity to electromagnetic interference need not be performed on SRECS subsystems or subsystem elements where adequate immunity of the SRECS for its intended application can be shown by analysis;

NOTE 2 The SRECS should, wherever practicable, be loaded with a typical application program, and all the peripheral lines (all digital, analogue and serial interfaces as well as the bus connections and power supply, etc.) are subjected to standard noise signals. In order to obtain a quantitative statement, it is sensible to approach any limits carefully.

- c) fault insertion testing shall be performed where the required safe failure fraction $\geq 90\%$. These tests shall introduce or simulate faults in the SRECS hardware and the response documented.

8.3.2 In addition, one or more of the following groups of analytical techniques shall be applied taking into account the complexity of the SRECS and the assigned SIL:

a) static and failure analysis;

NOTE 1 This combination of analytical techniques is only considered suitable for SRECS that implement SRCFs with an assigned SIL not exceeding SIL2.

NOTE 2 Further information can be found in IEC 61508-7:2010, B.6.4 and B.6.6.

b) static, dynamic and failure analysis;

NOTE 3 This combination of analytical techniques is not recommended for SRECS that implement SRCFs with an assigned SIL below SIL2.

NOTE 4 Further information can be found in IEC 61508-7:2010, B.6.4, B.6.5 and B.6.6.

c) simulation and failure analysis.

NOTE 5 This combination of analytical techniques is only considered suitable for SRECS that implement SRCFs with an assigned SIL not exceeding SIL2.

NOTE 6 Further information can be found in IEC 61508-7:2010, B.3.6 and B.6.6.

8.3.3 In addition, one or more of the following groups of testing techniques shall be applied taking into account the complexity of the SRECS and the assigned SIL:

a) black-box testing: a test(s) of the dynamic behaviour under real functional conditions to reveal failures to meet the SRECS functional specification, and to assess utility and robustness of the SRECS;

NOTE 1 See also 6.12.2.1.

b) fault insertion (injection) testing shall be performed where the required safe failure fraction <90 %. These tests shall introduce or simulate faults in the SRECS hardware and the results documented;

c) "worst-case" testing shall be performed to assess the extreme (i.e. worst) cases specified by application of the analytical techniques (see 8.3.2);

NOTE 2 The operational capacity of the SRECS and its component dimensioning is tested under worst-case conditions. The environmental conditions are changed to their highest permissible marginal values. The most essential responses of the SRECS are inspected and compared with the safety requirements specification.

d) field experience: the use of field experience from different applications as one of the measures to avoid faults during SRECS validation.

NOTE 3 See also 6.12.2.

9 Modification

9.1 Objective

9.2 This Clause specifies the modification procedure(s) to be applied when modifying the SRECS during design, integration and validation (e.g. during SRECS installation and commissioning).Modification procedure

9.2.1 The request for a modification of the SRECS can arise from, for example:

- safety requirements specification changed;
- conditions of actual use;
- incident/accident experience;
- change of material processed;
- modifications of the machine or of its operating modes.

NOTE Interventions (e.g. adjustment, setting, repairs) on the SRECS made in accordance with the information for use or instruction manual for the SRECS are not considered to be a modification in the context of this Clause.

9.2.2 The reason(s) for the request for a modification of the SRECS shall be documented.

9.2.3 The effect of the requested modification shall be analyzed to establish the effect on the functional safety of the SRECS.

9.2.4 The modification impact analysis and the effect on the functional safety of the SRECS shall be documented.

9.2.5 All accepted modifications that have an effect on the SRECS shall initiate a return to an appropriate design phase for its hardware and/or its software (e.g. specification, design, integration, installation, commissioning, and validation). All subsequent phases shall then be carried out in accordance with the procedures specified for the specific phases in this standard. All relevant documents shall be revised, amended and reissued accordingly.

9.2.6 Based on those revised documents, a complete action plan shall be prepared and documented before carrying out any modification.

9.3 Configuration management procedures

9.3.1 The configuration management procedures shall be implemented in accordance with the functional safety plan (see 4.2.1) taking into account the following:

- a) a plan of each modification process;
- b) a documentation of the decision making process and each SRECS-relevant decision;
- c) a chronological documentation (e.g. a logbook) of the change request procedures including
 - identified hazards which can be affected;
 - description of the change request (hardware and/or software);
 - reason(s) for the change request (see also 9.2.1);
 - decision made (and authorization for each decision);
 - the impact analysis;
 - re-verification (of each phase) and revalidation;
 - all documents affected by the change request activities;
 - all activities which were carried out during the change process and the persons/entities who were responsible for them;
- d) documentation of the following information to permit a subsequent audit:
 - configuration status;
 - release status;
 - the justification for and approval of all modifications;
 - the details of the modification.

9.3.2 The procedures for an appropriate change-control-process should consider the requirements of

- a) procedures for defining a unique baseline of each version of the SRECS;
- b) definition of all configuration items of a baseline. This shall include at least

- 1) safety requirements analysis and specification;
- 2) relevant design documents;
- 3) hardware and/or software modules;
- 4) test plans and results;
- 5) verification and validation reports;
- 6) pre-existing software components which are to be incorporated into the SRECS;
- 7) tools and development environments which are used for create and test;
- 8) accurately maintaining with unique identification of all configuration items which are necessary to maintain the integrity of the SRECS;
- 9) change control procedures to:
 - prevent unauthorized modifications,
 - document change requests,
 - analyse the impact of a proposed change request and approve or reject the request,
 - document the details of and the authorization for all approved modifications,
 - establish a configuration baseline at appropriate points in the hardware or software development and to document the (partial) integration testing which justifies the baseline,
 - guarantee the composition of and the building of all hardware or software baselines (including the rebuilding of earlier baselines);
- 10) an effect analysis, which should assess the impact of each change request. This analysis shall include also an appropriate hazard analysis and shall take into account all other modification activities of a SRECS;
- 11) returning to an appropriate design phase for the hardware and/or software (e.g. specification, design, integration, installation, commissioning and validation) of the SRECS for all accepted modifications that have an impact on the SRECS. All subsequent phases shall then be carried out in accordance with this standard;
- 12) carrying out of all necessary operations to demonstrate that the required safety integrity has been reached;
- 13) authorization to carry out the required change request activity shall be dependent on the results of the impact analysis.

9.3.3 The documentation of the change control process shall contain at least

- a) a plan of each modification process;
- b) a documentation of each of the above mentioned organizational requirements and procedures;
- c) a documentation of the decision making process and each SRECS-relevant decision made;
- d) a chronological documentation (logbook) of the change request procedures including
 - identified hazards which may be affected;
 - description of the change request (hardware and/or software);
 - reason(s) for the change request (see also 9.2.1);
 - decision made (and authorization for each decision);
 - the impact analysis;
 - reverification (of each phase) and revalidation;

- all documents affected by the change request activities;
 - all activities which were carried out during the change process and the persons/entities who were responsible for them;
- e) documentation of the following information to permit a subsequent audit:
- configuration status;
 - release status;
 - the justification for and approval of all modifications;
 - the details of the modification.

10 Documentation

10.1 The documentation shall:

- be accurate and concise;
- be easy to understand by those persons having to make use of it;
- suit the purpose for which it is intended;
- be accessible and maintainable.

10.2 The designer of the SRECS should distinguish between the documentation that is relevant to the user and that which is relevant to its design and construction.

10.3 The documents shall have titles or names indicating the scope of the contents.

10.4 The documents shall have a revision index (version numbers) to make it possible to identify different versions of the document.

NOTE See also IEC 82045-1: 2001 for further information on methods that can be used for the management of documentation.

10.5 Table 8 summarizes the information and documentation to be available, where appropriate.

Table 8 – Information and documentation of a SRECS

Information required	Subclause
Functional safety plan	4.2.1
Specification of requirements for SRCFs	5.2
Functional safety requirements specification for SRCFs	5.2.3
Safety integrity requirements specification for SRCFs	5.2.4
SRECS design	6.2.5
Structured design process	6.6.1.2
SRECS design documentation	6.6.1.8
Structure of function blocks	6.6.2.1.1
SRECS architecture	6.6.2.1.5
Subsystem safety requirements specification	6.6.2.1.7
Subsystem realisation	6.7.2.2
Subsystem architecture (elements & their interrelationships)	6.7.4.3.1.2
Fault exclusions claimed when estimating fault tolerance/SFF	6.7.6.1c)/6.7.7.3
Subsystem assembly	6.7.10
Software safety requirements specification	6.10.1

Information required	Subclause
Software based parameterization	6.11.2.4
Software configuration management items	6.11.3.2.2
Suitability of software development tools	6.11.3.4.1
Documentation of the application program	6.11.3.4.5
Results of application software module testing	6.11.3.4
Results of application software integration testing	6.11.3.8.2
Documentation of SRECS integration testing	6.12.1.3
Documentation of SRECS installation	6.13.2.2
Documentation for installation, use and maintenance	7.2
Documentation of SRECS validation testing	8.2.4
Documentation for SRECS configuration management	9.3.1

<http://www.china-gauges.com/>

Annex A (informative)

SIL assignment

A.1 General

This informative Annex provides one example of a qualitative approach for risk estimation and SIL assignment that can be applied to SRCFs for machines. Examples of other techniques that may be used for SIL assignment are given in IEC 61508-5 and will be outlined in a proposed future IEC TC 44 Technical Specification.

A1) NOTE A1 The methodology described in this annex uses qualitative estimation of risk and is intended to be generally applied for the assignment of a SIL(s) to SRCF(s) of machines. The risk parameters (see Figure A.2) used whilst applying this methodology to particular machines and their specific hazards should be subject to agreement with those involved to ensure that the SRECS can provide adequate risk reduction.

A1) Note deleted A1

For each specific hazard, the safety integrity requirements should be determined separately for the safety-related control function(s) to be performed by the SRECS (see 5.2.4.2).

Figure A.1 is an example of a practical way of carrying out a risk assessment at a specific hazard leading to estimation of a SIL requirement for a SRECS function. This methodology should be performed for each risk that is to be reduced by a safety-related control function that is to be implemented by a SRECS. Figure A.1 should be used in conjunction with the guidance information in this Annex.

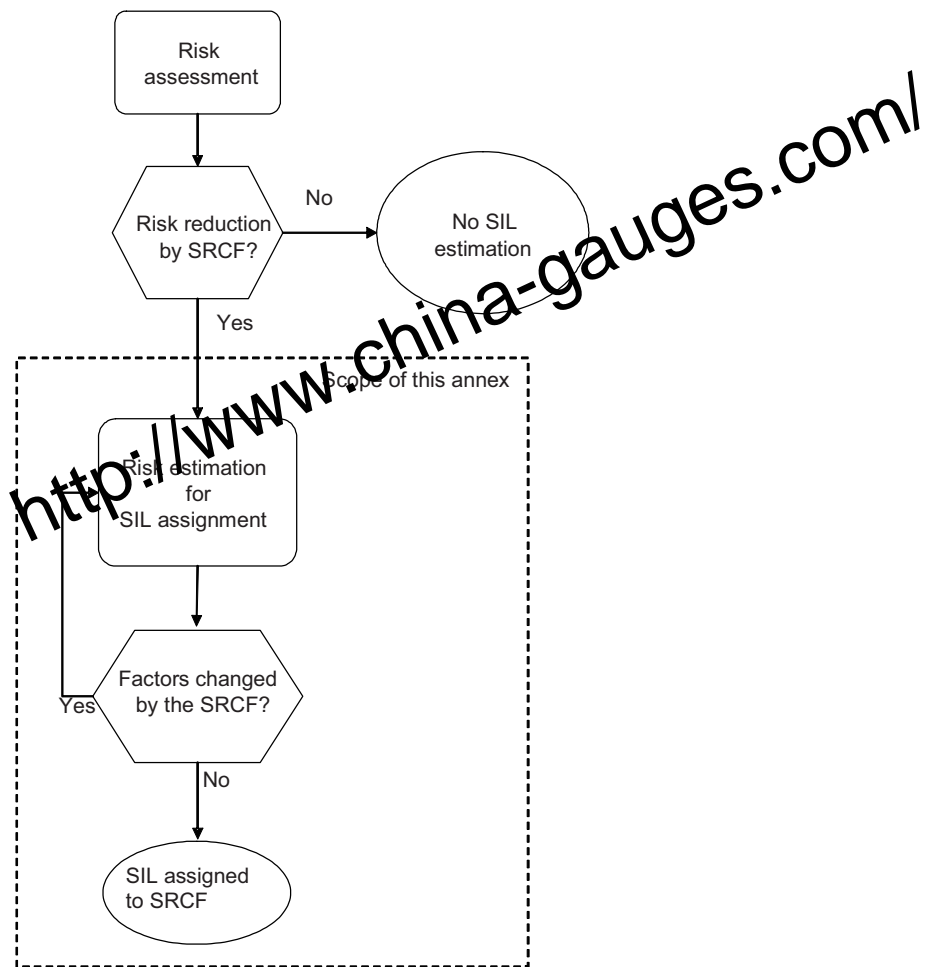


Figure A.1 – Workflow of SIL assignment process

Risk estimation is an iterative process, this means that the process will need to be carried out more than once.

Figure A.1 shows a feedback arrow to risk estimation. This is required because the provision of a particular protective measure to implement an SRCF may have an affect on the risk parameters (e.g. the use of a protective light curtain may result in a greater frequency of access). A failure of the light curtain will then expose the operator to a greater risk than originally envisaged. This requires that the process should be repeated following the same method but using the amended risk parameter(s).

At the end of the process shown in Figure A.1, the SIL estimated is the SIL requirement for the safety-related control function.

A.2 Risk estimation and SIL assignment

A.2.1 Hazard identification/indication

Indicate the hazards, including those from reasonable foreseeable misuse, whose risks are to be reduced by implementing an SRCF. List them in the hazard column in Table A.5.

A.2.2 Risk estimation

Risk estimation should be carried out for each hazard by determining the risk parameters that as shown in Figure A.2 should be derived from the following:

- severity of harm, Se; and
- probability of occurrence of that harm, which is a function of:
 - frequency and duration of the exposure of persons to the hazard, Fr;
 - probability of occurrence of a hazardous event, Pr; and
 - possibilities to avoid or limit the harm, Av.

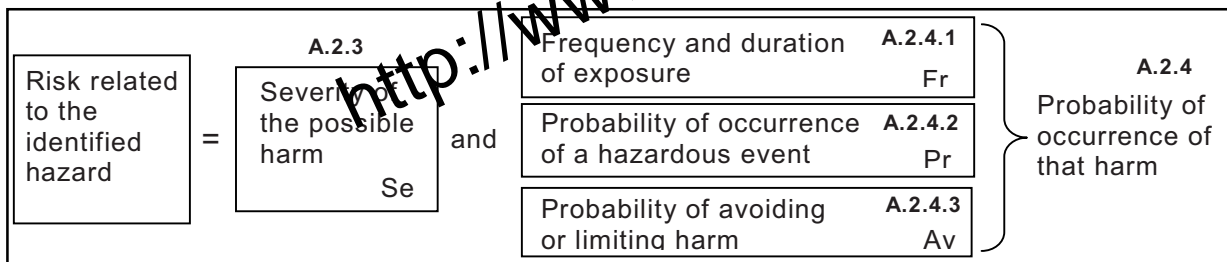


Figure A.2 – Parameters used in risk estimation

The estimates entered into Table A.5 should normally be based on worst-case considerations for the SRCF. However, in a situation where, for example, an irreversible injury is possible but at a significantly lower probability than a reversible one, then each severity level should have a separate line on the table. It may be the case that a different SRCF is implemented for each line. If one SRCF is implemented to cover both lines, then the highest target SIL requirement should be used.

A.2.3 Severity (Se)

Severity of injuries or damage to health can be estimated by taking into account reversible injuries, irreversible injuries and death. Choose the appropriate value of severity from Table A.1 based on the consequences of an injury, where:

- 4 means a fatal or a significant irreversible injury such that it will be very difficult to continue the same work after healing, if at all;
- 3 means a major or irreversible injury in such a way that it can be possible to continue the same work after healing. It can also include a severe major but reversible injury such as broken limbs;
- 2 means a reversible injury, including severe lacerations, stabbing, and severe bruises that requires attention from a medical practitioner;
- 1 means a minor injury including scratches and minor bruises that require attention by first aid.

Select the appropriate row for consequences (Se) of Table A.1. Insert the appropriate number under the Se column in Table A.5.

Table A.1 – Severity (Se) classification

Consequences	Severity (Se)
Irreversible: death, losing an eye or arm	4
Irreversible: broken limb(s), losing a finger(s)	3
Reversible: requiring attention from a medical practitioner	2
Reversible: requiring first aid	1

A.2.4 Probability of occurrence of harm

Each of the three parameters of probability of occurrence of harm (i.e. Fr, Pr and Av) should be estimated independently of each other. A worst-case assumption needs to be used for each parameter to ensure that SRCF(s) are not incorrectly assigned a lower SIL than is necessary. Generally, the use of a form of task-based analysis is strongly recommended to ensure that proper consideration is given to estimation of the probability of occurrence of harm.

A.2.4.1 Frequency and duration of exposure

Consider the following aspects to determine the level of exposure:

- need for access to the danger zone based on all modes of use, for example normal operation, maintenance; and
- nature of access, for example manual feed of material, setting.

It should then be possible to estimate the average interval between exposures and therefore the average frequency of access.

AC2 It should also be possible to foresee the duration, for example if it will be longer than 10 min. Where the duration is shorter than 10 min, the value may be decreased to the number in the row below in Table A.2. This does not apply to frequency of **A2** exposure ≥ 1 per h, which should not be decreased in value at any time **A2**. **AC2**

NOTE The duration is related to the performance of activities that are carried out under the protection of the SRCF. The requirements of IEC 60204-1 and ISO 14118 with regard to power isolation and energy dissipation should be applied for major interventions.

This factor does not include consideration of the failure of the SRCF.

Select the appropriate row for frequency and duration of exposure (Fr) of Table A.2. Insert the appropriate number under the Fr column in Table A.5.

Table A.2– Frequency and duration of exposure (Fr) classification

Frequency and duration of exposure (Fr)	
Frequency of exposure	Frequency, Fr (see A.2.4.1)
A1 ≥ 1 per h A1	5
< 1 per h to ≥ 1 per day	5
< 1 per day to ≥ 1 per 2 weeks	4
< 1 per 2 weeks to ≥ 1 per year	3
< 1 per year	2

AC2

A.2.4.2 Probability of occurrence of a hazardous event

The probability of occurrence of harm should be estimated independently of other related parameters F_r and A_v . A worst-case assumption should be used for each parameter to ensure that SRCF(s) are not incorrectly assigned a lower SIL than is necessary. To prevent this occurring the use of a form of task-based analysis is strongly recommended to ensure that proper consideration is given to estimation of the probability of occurrence of harm.

This parameter can be estimated by taking into account:

- a) Predictability of the behaviour of component parts of the machine relevant to the hazard in different modes of use (e.g. normal operation, maintenance, fault finding).

This will necessitate careful consideration of the control system especially with regard to the risk of unexpected start up. Do not take into account the protective effect of any SRECS. This is necessary in order to estimate the amount of risk that will be exposed if the SRECS fails. In general terms, it must be considered whether the machine or material being processed has the propensity to act in an unexpected manner.

The machine behaviour will vary from very predictable to not predictable but unexpected events cannot be discounted.

NOTE 1 Predictability is often linked to the complexity of the machine function.

- b) The specified or foreseeable characteristics of human behaviour with regard to interaction with the component parts of the machine relevant to the hazard. This can be characterised by:

- stress (e.g. due to time constraints, work task, perceived damage limitation); and/or
- lack of awareness of information relevant to the hazard. This will be influenced by factors such as skills, training, experience, and complexity of machine/process.

These attributes are not usually directly under the influence of the SRECS designer, but a task analysis will reveal activities where total awareness of all issues, including unexpected outcomes, cannot be reasonably assumed.

“Very high” probability of occurrence of a hazardous event should be selected to reflect normal production constraints and worst case considerations. Positive reasons (e.g. well-defined application and knowledge of high level of user competences) are required for any lower values to be used.

NOTE 2 Any required or assumed skills, knowledge, etc. should be stated in the information for use.

Select the appropriate row for probability of occurrence of hazardous event (P_r) of Table A.3. Indicate the appropriate number under the P_r column in Table A.5.

Table A.3– Probability (P_r) classification

Probability of occurrence	Probability (P_r)
Very high	5
Likely	4
Possible	3
Rarely	2
Negligible	1

A.2.4.3 Probability of avoiding or limiting harm (Av)

This parameter can be estimated by taking into account aspects of the machine design and its intended application that can help to avoid or limit the harm from a hazard. These aspects include, for example

- sudden, fast or slow speed of appearance of the hazardous event;
- spatial possibility to withdraw from the hazard;
- the nature of the component or system, for example, a knife is usually sharp, a pipe in a dairy environment is usually hot, electricity is usually dangerous by its nature but is not visible; and
- possibility of recognition of a hazard, for example electrical hazard: a copper bar does not change its aspect whether it is under voltage or not; to recognize if one needs an instrument to establish whether electrical equipment is energised or not; ambient conditions, for example high noise levels can prevent a person hearing a machine start.

Select the appropriate row for probability of avoidance or limiting harm (Av) of Table A.4. Insert the appropriate number under the Av column in Table A.5.

Table A.4– Probability of avoiding or limiting harm (Av) classification

Probabilities of avoiding or limiting harm (AV)	
Impossible	5
Rarely	3
Probable	1

A.2.5 Class of probability of harm (CI)

For each hazard, and as applicable, for each severity level add up the points from the Fr, Pr and Av columns and enter the sum into the column CI in Table A.5.

Table A.5– Parameters used to determine class of probability of harm (CI)

Serial no.	Hazard	Se	Fr	Pr	Av	CI
1						
2						
3						
4						

A.2.6 SIL assignment

Using Table A.6, where the severity (Se) row crosses the relevant column (CI), the intersection point indicates whether action is required. The black area indicates the SIL assigned as the target for the SRCF. The lighter shaded areas should be used as a recommendation that other measures (OM) be used.

Table A.6 – SIL assignment matrix

Severity (Se)	Class (Cl)				
	4	5-7	8-10	11-13	14-15
4	SIL 2	SIL 2	SIL 2	SIL 3	SIL 3
3		(OM)	SIL 1	SIL 2	SIL 3
2			(OM)	SIL 1	SIL 2
1				(OM)	SIL 1

EXAMPLE: For a specific hazard with an Se assigned as 3, an Fr as 4, an Pr as 5 and an Av as 5 then:

<http://www.china-s.com/>

$$Cl = Fr + Pr + Av = 4 + 5 + 5 = 14$$

Using Table A.6, this would lead to a SIL 3 being assigned to the SRCF that is intended to mitigate against the specific hazard.

Figure A.3 shows an example of documentation that may be used to record the results of a SIL assignment exercise using this informative Annex.

AC2

Document No.:
 Part of:

Product: _____
 Issued by: _____
 Date: _____

Risk assessment and safety measures

- Pre risk assessment
- Intermediate risk assessment
- Follow up risk assessment

Black area = Safety measures required
 Grey area = Safety measures recommended

Consequences	Severity Se	Class Cl						Frequency, Fr	Probability of hzd. event, Pr	Avoidance Av
		4	5 - 7	8 - 10	11 - 13	14 - 15				
Death, losing an eye or arm	4	SIL 2	SIL 2	SIL 2	SIL 3	SIL 3	≥ 1 per hr	Common	5	
Permanent, losing fingers	3		OM	SIL 1	SIL 2	SIL 3	<1 per hr - ≥ 1 per day	Likely	4	
Reversible, medical attention	2			OM	SIL 1	SIL 2	<1 per day - ≥1 per 14days	Possible	3	
Reversible, first aid	1				OM	SIL 1	<1 per 2wks - ≥1 per yr	Rarely	2	
							<1 per yr	Negligible	1	

Ser. Hzd. No. No.	Hazard	Se	Fr	Pr	Av	Cl	Safety measure

Comments

http://www.china-gauges.com/

Figure A.3 – Example proforma for SIL assignment process

AC2

Annex B
(informative)

**Example of safety-related electrical control system (SRECS) design
using concepts and requirements of Clauses 5 and 6**

B.1 General

The structured approach to the design of a SRECS used by this standard defines a methodology whereby functional and safety integrity requirements for safety-related control functions are decomposed into a number of sub-functions. This process is used to implement into the machinery sector a technical framework for functional safety and Figure B.1 describes the terminology used at each of these levels that is important when integrating a SRECS design at a machine installation.

This design methodology can by verification and validation processes be used to demonstrate that a SRECS fulfils the safety requirements specification described in Clause 5.

The following example of a SRECS design is intended to clarify the principles of functional decomposition and the realisation of a specified safety-related control function in accordance with the requirements of Clause 6. Consequently this example is simplified and does not consider additional measures that can be required in practice, for example hold-to-run devices.

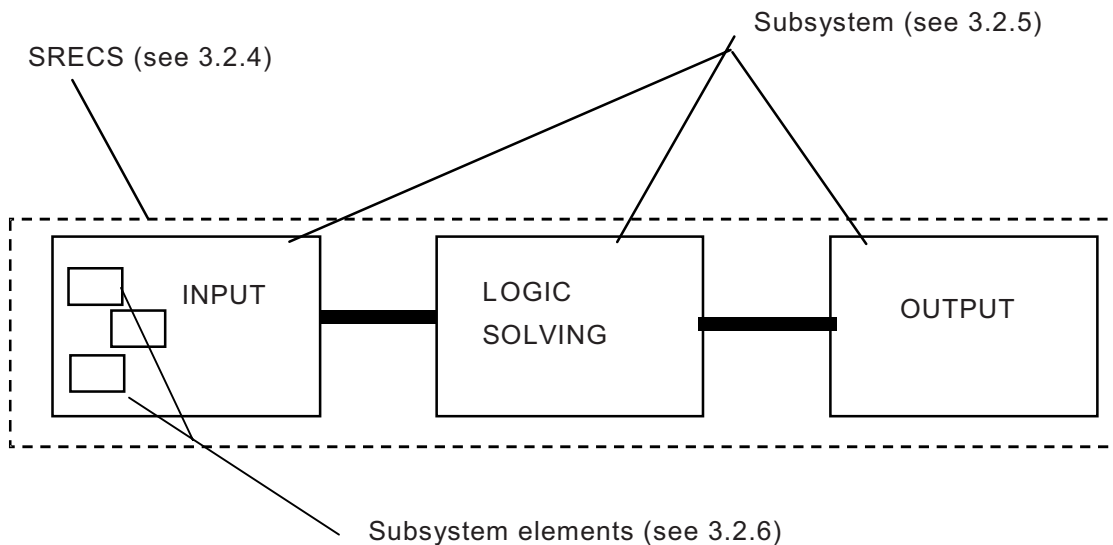


Figure B.1 – Terminology used in functional decomposition

In general, the terms presented in Figure B.1 are intended to delineate the design process into two key stages namely:

- SRECS design that may be carried out by a machinery designer or a control systems integrator; and

- subsystem (and subsystem element) design that is applicable to the vendors of electrical equipment and controlgear (e.g. contactors, interlocking switches, programmable logic controllers) and the machine designers or control system integrators.

B.2 Example

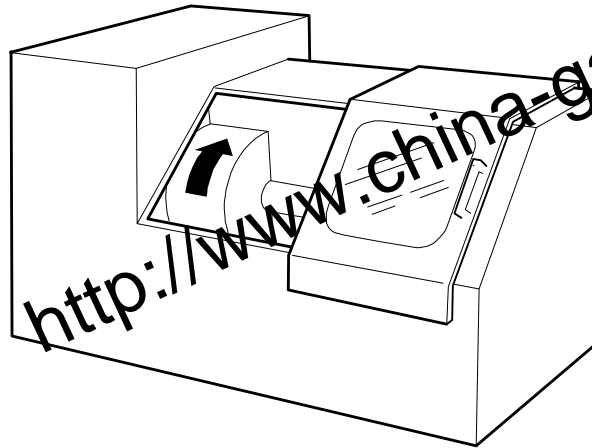


Figure B.2 – Example machine

The methodology used in this standard is based upon a structured top-down approach to the specification of safety-related control functions and the design of the SRECS that implements those functions.

Step 1: SRCF safety requirements specification (Clause 5)

From a SRCF safety requirements specification the following information can be derived:

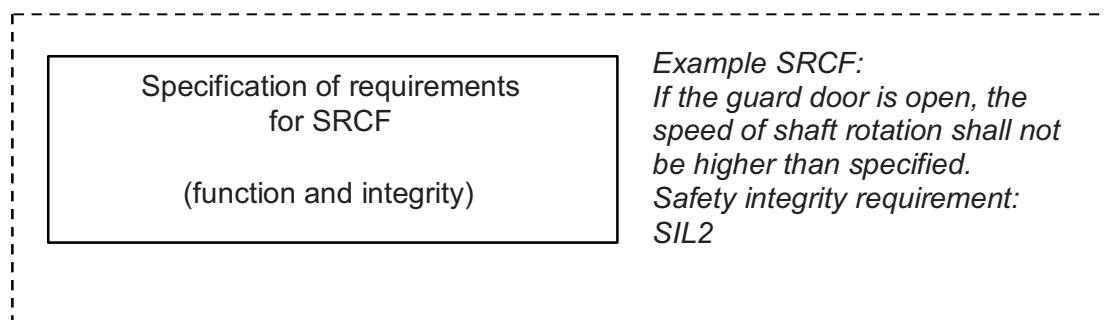


Figure B.3 – Specification of requirements for an SRCF

Step 2: SRECS design and development process (see 6.6.2)

Step 2.1: The safety-related control function as specified in the safety requirements specification is decomposed to a structure of function blocks.

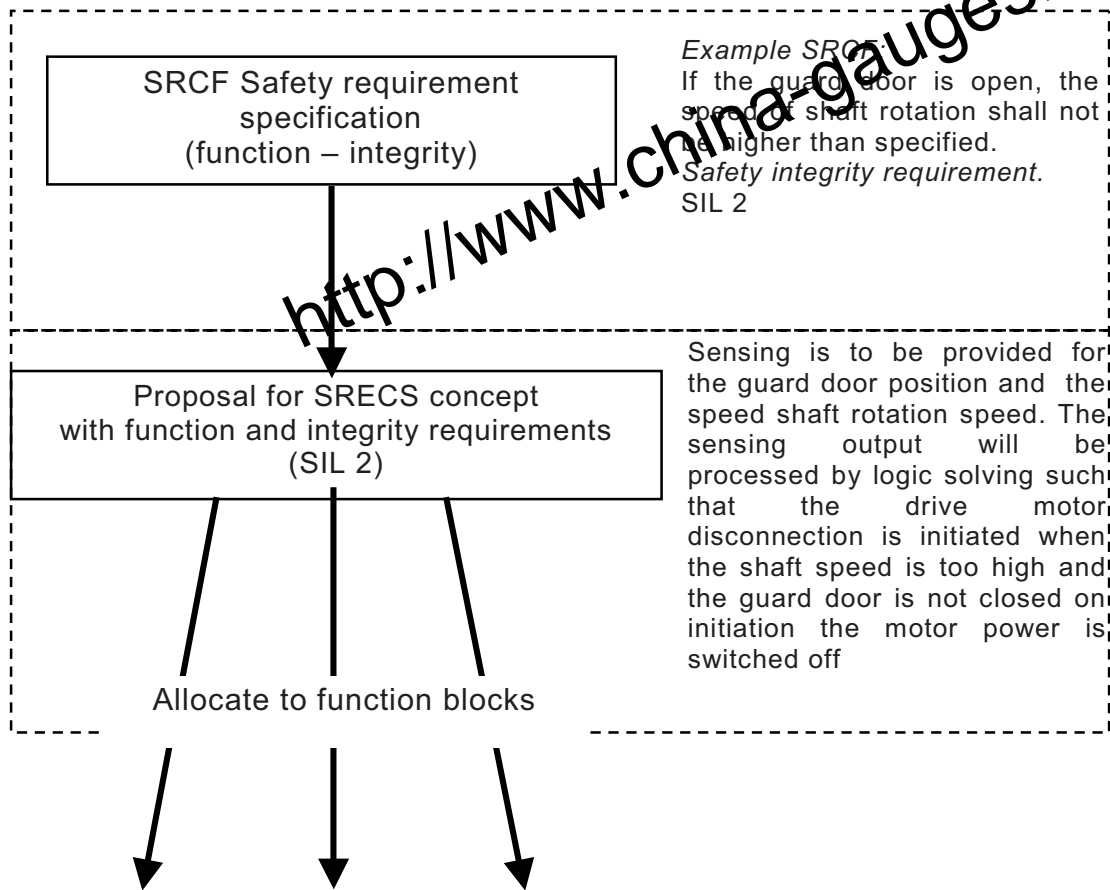


Figure B.4 – Decomposition to a structure of function blocks

Step 2.2: The structure of function blocks provides an initial concept for an architecture of the SRECS. The safety requirements for each function block are derived from the safety requirements specification of the corresponding safety-related control function.

The element(s) that implement each function block must achieve at least the same SIL capability as that assigned to the SRCF. This is shown in Figure B.5 as a SIL 2 capability (i.e. FB1 SILCL2, etc.).

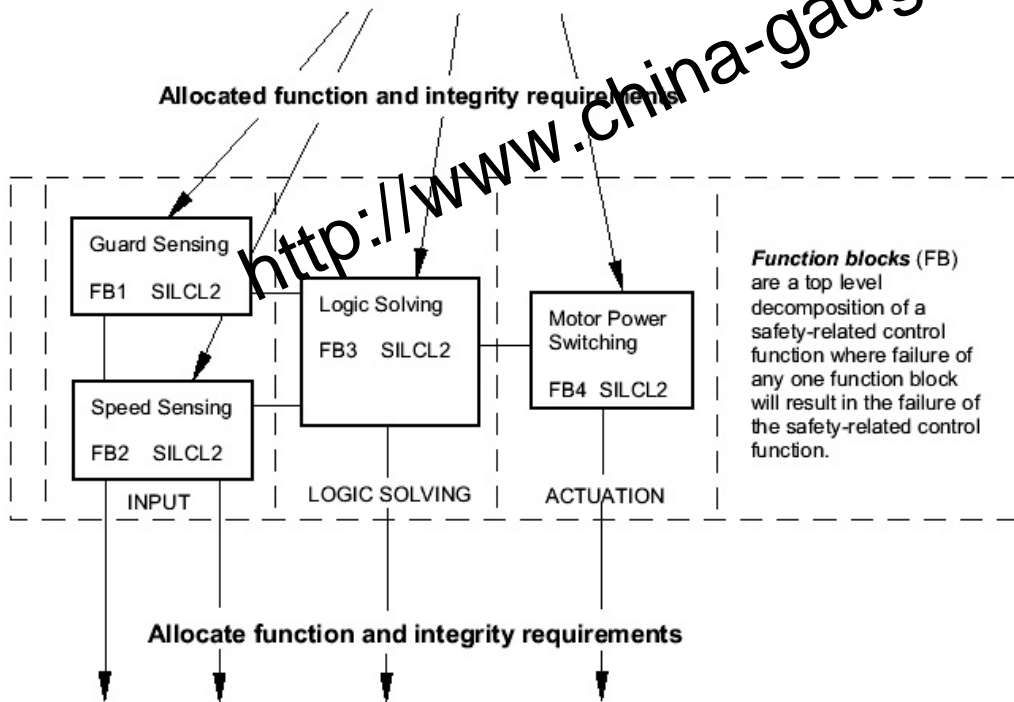


Figure B.5 – Initial concept of an architecture for a SRECS

Step 3: Each function block is allocated to a subsystem within the architecture of the SRECS. Each subsystem may consist of subsystem elements and, as necessary, diagnostic functions to ensure that faults can be detected and appropriate action taken (see 6.2).

The architecture should describe the SRECS in terms of its subsystems and their interrelationship. For this example there are a number of alternatives that can be used for realisation of the SRECS and its subsystems architecture.

Example 1: In this example (see Figure B.6), the diagnostic functions are embedded within each subsystem.

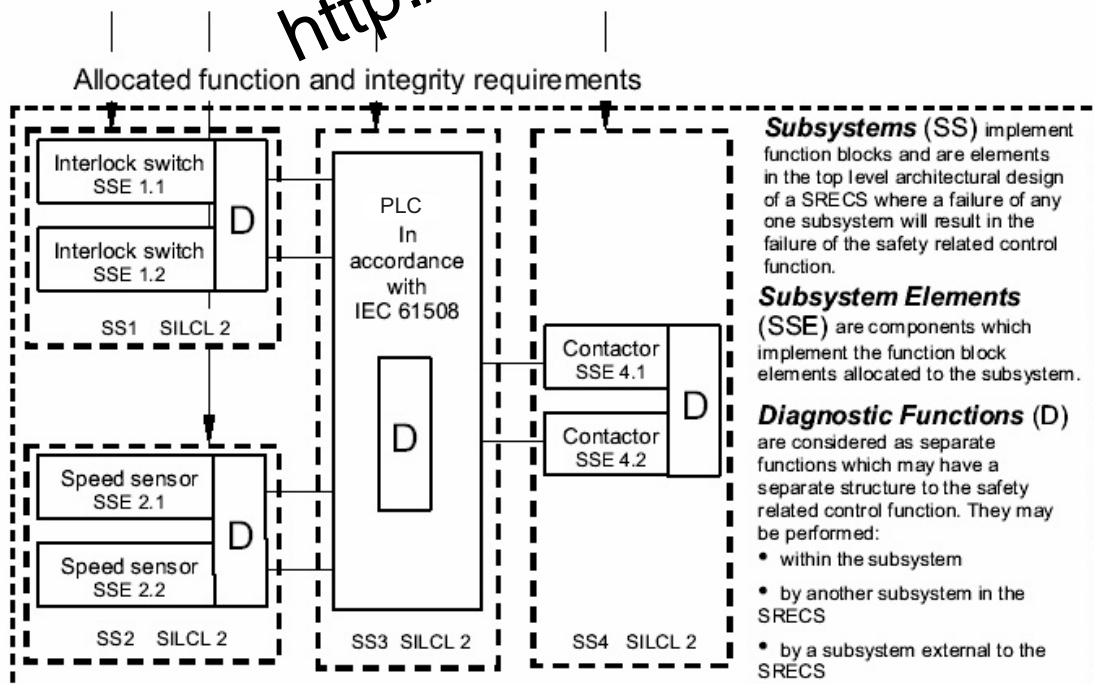


Figure B.6 – SRECS architecture with diagnostic functions embedded within each subsystem (SS1 to SS4)

Example 2: In this example (see Figure B.7), the diagnostic functions are embedded within a programmable logic controller (PLC) in SS3 that satisfies relevant aspects of IEC 61508.

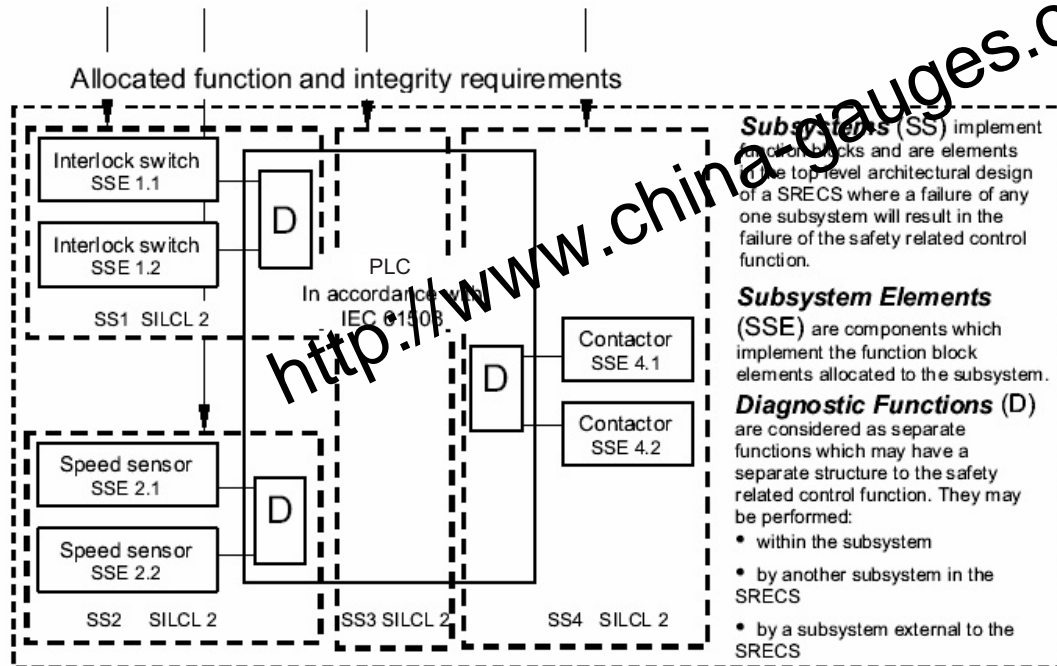


Figure B.7 – SRECS architecture with diagnostic functions embedded within subsystem SS3

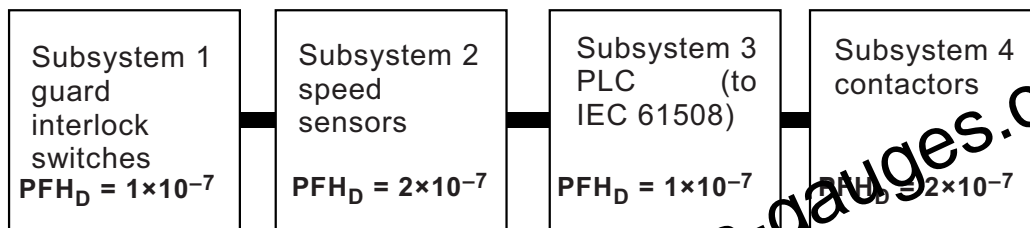
Step 4: Estimation of the SIL achieved by the SRECS (see 6.6.3)

The SIL that can be claimed for the SRECS shall be less than or equal to the lowest value of the SILCLs of any of the subsystems. The probability of dangerous random hardware failure of the SRECS (PFH_{DSRECS}) is the sum of the probabilities of dangerous failure per hour of all subsystems (PFH_{D1} to PFH_{Dn}) involved in the performance of the safety-related control function and shall include, where appropriate, the probability of dangerous transmission errors (P_{TE}) for digital data communication processes as:

$$PFH_{DSRECS} = PFH_{D1} + \dots + PFH_{Dn} + P_{TE}$$

For this example, the target failure value for the safety-related control function is SIL 2 and from Table 3 (see 5.2.4.2) this is equivalent to a probability of dangerous failure per hour (PFH_D) in the range $\geq 10^{-7}$ to $< 10^{-6}$. Therefore, assuming that the probabilities of dangerous failure per hour of each of the subsystems are as shown below, the sum of the probabilities of dangerous failure per hour of all subsystems can be estimated as shown in Figure B.8.

Therefore, in this example, the design of SRECS can be shown to satisfy all the requirements to implement the assigned safety-related control function at SIL 2.



$$PFH_{DSRECS} = (1 \times 10^{-7}) + (2 \times 10^{-7}) + (1 \times 10^{-7}) + (2 \times 10^{-7}) = 6 \times 10^{-7}$$

Figure B.8 → Estimation of PFH_D for a SRECS

Annex C (informative)

Guide to embedded software design and development

NOTE This informative Annex is provided to indicate the basic approach required in order to satisfy the requirements of IEC 61508-3. It cannot in itself provide conformance with IEC 61508-3 without applying further measures.

C.1 General

This Annex is provided to assist persons in the design and development of embedded software for implementing safety-related control functions within a SRECS.

The major objective dealt with here is general guidance on the prevention of embedded software failures and any other unexpected behaviour of embedded software that might lead to the creation of dangerous faults in the system.

In order to satisfy these objectives, consideration is given to the following points:

- a description of the main characteristics that software elements of a SRECS should possess to guarantee its quality and safety (software element guidelines);
- the establishment of all relevant technical activities and provisions associated with software development, for those involved in software design. These can then be used to guide the designer during the production of this type of software (software development process guidelines);
- a reference framework for software evaluation. This allows the software designer and/or analyst to decide that software elements satisfy the safety requirements of the SRECS or SRECS subsystem to be analysed (software verification guidelines).

This Annex provides a set of basic guidelines, coherent with the IEC 61508-3, that are adapted to embedded software for microprocessors.

C.2 Software element guidelines

This Clause presents the guidelines that an embedded software element of a SRECS or SRECS subsystem should fulfil to be safe in operation and of satisfactorily high quality. To obtain such a software element, a number of activities, a certain organisation and a number of principles should all be established. This should take place as early as possible in the development cycle.

C.2.1 Interface with system architecture

The list of constraints imposed by hardware architecture on software should be defined and documented. Consequences of any hardware/software interaction on the safety of the machine or system being monitored should be identified and evaluated by the designer, and taken into account in the software design.

NOTE Constraints include: protocols and formats, input/output frequencies, by rising and falling edge or by level, input data using reverse logic, etc. Listing these constraints allows them to be taken into account at the start of the development activity, and reduces the risk of incompatibilities between software and hardware when the former is installed in the target hardware.

C.2.2 Software specifications

Software specifications should take the following points into account:

- safety-related control functions with a quantitative description of the performance criteria (precision, exactness) and temporal constraints (response time), all with tolerances or margins when possible;
- system configuration or architecture;
- instructions relevant to hardware safety integrity (logic solvers, sensors, actuators, etc.);
- instructions relevant to software integrity;
- constraints related to memory capacity and system response time;
- operator and equipment interfaces;
- instructions for software self-monitoring and for hardware monitoring carried out by the software;
- instructions that allow all the safety-related control functions to be verified while the systems are working (e.g. on-line testing, capture time for fleeting signals, coincidence with scan rate).

NOTE 1 The instructions for monitoring, developed taking safety objectives and operating constraints (duration of continuous operation, etc.) into account, can include devices such as watch dogs, central processing unit (CPU) load monitoring, feedback of output to input for software self-monitoring. For hardware monitoring, CPU and memory monitoring, etc. instructions for safety-related control function verification: for example, the possibility of periodically verifying the correct operation of safety devices should be included in the specifications.

Functional requirements should be specified for each functional mode. The transition from one mode to the other should be specified.

NOTE 2 Functional modes can include nominal modes, and one or more degraded modes. The objective is to specify the behaviour in all situations in order to avoid unexpected behaviours in non-nominal contexts.

C.2.3 Pre-existent software

The term "pre-existent" software refers to source modules that have not been developed specifically for the system at hand, and are integrated into the rest of the software. These include software elements developed by the designer for previous projects, or commercially available software (e.g. modules for calculations, algorithms for data sorting).

When dealing with this type of software, and especially in the case of commercial software elements, the designer does not always have access to all the elements needed to satisfy the previous requirements (e.g. what tests have been carried out, is the design documentation available). Specific co-ordination with the analyst can therefore be necessary at the earliest possible moment.

The designer should indicate the use of pre-existent software to the analyst, and the designer should demonstrate that pre-existent software has the same level as the other software elements. Such a demonstration should be done:

- a) either by using the same verification activities on the pre-existent software as on the rest of the software; and/or
- b) through practical experience where the pre-existent software has functioned on a similar system in a comparable executable environment (e.g. it may be necessary to evaluate the consequences of a change of the compiler or of a different software architecture format).

NOTE 1 The goal of indicating the use of pre-existent software is to open up consultation with the analyst as early as possible about any eventual difficulties that this type of software might cause. The integration of pre-existent source modules can be the cause of certain anomalies or unsafe behaviour if they were not developed with the same rigour as the rest of the software.

Pre-existent software should be identified using the same configuration management and version control principles that are applied to the rest of the software.

NOTE 2 Configuration management and version control should be exercised over all the software components, regardless of their origin.

C.2.4 Software design

Description of the software design should include a description of:

- the software architecture that defines the structure decided to satisfy specifications;
- inputs and outputs (e.g. in the form of an internal and external data dictionary), for all the modules making up the software architecture;
- the interrupts;
- the global data;
- each software module (inputs/outputs, algorithm, design particularities, etc.);
- module or data libraries used;
- pre-existent software used.

Software should be modular and written in a logical manner in order to facilitate its verification or maintenance:

- each module or group of modules should correspond, if possible, to a function in the specification(s);
- interfaces between modules should be as simple as possible.

NOTE The general characteristic of correct software architecture can be summed up in the following way: a module should possess a high level of functional cohesion and a simple interface with its environment.

Software should:

- limit the number or extent of global variables;
- control the layout of arrays in memory (to avoid a risk of array overflows).

C.2.5 Coding

The source code should:

- be readable, understandable, and subject to tests;
- satisfy design specifications of the software module;
- obey the coding manual instructions.

C.3 Software development process guidelines

C.3.1 Development process: software lifecycle

The objective of the following guidance applicable to the software lifecycle is to obtain a formalized description of the organization of software development and, in particular, the different technical tasks making up this development.

The software development lifecycle should be specified and documented (e.g. in a software quality plan). The lifecycle should include all the technical activities and phases necessary and sufficient for software development.

Each phase of the lifecycle should be divided into its elementary tasks and should include a description of:

- inputs (documents, standards, etc.);
- outputs (documents produced, analytical reports, etc.);
- activities to be carried out;
- verifications to be performed (analyses, tests, etc.).

C.3.2 Documentation : documentation management

The documentation should conform to Clause 10 of this standard.

C.3.3 Configuration and software modification management

Management of the configuration and therefore of the version is an indispensable part of any development which may require approval. Indeed, approval is only valid where a given configuration can be identified. Configuration management includes configuration identification activities, modification management, the establishment of reference points and the archiving of software elements, including the associated data (documents, records of tests, etc.). Throughout the entire project lifecycle, the principal objectives are to provide:

- a defined and controlled software configuration that guarantee physical archiving and that can be used to reproduce an executable code coherently (with future software production or modification in mind);
- a reference basis for modifications management;
- a means of control so that any problems are properly analysed, and that the approved modifications are properly carried out.

Concerning the modifications, their reasons could arise from, for example:

- functional safety below that specified;
- systematic fault experience;
- new or amended safety legislation;
- modifications to the machine or its use;
- modification to the overall safety requirements;
- analysis of operations and maintenance performance, indicating that the performance is below target.

C.3.4 Configuration and archiving management

A procedure for configuration management and modifications management should be defined and documented. This procedure should, as a minimum, include the following items:

- articles managed by the configuration, at least: software specification, preliminary and detailed software design, source code modules, plans, procedures and results of the validation tests;
- identification rules (of a source module, of a software version, etc.);
- treatment of modifications (recording of requests, etc.).

For each article of configuration, it should be possible to identify any changes that may have occurred and the versions of any associated elements.

NOTE 1 The purpose is to be able to trace the historical development of each article: what modifications have been made, why, and when.

Software configuration management should allow a precise and unique software version identification to be obtained. Configuration management should associate all the articles (and their version) needed to demonstrate the functional safety.

All articles in the software configuration should be covered by the configuration management procedure before being tested or being requested by the analyst for final software version evaluation.

NOTE 2 The objective here is to ensure that the evaluation procedure be performed on software with all elements in a precise state. Any subsequent change may necessitate revision of the software so that it can be identifiable by the analyst.

Procedures for the archiving of software and its associated data should be established (methods for storing backups and archives).

NOTE 3 These backups and archives can be used to maintain and modify software during its functional lifetime.

C.3.5 Software modifications management

Any software modification which has an impact on the functional safety of the SRECS should be subject to the rules established for modification and configuration management such that the development process be recommenced at the highest "upstream" point needed to take the modification into account without diminishing the functional safety.

NOTE In particular, the documentation should also be updated, and all necessary verification activities carried out. This guarantees that the software will keep all its initial properties after any modification.

C.4 Development tools

Tools used during the development procedure (compiler, linker, tests, etc.) should be identified (name, reference, version, etc.) in the documentation associated with the software version (e.g. in the version control documentation).

NOTE Different versions of tools do not necessarily produce the same results. Precise identification of tools thus directly demonstrates the continuity of the process of generation of an executable version in the event that a version is modified.

C.5 Reproduction, delivery

C.5.1 Executable code production

Any option or change in the generation, during the software production should be recorded (e.g. in the version sheet) so that it is possible to say how and when the software was generated.

C.5.2 Software installation and exploitation

All failures linked to safety-related control functions brought to the attention of the designer of the system should be recorded and analysed.

NOTE This means that the designer is aware of any safety-related failures that are communicated to him and that he takes the appropriate action (e.g. warning other users, software modification, etc.).

C.6 Software verification and validation

The purpose of verification activities is to demonstrate that software elements stemming from a given phase of the development cycle conform to the specifications established during the previous phases and to any applicable standards or rules. They also serve as a means of detecting and accounting for any errors that might have been introduced during software development.

Software verification is not simply a series of tests, even though this is the predominant activity for the relatively small software element considered in this Annex. Other activities such as reviews and analyses, whether associated with these tests or not, are also considered to be verification activities. In certain cases, they can replace some tests (e.g. in the event that a test cannot be carried out because it would cause deterioration of a hardware component).

C.7 General verification and validation guidelines

The analyst should be able to carry out the evaluation of software conformity by conducting any audits or expertises deemed useful during the different software development phases.

All technical aspects of software lifecycle processes are subject to evaluation by the analyst. The analyst should be allowed to consult all verification reports (tests, analyses, etc.) and all technical documents used during software development.

NOTE 1 The intervention of the analyst at the specification phase is preferable to an *a posteriori* intervention since it should limit the impact of any decisions made. On the other hand, financial and human aspects of the project are not subject to evaluation.

NOTE 2 It is in the interest of the applicant to provide satisfactory evidence of all activities carried out during software development.

NOTE 3 The analyst should have all the necessary elements at his or her disposal in order to formulate an opinion.

Evaluation of software conformity is performed for a specific, referenced software version. Any modification of previously evaluated software that has received a final opinion from the analyst should be pointed out to the latter so that any additional evaluation activities can be carried out to update this opinion.

NOTE 4 Any modification can modify software behaviour; the evaluation performed by the analyst can therefore only be applied to a precise software version.

C.8 Verification and validation review

Analysis activities and software design verification should verify the conformity to specifications.

NOTE 1 The purpose is to ensure that the software specification and design (both detailed and preliminary) are coherent.

An external validation review (with the analyst) should be held at the end of the validation phase.

NOTE 2 This can be used to ascertain whether or not the element satisfies the specifications.

The result of each review should be documented and archived. It should include a list of all actions decided on in the review process, and the review conclusion (decision on whether or not to move on to the next activity). The activities defined in the review should be monitored and treated.

C.9 Software testing

C.9.1 General validation

Before writing the first test sheets, it is important to establish a test strategy in a test plan. This strategy indicates the approach adopted, the objectives that have been set in terms of test coverage, the environments and specific techniques used, the success criteria to be applied, etc.

The test objectives should be adapted to the type of software, and to the specific factors. These criteria determine the types of test to be undertaken – functional tests, limit tests, out of limit tests, performance tests, load tests, internal equipment failure tests, configuration tests – as well as the range of objects to be covered by the tests (functional mode tests, safety-related control function tests, tests of each element in the specification, etc.).

Verification of a new software version should include non-regression tests.

NOTE Non-regression tests are used to ensure that the modifications performed on the software have not modified the behaviour of the software in any unexpected way.

C.9.2 Software specification verification: validation tests

The purpose of these verifications is to detect errors associated with the software in the target system environment. Errors detected by this type of verification include: any incorrect mechanism to treat interruptions, insufficient respect of running time requirements, incorrect response from the software operating in transient mode (start-up, input flow, switching in a degraded mode, etc.), conflicts of access to different resources or organizational problems in the memory, inability of integrated tests to detect faults, software/hardware interface errors, stack overflows. Validation tests are the principal component of software specification verification.

The test coverage should be made explicit in a traceability matrix and ensure that:

- each element of the specification, including safety mechanisms, is covered by a validation test; and
- the real-time behaviour of the software in any operational mode can be verified.

Furthermore, the validation should be carried out in conditions representative of the operational conditions of the SRECS or the SRECS subsystem.

NOTE 1 This guarantees that the software reacts as expected in operation. It applies only to cases where the test conditions can be destructive for hardware (e.g. physical fault of a component that cannot be simulated). To be significant, validation should be performed in the operational conditions of the SRECS or SRECS subsystem (i.e. with the final versions of software and hardware, and the software installed in the target system). Any other combination could decrease the efficiency of the test and require analysis of its representation.

Validation results should be recorded in a validation report that should cover at least the following points:

- the versions of software and system that were validated;
- a description of the validation tests performed (inputs, outputs, testing procedures);
- the tools and equipment used to validate or evaluate the results;

- the results showing whether each validation test was a success or failure;
- a validation assessment: identified non-conformities, impact on safety, decision as to whether or not to accept the validation.

A validation report should be made available for each delivered software version and should correspond to the final version of each delivered software element.

NOTE 2 This report can be used to provide proof that tests were indeed carried out and that the results were correct (or contained explainable deviations). It can also be used to redo tests at a later date, for a future software version or for another project. It provides a guarantee that each delivered version has been validated in its final form. On the other hand, it does not impose a complete validation of each modification of an existing code – an impact analysis can, in certain cases, justify partial validation.

C.9.3 Software design verification: software integration tests

This verification focuses on the correct assembly of software modules and on the mutual relationships between software components. It can be used to reveal errors of the following kind: incorrect initialization of variables and constants, errors in the transfer of parameters, any data alteration, especially global data, incorrect sequencing of events and operations.

Software integration tests should be able to verify:

- correct sequencing of the software execution;
- exchange of data between modules;
- respect of the performance criteria;
- non-alteration of global data.

The test coverage should be given explicitly in a traceability matrix demonstrating the correspondence between the tests to be undertaken and the objectives of the tests defined.

Integration test results should be recorded in a software integration test report, which should, as a minimum, contain the following points:

- the version of the integrated software;
- a description of the tests performed (inputs, outputs, procedures);
- the integration tests results and their evaluation.

C.9.4 Detailed design verification: module tests

Module tests focus on software modules and their conformity with the detailed design. This activity can be indispensable for large and complex software elements, but is only recommended for the relatively small software elements dealt with here. This phase of the verification procedure allows detection of the following types of errors:

- inability of an algorithm to satisfy software specifications;
- incorrect loop operations;
- incorrect logical decisions;
- inability to compute valid combinations of input data correctly;
- incorrect responses to missed or altered input data;
- violation of array boundaries;
- incorrect calculation sequences;

- inadequate precision;
- accuracy or performance of an algorithm.

Each software module should be submitted to a series of tests to verify, using input data, that the module fulfils the functions specified at the detailed design stage.

The test coverage should be provided in a traceability matrix that demonstrates the correspondence between the test results and the objectives of the tests defined.

<http://www.china-gauges.com/>

Annexes D and E deleted

<http://www.china-gauges.com/>

Annex F
 (informative)

**Methodology for the estimation of susceptibility
 to common cause failures (CCF)**

F.1 General

This informative Annex provides a simple qualitative approach for the estimation of CCF that can be applied to the subsystem design.

F.2 Methodology

The proposed design of a subsystem should be assessed to establish the effectiveness of the measures used to safeguard against CCF. The items in Table F.1 that are applicable should be identified and an overall score established, which is used to determine the common cause failure factor from Table F.2 as a percentage value.

Table F.1 – Criteria for estimation of CCF

Item	Reference	Score
Separation/segregation		
Are SRECS signal cables for the individual channels routed separately from other channels at all positions or sufficiently shielded?	1a	5
Where information encoding/decoding is used, is it sufficient for the detection of signal transmission errors?	1b	10
Are SRECS signal and electrical energy power cables separate at all positions or sufficiently shielded?	2	5
If subsystem elements can contribute to a CCF, are they provided as physically separate devices in their local enclosures?	3	5
Diversity/redundancy		
Does the subsystem employ different electrical technologies for example, one electronic or programmable electronic and the other an electromechanical relay?	4	8
Does the subsystem employ elements that use different physical principles (e.g. sensing elements at a guard door that use mechanical and magnetic sensing techniques)?	5	10
Does the subsystem employ elements with temporal differences in functional operation and/or failure modes?	6	10
Do the subsystem elements have a diagnostic test interval of ≤ 1 min?	7	10
Complexity/design/application		
Is cross-connection between channels of the subsystem prevented with the exception of that used for diagnostic testing purposes?	8	2
Assessment/analysis		
Have the results of the failure modes and effects analysis been examined to establish sources of common cause failure and have predetermined sources of common cause failure been eliminated by design?	9	9
Are field failures analysed with feedback into the design?	10	9
Competence/training		
Do subsystem designers understand the causes and consequences of common cause failures?	11	4

Item	Reference	Score
Environmental control		
Are the subsystem elements likely to operate always within the range of temperature, humidity, corrosion, dust, vibration, etc. over which it has been tested, without the use of external environmental control?	12	9
A1 Is the subsystem immune to adverse influences from electromagnetic interference up to and including the limits specified in IEC 61326-3-1? A1	7	9
NOTE An alternative item (e.g. references 1a and 1b) is given in Table F.1 where it is intended that a claim can be made for a contribution towards avoidance of CCF from only the most relevant item.		

Using Table F.1 those items that are considered to affect the subsystem design should be added to provide an overall score for the design that is to be implemented. Where it can be shown that equivalent means of avoiding of CCF can be achieved through the use of specific design measures (e.g. the use of opto-isolated devices rather than shielded cables) then the relevant score can be claimed as this can be considered to provide the same contribution to the avoidance of CCF.

This overall score can be used to determine a common cause failure factor (β) using Table F.2.

AC2 **Table F.2 – Estimation of CCF factor (β)**

Overall score	Common cause failure factor (β)
≤ 35	10 % (0,1)
36 – 65	5 % (0,05)
66 – 85	2 % (0,02)
86 – 100	1 % (0,01)

AC2

The value of β derived should be used in the estimation of the probability of dangerous failure as required in 6.7.8.1.

<http://www.china-gauges.com/>

British Standards Institution (BSI)

BSI is the national body responsible for preparing British Standards and other standards-related publications, information and services.

BSI is incorporated by Royal Charter. British Standards and other standardization products are published by BSI Standards Limited.

About us

We bring together business, industry, government, consumers, innovators and others to shape their combined experience and expertise into standards-based solutions.

The knowledge embodied in our standards has been carefully assembled in a dependable format and refined through our open consultation process. Organizations of all sizes and across all sectors choose standards to help them achieve their goals.

Information on standards

We can provide you with the knowledge that your organization needs to succeed. Find out more about British Standards by visiting our website at bsigroup.com/standards or contacting our Customer Services team or Knowledge Centre.

Buying standards

You can buy and download PDF versions of BSI publications, including British and adopted European and international standards, through our website at bsigroup.com/shop, where hard copies can also be purchased.

If you need international and foreign standards from other Standards Development Organizations, hard copies can be ordered from our Customer Services team.

Subscriptions

Our range of subscription services are designed to make using standards easier for you. For further information on our subscription products go to bsigroup.com/subscriptions.

With **British Standards Online (BSOL)** you'll have instant access to over 55,000 British and adopted European and international standards from your desktop. It's available 24/7 and is refreshed daily so you'll always be up to date.

You can keep in touch with standards developments and receive substantial discounts on the purchase price of standards, both in single copy and subscription format, by becoming a **BSI Subscribing Member**.

PLUS is an updating service exclusive to BSI Subscribing Members. You will automatically receive the latest hard copy of your standards when they're revised or replaced.

To find out more about becoming a BSI Subscribing Member and the benefits of membership, please visit bsigroup.com/shop.

With a **Multi-User Network Licence (MUNL)** you are able to host standards publications on your intranet. Licences can cover as few or as many users as you wish. With updates supplied as soon as they're available, you can be sure your documentation is current. For further information, email bsmusales@bsigroup.com.

BSI Group Headquarters

389 Chiswick High Road London W4 4AL UK

Revisions

Our British Standards and other publications are updated by amendment or revision.

We continually improve the quality of our products and services to benefit your business. If you find an inaccuracy or ambiguity within a British Standard or other BSI publication please inform the Knowledge Centre.

Copyright

All the data, software and documentation set out in all British Standards and other BSI publications are the property of and copyrighted by BSI, or some person or entity that owns copyright in the information used (such as the international standardization bodies) and has formally licensed such information to BSI for commercial publication and use. Except as permitted under the Copyright, Designs and Patents Act 1988 no extract may be reproduced, stored in a retrieval system or transmitted in any form or by any means – electronic, photocopying, recording or otherwise – without prior written permission from BSI. Details and advice can be obtained from the Copyright & Licensing Department.

Useful Contacts:

Customer Services

Tel: +44 845 086 9001

Email (orders): orders@bsigroup.com

Email (enquiries): cservices@bsigroup.com

Subscriptions

Tel: +44 845 086 9001

Email: subscriptions@bsigroup.com

Knowledge Centre

Tel: +44 20 8996 7004

Email: knowledgecentre@bsigroup.com

Copyright & Licensing

Tel: +44 20 8996 7070

Email: copyright@bsigroup.com